Sheffield Hallam University

Modelling an hCG Immunoassay for Predicting Treatment in Low-Risk Gestational Trophoblastic Disease

JABBAR, Imran

Available from the Sheffield Hallam University Research Archive (SHURA) at:

https://shura.shu.ac.uk/38059/

A Sheffield Hallam University thesis

This thesis is protected by copyright which belongs to the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Please visit https://shura.shu.ac.uk/38059/ and http://shura.shu.ac.uk/information.html for further details about copyright and re-use permissions.

Modelling an hCG Immunoassay for Predicting Treatment in Low-Risk Gestational Trophoblastic Disease

Imran Jabbar

A thesis submitted in partial fulfilment of the requirements of

Sheffield Hallam University

for the degree of Doctor of Professional Studies

May 2025

Candidate Declaration

I hereby declare that:

1. I have not been enrolled for another award of the University, or other academic or professional organisation, whilst undertaking my research degree.

Material submitted for another award

2. None of the material contained in the thesis has been used in any other submission for an academic award.

Authorship and Collaboration

3. I certify that this thesis is my own work. The use of all published or other sources of material consulted have been properly and fully acknowledged.

Research conduct

4. The work undertaken towards the thesis has been conducted in accordance with the SHU Principles of Integrity in Research and the SHU Research Ethics Policy, and ethics approval has been granted for all research studies in the thesis, as shown in the table below.

Ethics review reference number	Title of research study	Approval date	Date of any post-approval amendments (if applicable)
ER19210156	Modelling an hCG immunoassay for treatment response in low-risk GTN	23/10/2019	Not applicable

Word count

5. The word count of the thesis is 67,119 (of which ~15,000 are Python code)

Signature

Please complete electronically:

Name	Imran Jabbar
Award	Doctor of Professional Studies
Date of Submission	7th May 2025
College	Health, Wellbeing and Life Sciences
Director(s) of Studies	Dr. Neil Cross

Table of Contents

Abstract..... i

Abbreviations..... ii

List of Figures..... iv

List of Tables..... ix

1.0 Introduction 1

1.1 Rationale for the Research 2

1.2 Historical Background of GTD and Epidemiology 2

1.2.1 Early Observations of Trophoblastic Disease 2

1.2.2 Chemotherapy and the Curative Era 3

1.2.3 Recognition of Rare Subtypes 4

1.2.4 Epidemiology and Risk Factors 4

1.3 Pathophysiology of GTD 7

1.3.1 Normal Trophoblastic Development 7

1.3.2 Genetic Basis of Hydatidiform Moles 8

1.3.3 Molecular Mechanisms of Trophoblastic Proliferation 9

1.3.4 High Cure Rates and Unique Malignancy 10

1.4 Biomarkers and hCG Assay Technology 10

1.4.1 hCG as the Central Biomarker 11

1.4.2 Analytical Platforms for hCG Measurement 11

1.4.3 Sheffield In-House Radioimmunoassay 12

1.4.4 Siemens IMMULITE 2000 chemiluminescent immunoassay 13

1.4.5 Global hCG Immunoassays 14

1.4.6 Immunoassay Interference 15

1.4.7 Clinical Utility of hCG in GTD 15

1.5 Hydatidiform Moles and Low-Risk Gestational Trophoblastic Disease 17

1.5.1 Pathophysiology and Molecular Mechanisms 18

1.5.2 Clinical Presentation and Symptoms 19

1.5.3 Diagnosis and Classification 22

1.5.4 Management Strategies 26
1.5.5 Risk Factors for Persistent GTD 28
1.5.6 Prognosis and Outcomes 30
1.5.7 Recent Advances and Research Directions 33
1.6 Organisation of GTD Services in the UK and Dataset Context 37
1.7 Malignant Gestational Trophoblastic Neoplasia (GTN) 39
1.7.1 Pathophysiology and Molecular Mechanisms 40
1.7.2 Epidemiology and Risk Factors 41
1.7.3 Clinical Presentation and Symptoms 42
1.7.4 Diagnostic Approaches 43
1.7.5 Treatment Strategies 47
1.7.6 Prognosis and Outcomes 49
1.8 Chemotherapy in GTD 50
1.8.1 Molecular Mechanisms of Action 50
1.8.2 Clinical Applications in GTD 53
1.8.3 Resistance Mechanisms 54
1.8.4 Methotrexate Pharmacokinetics 58
1.8.5 Actinomycin D Pharmacokinetics 59
1.9 Global Evidence on hCG Kinetics in Low-Risk GTD 60
1.9.1 Early Descriptive Kinetics (2000 – 2010) 61
1.9.2 Parametric Population-Kinetic Models (2010 – 2015) 61
1.9.3 Nomograms and Machine-Learning (2016 – present) 62
1.9.4 Synthesis and Implications for the Present Thesis 62
1.10 Hypothesis and Aims 63
1.10.1 Hypothesis 1 63
1.10.2 Hypothesis 2 63
1.10.3 Hypothesis 3 63
1.10.4 Hypothesis 4 64
2.0 Methods Chapter 65
2.1 Data Sources 65
2.1.1 Urine Dataset (Hypothesis 1 and 2) 66
2.1.2 Serum Dataset (Hypothesis 3 and 4) 69

2.2 Data Acquisition and Computational Environment 71
2.2.1 Data Loading and Initial Inspection 72
2.3 Logistic Regression Modelling – Serum Dataset 74
2.3.1 Data Upload in Google Colab 74
2.3.2 Data Loading and Inspection 74
2.3.3 Predictor and Target Definition 74
2.3.4 Train–Test Split 75
2.3.5 Predictor Scaling 76
2.3.6 Baseline Logistic Regression 77
2.3.7 Coefficient-Based Feature Selection 78
2.3.8 Refined Logistic Regression 79
2.3.9 Evaluation Metrics 80
2.3.10 Confusion Matrix Visualisation 83
2.3.11 ROC Curve 84
2.3.12 Cross-Validation 87
2.3.13 Feature Count Reporting 88
2.3.14 Coefficient Table 89
2.3.15 Odds Ratio Presentation 90
2.3.16 Parsimonious Model Development 91
2.3.17 Inclusion of Baseline hCG (Day 0) 95
2.3.18 Cross-Validation of Six-Feature Model 96
2.3.19 Clinical Prediction Tool Development 97
2.4 Gradient Boosting Modelling – Serum Dataset 101
2.4.1 Data Upload in Google Colab 101
2.4.2 Data Loading and Inspection 101
2.4.3 Data Cleaning and Feature Selection 101
2.4.4 Predictor and Target Definition 102
2.4.5 Train–Test Split 103
2.4.6 Feature Scaling Considerations 104
2.4.7 Baseline Gradient Boosting Model 105
2.4.8 Evaluation Metrics 107
2.4.9 ROC Curve 110

2.4.10 Feature Importance Visualisation 111
2.4.11 Hyperparameter Optimisation 112
2.5 Random Forest Modelling – Serum Dataset 122
2.5.1 Data Upload in Google Colab 122
2.5.2 Data Loading and Inspection 122
2.5.3 Predictor and Target Definition 123
2.5.4 Train–Test Split 124
2.5.5 Feature Scaling Considerations 125
2.5.6 Baseline Random Forest Model 126
2.5.7 Evaluation Metrics 127
2.5.8 ROC Curve 130
2.5.9 Feature Importance Visualisation 131
2.5.10 Hyperparameter Optimisation 132
2.5.11 Feature Importance Visualisation 142
2.5.12 Reduced-Feature Training 143
2.5.13 ROC Curve – 5 Feature 148
2.5.14 Comparative ROC Curves 149
2.6 Logistic Regression Modelling – Urine Dataset 151
2.6.1 Data Upload in Google Colab 151
2.6.2 Data Loading and Inspection 151
2.6.3 Library Imports 151
2.6.4 Data Loading and Inspection 152
2.6.5 Predictor and Target Definition 152
2.6.6 Train–Test Split 152
2.6.7 Logistic Regression 153
2.6.8 Evaluation Metrics 153
2.6.9 ROC Curve 154
2.6.10 Coefficient Analysis 155
2.7 Gradient Boosting Modelling – Urine Dataset 156
2.7.1 Data Upload in Google Colab 156
2.7.2 Data Loading and Inspection 156
2.7.3 Library Imports 156

2.7.4 Predictor and Target Definition 157
2.7.5 Train–Test Split 157
2.7.6 Baseline Gradient Boosting Model 158
2.7.7 Evaluation Metrics 158
2.7.8 ROC Curve 159
2.7.9 Confusion Matrix 160
2.7.10 Hyperparameter Optimisation – Grid Search 161
2.7.11 Hyperparameter Optimisation – Random Search 162
2.7.12 Evaluation of Tuned Model 167
2.7.13 ROC Curve – Tuned Model 168
2.7.14 Comparative ROC Curves 169
2.7.15 Confusion Matrix – Tuned Model 170
2.8 Random Forest Modelling – Urine Dataset 172
2.8.1 Data Upload in Google Colab 172
2.8.2 Data Loading and Inspection 172
2.8.3 Library Imports import numpy as np 172
2.8.4 Predictor and Target Definition 173
2.8.5 Train–Test Split 173
2.8.6 Baseline Random Forest Model 174
2.8.7 Evaluation Metrics 174
2.8.8 ROC Curve 175
2.8.9 Confusion Matrix 176
2.8.10 Hyperparameter Optimisation – Grid Search 177
2.8.11 Hyperparameter Optimisation – Random Search 177
2.8.11.1 Parameter Space Definition 177
2.8.12 Evaluation of Tuned Model 179
2.8.13 ROC Curve – Tuned Model 180
2.8.14 Confusion Matrix – Tuned Model 182
2.8.15 Comparative ROC Curves 182
2.8.16 Performance Comparison Table 183
2.9 Technical Notes and Reproducibility 186
2.9.1 Programming Languages and Libraries 186

2.9.2 Programming Languages and Scripting 186
2.9.3 Python libraries utilised 186
2.9.4 Machine-learning models implemented 187
2.9.5 Pre-processing & Feature Engineering 187
2.9.6 Hyperparameter Optimisation & Validation 187
2.9.7 Performance Metrics Reported 187
2.9.8 Reproducibility Measures 188
2.9.9 Ethical & Governance Controls 188
3.0 Results Chapter 189
3.1 Introduction 189
3.1.1 Model Evaluation Metrics 190
3.2 Serum Dataset Modelling (MTX-Treated Patients) 194
3.2.1 Dataset Summary 194
3.2.2 Logistic Regression (12-Feature Model) 195
3.2.3 Logistic Regression (Top 5 Feature Model) 200
3.2.4 Logistic Regression (Top 5 + Day0_hCG Model) 203
3.2.5 Logistic Regression Model Comparison 206
3.2.6 Gradient Boosting (Untuned) 209
3.2.7 Gradient Boosting (Tuned) 212
3.2.8 Gradient Boosting Model Comparison 215
3.2.9 Random Forest (Untuned) 217
3.2.10 Random Forest (Tuned) 220
3.2.11 Random Forest (Tuned and Top 5 Features) 223
3.2.12 Random Forest Model Comparison 226
3.3 Urine Dataset Modelling (Untreated Surveillance Patients) 228
3.3.1 Dataset Summary 228
3.3.2 Logistic Regression (Full 12-Feature Model) 229
3.3.3 Gradient Boosting (Untuned) 232
3.3.4 Gradient Boosting (Tuned) 235
3.3.5 Gradient Boosting Model Comparison 238
3.3.6 Random Forest (Untuned) 241
3.3.7 Random Forest (Tuned) 244

3.3.8 Random Forest Model Comparison 247
3.4 Final Performance Metrics 250
3.4.1 Serum Models 250
3.4.2 Urine Models 252
3.4.3 Cross-Dataset Insights and Model Family Comparison 253
3.4.4 Conclusion of Comparative Analysis 253
4.0 Discussion Chapter 255
4.1 Introduction 255
4.2 Urine hCG Dynamics and Identification of Post-Molar GTD (Hypotheses 1 & 2) 255
4.3 Serum hCG Kinetics and Early Prediction of MTX Resistance (Hypotheses 3 & 4) 258
4.4 Strengths and Limitations of the Study 265
4.4.1 Strengths 265
4.4.2 Limitations 267
4.5 Implications for Clinical Practice 270
4.5.1 Early identification and intervention 270
4.5.2 Optimisation of treatment pathways 270
4.5.3 Personalisation of follow-up and patient counselling 271
4.5.4 Integration into clinical workflows 271
4.5.5 Potential pitfalls and risks 272
4.5.6 Generalisability and variation across settings 272
4.5.7 Future directions and validation 273
4.6 Recommendations for Further Research 275
4.6.1 Prospective Validation 275
4.6.2 Model Refinement and Advanced Modelling Approaches 275
4.6.3 Integration of Additional Clinical Predictors 276
4.6.4 Clinical Evaluation and Implementation 276
4.6.5 Understanding Model Error and Further Optimisation 277
5.0 Conclusion 278
6.0 References 280
7.0 Appendix 303

Abstract

Gestational trophoblastic disease (GTD) encompasses disorders of abnormal trophoblast proliferation, including hydatidiform mole and choriocarcinoma, which are clinically monitored by serial measurements of human chorionic gonadotropin (hCG). Early identification of patients whose hCG levels will normalise spontaneously versus those requiring chemotherapy is critical for optimising care and avoiding overtreatment. This study develops predictive machine learning tools to stratify GTD patients based on early hCG trajectory patterns. Two large cohorts were analysed: a post-molar pregnancy cohort with longitudinal urinary hCG measurements, and a cohort of GTD patients treated with first-line methotrexate (MTX) with serial serum hCG monitoring. Time-series feature engineering was applied to capture kinetic characteristics of hCG decline (such as rates of change and pattern metrics), which were used as inputs to classifiers including logistic regression, gradient boosting machines, and random forests. Ensemble models (such as gradient boosting and random forest, which combine multiple decision trees to improve predictive performance) outperformed the linear logistic regression model by capturing nonlinear trends in hCG dynamics. In the urine cohort, the ensemble classifier achieved high specificity and strong sensitivity for predicting cases that ultimately required chemotherapy, indicating robust early recognition of high-risk patients. In the MTX-treated serum cohort, the models detected early deviations in hCG decline indicative of emerging methotrexate resistance, enabling identification of patients likely to need second-line therapy. These findings demonstrate that early hCG trajectories contain informative patterns correlated with treatment outcome. Integrating these predictive models into clinical practice could support decision-making by forecasting individual patient courses, enabling prompt intervention for non-responders while reducing unnecessary chemotherapy. Together, this work supports the use of early hCG trajectories for personalised risk stratification, enabling timely intervention while minimising unnecessary chemotherapy.

Abbreviations

- **ActD:** Actinomycin D
- **α**: Alpha
- **AFP**: Alpha-Fetoprotein
- **AUC:** Area Under the Curve
- **β:** Beta
- **CA 19-9:** Cancer Antigen 19-9
- **CA-125:** Cancer Antigen 125
- **CHM**: Complete Hydatidiform Mole
- **CEA:** Carcinoembryonic Antigen
- **CSF:** Cerebrospinal Fluid
- **EMA-CO:** Etoposide, Methotrexate, Actinomycin D, Cyclophosphamide, and Vincristine
- **EP:** Etoposide and Cisplatin
- **ETT:** Epithelioid Trophoblastic Tumour
- **FIGO:** International Federation of Gynaecology and Obstetrics
- **FN:** False Negative
- **FP:** False Positive
- **FSH:** Follicle-Stimulating Hormone
- **GBM:** Gradient Boosting Machine
- **GTD:** Gestational Trophoblastic Disease
- **GTN:** Gestational Trophoblastic Neoplasia
- **hCG:** Human Chorionic Gonadotropin
- **HDCT + ASCT:** High Dose Chemotherapy with Autologous Stem Cell Transplantation
- **HhCG:** Hyperglycosylated hCG
- **HM:** Hydatidiform Mole

- **hPL:** Human Placental Lactogen
- **IM**: Intramuscular
- **IV:** Intravenous
- **IU/L:** International Units per Litre
- **L-BFGS:** Limited-memory Broyden–Fletcher–Goldfarb–Shanno
- **LDH:** Lactate Dehydrogenase
- **LH:** Luteinising Hormone
- **LR:** Logistic Regression
- **MDT:** Multi-Disciplinary Team
- **ML:** Machine Learning
- **MTX:** Methotrexate
- **NPV:** Negative Predictive Value
- **P-GP:** P-Glycoprotein
- **PHM:** Partial Hydatidiform Mole
- **PPV**: Positive Predictive Value
- **PSTT:** Placental Site Trophoblastic Tumour
- **PTD:** Persistent Trophoblastic Disease
- **RF:** Random Forest
- **RIA:** Radioimmunoassay
- **ROC:** Receiver Operating Characteristic
- **SD:** Standard Deviation
- **STH**: Sheffield Teaching Hospitals
- **TGF-β:** Transforming Growth Factor-Beta
- **TIMPs:** Tissue Inhibitors of Metalloproteinases
- **TN:** True Negative

- **TP:** True Positive
- **TP-TE:** Paclitaxel and Etoposide
- **TSH:** Thyroid-Stimulating Hormone
- **UK:** United Kingdom
- **NEQAS:** National External Quality Assessment Scheme
- **US:** Ultrasound
- **WHO:** World Health Organisation

List of Figures

Label	Full Description	Page
Figure 2.1	Google Colaboratory domain interface shown at initial access (). Source: Author's own screenshot.	71
Figure 2.2	Uploading files to Google Colab using files.upload() to manually select the dataset. Source: Author's own Python code executed in Google Colab.	72
Figure 2.3	Initial dataset preview using df.head() to verify successful import of hCG data. This dataset would differ depending on if the serum dataset is used, or the urine dataset. Source: Author's own Python code.	73
Figure 2.4	Code dropping non-predictive columns and separating predictors X and target y. Model: Serum Logistic Regression. Source: Author's own Python code.	74
Figure 2.5	Code separating Training and Testing Groups. Model: Serum Logistic Regression. Source: Author's own Python code.	76
Figure 2.6	Standardisation of numerical features using StandardScaler. Model: Serum Logistic Regression. Source: Author's own Python code.	77
Figure 2.7	Code training the model on the scaled dataset. Model: Serum Logistic Regression. Source: Author's own Python code.	78
Figure 2.8	Code to generate Feature Selection. Model: Serum Logistic Regression. Source: Author's own Python code.	79
Figure 2.9	Code retraining the model based on Feature Selection. Model: Serum Logistic Regression. Source: Author's own Python code.	79
Figure 2.10	Code to generate the Model Evaluation. Model: Serum Logistic Regression. Source: Author's own Python code.	83
Figure 2.11	Code to display the Confusion Matrix. Model: Serum Logistic Regression. Source: Author's own Python code.	84
Figure 2.12	Code to display the ROC curve. Model: Serum Logistic Regression. Source: Author's own Python code.	86
Figure 2.13	Code to generate Cross Validation AUC score. Model: Serum Logistic Regression. Source: Author's own Python code.	88
Figure 2.14	Code to generate Feature Selection. Model: Serum Logistic Regression. Source: Author's own Python code	89
Figure 2.15	Code to present Coefficient Table. Model: Serum Logistic Regression. Source: Author's own Python code	90
Figure 2.16	Code to present Odds Ratios. Model: Serum Logistic Regression. Source: Author's own Python code	91
Figure 2.17	Code to generate the Model Evaluation. Model: Serum Logistic Regression (Reduced). Source: Author's own Python code	94
Figure 2.18	Code to generate the Model Evaluation. Model: Serum Logistic Regression (Reduced+1). Source: Author's own Python code	96
Figure 2.19	Code to generate Cross Validation AUC score. Model: Serum Logistic Regression (Reduced+1). Source: Author's own Python code	97
Figure 2.20	Code to drop irrelevant Columns. Model: Serum Gradient Boosting. Source: Author's own Python code	102
Figure 2.21	Code separating predictors X and target y. Model: Serum Gradient Boosting. Source: Author's own Python code	103
Figure 2.22	Code separating Training and Testing Groups. Model: Serum Gradient Boosting. Source: Author's own Python code.	104
Figure 2.23	Code training Model on default Hyperparameters. Model: Serum Gradient Boosting. Source: Author's own Python code.	106
Figure 2.24	Code to prevent label_encoder=False error. Model: Serum Gradient Boosting. Source: Author's own Python code	107

Figure 2.25	Code to generate the Model Evaluation. Model: Serum Gradient Boosting. Source: Author's own Python code	109
Figure 2.26	Code to plot ROC Curve. Model: Serum Gradient Boosting. Source: Author's own Python code	111
Figure 2.27	Code to Visualise Feature importance. Model: Serum Gradient Boosting. Source: Author's own Python code	112
Figure 2.28	Code to set a Parameter Grid. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	114
Figure 2.29	Code to GridSearchCV and generate the best Hyperparameters. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	116
Figure 2.30	Fitting the Hyperparameters to the Model. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	117
Figure 2.31	Code to generate the Model Evaluation and produce a ROC curve. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	121
Figure 2.32	Code to drop irrelevant Columns, and separating predictors X and target y. Model: Serum Random Forest. Source: Author's own Python code.	124
Figure 2.33	Code separating Training and Testing Groups and displaying results for confirmation. Model: Serum Random Forest. Source: Author's own Python code.	125
Figure 2.34	Code training Model on default Hyperparameters. Model: Serum Random Forest. Source: Author's own Python code	127
Figure 2.35	Code to generate the Model Evaluation. Model: Serum Random Forest. Source: Author's own Python code.	129
Figure 2.36	Code to generate ROC curve. Model: Serum Random Forest. Source: Author's own Python code.	131
Figure 2.37	Code to Visualise Feature importance. Model: Serum Random Forest. Source: Author's own Python code.	132
Figure 2.38	Code to set a Parameter Grid. Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code.	134
Figure 2.39	Code to GridSearchCV and generate the best Hyperparameters. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	135
Figure 2.40	Code to Visualise the best Hyperparameters. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	136
Figure 2.41	Fitting the Hyperparameters to the Model. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code	137
Figure 2.42	Code to generate the Model Evaluation. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	140
Figure 2.43	Code to generate the ROC Curve. Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	141
Figure 2.44	Code to Visualise Feature importance. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	143
Figure 2.45	Code to Visualise Top 5 Features and their Importance Score. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	144
Figure 2.46	Code to Substitute Original Feature data to the Top 5 Features only. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	146

Figure 2.47	Code Training the Model on the Top 5 Features only. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	146
Figure 2.48	Code to generate the Model Evaluation. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	147
Figure 2.49	Code to generate the ROC Curve. Model: Serum Reduced Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.	148
Figure 2.50	Code to generate a comparison of all three Gradient Boosted ROC Curves. Source: Author's own Python code.	150
Figure 2.51	Code to Import the Required Libraries. Model: Urine Logistic Regression. Source: Author's own Python code.	151
Figure 2.52	Code to Visualise the Dataset. Model: Urine Logistic Regression. Source: Author's own Python code.	152
Figure 2.53	Code to remove unnecessary columns and assigns y as the target variable. Model: Urine Logistic Regression. Source: Author's own Python code.	152
Figure 2.54	Code separating Training and Testing Groups. Model: Urine Logistic Regression. Source: Author's own Python code.	152
Figure 2.55	Code fitted to training data and optimised. Model: Urine Logistic Regression. Source: Author's own Python code.	153
Figure 2.56	Code to generate the Model Evaluation. Model: Urine Logistic Regression. Source: Author's own Python code.	154
Figure 2.57	Code to generate ROC Curve. Model: Urine Logistic Regression. Source: Author's own Python code.	155
Figure 2.58	Code to view Feature Coefficients. Model: Urine Logistic Regression. Source: Author's own Python code.	155
Figure 2.59	Code to Import Required Libraries. Model: Urine Gradient Boosting. Source: Author's own Python code.	157
Figure 2.60	Code to remove unnecessary columns and assigns y as the target variable. Model: Urine Gradient Boosting. Source: Author's own Python code.	157
Figure 2.61	Code separating Training and Testing Groups. Model: Urine Gradient Boosting. Source: Author's own Python code.	157
Figure 2.62	Code training Model on default Hyperparameters. Model: Urine Gradient Boosting. Source: Author's own Python code.	158
Figure 2.63	Code to generate the Model Evaluation. Model: Urine Gradient Boosting. Source: Author's own Python code.	159
Figure 2.64	Code to generate the ROC Curve. Model: Urine Gradient Boosting. Source: Author's own Python code.	160
Figure 2.65	Code to generate the Confusion Matrix. Model: Urine Gradient Boosting. Source: Author's own Python code.	160
Figure 2.66	Randomised Hyperparameter distribution defined for tuning the Model. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	164
Figure 2.67	Code to RandomizedSearchCV, using 5-fold cross-validation. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	165
Figure 2.68	Code to fit the training data to RandomizedSearchCV. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	166
Figure 2.69	Code for retrieval of the best-performing hyperparameter. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	167
Figure 2.70	Code to generate the Model Evaluation. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	168

Figure 2.71	Code to generate the ROC Curve. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.	169
Figure 2.72	Code to generate both the untuned and hyperparameter tuned ROC Curves. Source: Author's own Python code.	170
Figure 2.73	Code to generate the Confusion Matrix. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code	171
Figure 2.74	Code to Import Required Libraries. Model: Urine Random Forest. Source: Author's own Python code.	173
Figure 2.75	Code to remove unnecessary columns and assigns y as the target variable. Model: Urine Random Forest. Source: Author's own Python code.	173
Figure 2.76	Code separating Training and Testing Groups. Model: Urine Random Forest. Source: Author's own Python code.	173
Figure 2.77	Code training Model on default Hyperparameters. Model: Urine Random Forest. Source: Author's own Python code.	174
Figure 2.78	Code to generate the Model Evaluation. Model: Urine Random Forest. Source: Author's own Python code.	175
Figure 2.79	Code to generate the ROC Curve. Model: Urine Random Forest. Source: Author's own Python code.	176
Figure 2.80	Code to generate the Confusion Matrix. Model: Urine Random Forest. Source: Author's own Python code.	176
Figure 2.81	Randomised Hyperparameter distribution defined for tuning the Model. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	177
Figure 2.82	Code to RandomizedSearchCV, using 5-fold cross-validation. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	178
Figure 2.83	Code to fit the training data to RandomizedSearchCV. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	179
Figure 2.84	Code for retrieval of the best-performing hyperparameters. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	179
Figure 2.85	Code to generate the Model Evaluation. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	180
Figure 2.86	Code to generate the ROC Curve. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	181
Figure 2.87	Code to generate the Confusion Matrix. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.	182
Figure 2.88	Code to generate the untuned and the Hyperparameter tuned ROC Curves. Source: Author's own Python code.	183
Figure 2.89	Code to generate an untuned vs Hyperparameter tuned comparison table. Source: Author's own Python code.	185
Figure 3.1	Confusion Matrix. Model: Serum Logistic Regression. Source: Generated in Python using Author's own code.	198
Figure 3.2	ROC Curve. Model: Serum Logistic Regression. Source: Generated in Python using Author's own code.	198
Figure 3.3	Confusion Matrix. Model: Serum Logistic Regression (5 Feature). Source: Generated in Python using Author's own code.	201
Figure 3.4	ROC Curve. Model: Serum Logistic Regression (5 Feature). Source: Generated in Python using Author's own code.	202
Figure 3.5	Confusion Matrix. Model: Serum Logistic Regression (6 Feature). Source: Generated in Python using Author's own code.	204
Figure 3.6	ROC Curve. Model: Serum Logistic Regression (6 Feature). Source: Generated in Python using Author's own code.	205
Figure 3.7	Performance Graph highlighting key metric comparison between Serum Logistic Regression Models. Source: Generated in Python using Author's own code.	206

Figure 3.8	ROC comparison between Serum Logistic Regression Models. Source: Generated in Python using Author's own code.	206
Figure 3.9	Confusion Matrix. Model: Serum Gradient Boosting (untuned). Source: Generated in Python using Author's own code.	210
Figure 3.10	ROC Curve. Model: Serum Gradient Boosting (untuned). Source: Generated in Python using Author's own code.	211
Figure 3.11	Confusion Matrix. Model: Serum Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.	213
Figure 3.12	ROC Curve. Model: Serum Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.	214
Figure 3.13	Performance Graph highlighting key metric comparison between Serum Gradient Boosting Models. Source: Generated in Python using Author's own code.	215
Figure 3.14	ROC comparison between Serum Gradient Boosting Models. Source: Generated in Python using Author's own code.	215
Figure 3.15	Confusion Matrix. Model: Serum Random Forest (untuned). Source: Generated in Python using Author's own code.	219
Figure 3.16	ROC Curve. Model: Serum Random Forest (untuned). Source: Generated in Python using Author's own code.	219
Figure 3.17	Confusion Matrix. Model: Serum Random Forest (Tuned). Source: Generated in Python using Author's own code.	222
Figure 3.18	ROC Curve. Model: Serum Random Forest (Tuned). Source: Generated in Python using Author's own code.	222
Figure 3.19	Confusion Matrix. Model: Serum Random Forest (Tuned + Top 5 Features). Source: Generated in Python using Author's own code.	225
Figure 3.20	*Figure 3.20: ROC. Model: Serum Random Forest (Tuned + Top 5 Features). Source: Generated in Python using Author's own code.*	225
Figure 3.21	Performance Graph highlighting key metric comparison between Serum Random Forest Models. Source: Generated in Python using Author's own code.	226
Figure 3.22	ROC comparison between Serum Random Forest Models. Source: Generated in Python using Author's own code.	226
Figure 3.23	Confusion Matrix. Model: Urine Logistic Regression. Source: Generated in Python using Author's own code.	230
Figure 3.24	ROC Curve. Model: Urine Logistic Regression. Source: Generated in Python using Author's own code.	231
Figure 3.25	Confusion Matrix. Model: Urine Gradient Boosting (Untuned). Source: Generated in Python using Author's own code.	233
Figure 3.26	ROC Curve. Model: Urine Gradient Boosting (Untuned). Source: Generated in Python using Author's own code.	234
Figure 3.27	Confusion Matrix. Model: Urine Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.	236
Figure 3.28	ROC Curve. Model: Urine Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.	237
Figure 3.29	Performance Graph highlighting key metric comparison between Urine Gradient Boosting Models. Source: Generated in Python using Author's own code.	238
Figure 3.30	ROC comparison between Urine Gradient Boosting Models. Source: Generated in Python using Author's own code.	238
Figure 3.31	Confusion Matrix. Model: Urine Random Forest (untuned). Source: Generated in Python using Author's own code.	242
Figure 3.32	ROC Curve. Model: Urine Random Forest (untuned). Source: Generated in Python using Author's own code.	243
Figure 3.33	Confusion Matrix. Model: Urine Random Forest (Tuned). Source: Generated in Python using Author's own code.	245

Figure 3.34	ROC Curve. Model: Urine Random Forest (Tuned). Source: Generated in Python using Author's own code	246
Figure 3.35	Performance Graph highlighting key metric comparison between Urine Random Forest Models. Source: Generated in Python using Author's own code.	247
Figure 3.36	ROC comparison between Urine Random Forest Models. Source: Generated in Python using Author's own code.	247

List of Tables

Table 1.1	FIGO/WHO Prognostic Scoring System for GTN, used for risk stratification and treatment allocation. Source: Adapted from International Federation of Gynaecology and Obstetrics (2002) and updated guidance (Ngan et al., 2018).	16
Table 2.1	Logistic Regression Model Coefficients Table for use in Excel Prediction Tool. Source: Generated in Python using Author's own code.	99
Table 2.2	Summary of Python libraries utilised during Chapter 2. Source: Compiled by Author manually.	186
Table 3.1	Performance Table highlighting key metric comparison between Serum Models. Source: Generated in Python using Author's own code.	250
Table 3.2	Performance Table highlighting key metric comparison between Urine Models. Source: Generated in Python using Author's own code.	252
Table 4.1	Strengths and Weaknesses of Published hCG-Based Models. Source: Author Generated from review of existing Literature.	264

1.0 Introduction

Gestational trophoblastic disease (GTD) refers to a group of disorders arising from abnormal proliferation of trophoblastic cells after conception. These conditions range from benign hydatidiform moles to malignant entities including persistent GTN, choriocarcinoma, placental site trophoblastic tumour (PSTT) and epithelioid trophoblastic tumour (ETT) (Singh et al., 2024). GTD represents a small subset of gynaecologic conditions but is clinically significant due to its potential for malignancy, rapid spread, and unique biology that distinguishes it from typical cancers (Seckl et al., 2010). Importantly, GTD is one of the earliest solid tumours to be rendered highly curable with chemotherapy – even metastatic disease often achieves remission – making it a model for successful cancer therapy (Lurain, 2010). In low-risk GTD (e.g. non-metastatic cases), cure rates approach 100% (Winter, 2021), and survival outcomes for high-risk cases have substantially improved following the introduction of multi-agent chemotherapy regimens (Berkowitz and Goldstein, 1995). Despite these successes, challenges remain in optimising care: identifying patients at risk of treatment failure, minimising chemotherapy toxicity, and addressing the psychological and reproductive impact of diagnosis and treatment are ongoing priorities (Winter, 2021; Singh et al., 2024). A hallmark of GTD is its dependence on the hormone human chorionic gonadotropin (hCG) as a diagnostic and monitoring biomarker. hCG, produced by trophoblastic tissue, is central to detection, management, and post-treatment surveillance of GTD. Serial quantitative hCG measurements allow early identification of persistent disease, assessment of treatment efficacy, and monitoring for relapse (Berkowitz and Goldstein, 2009). However, biological variation in hCG kinetics between patients, alongside analytical complexities associated with hCG measurement, can complicate interpretation of serial results and clinical decision-making (McMahon et al., 2024). From a research perspective, GTD offers fascinating insights into trophoblastic cell biology and tumourigenesis. Unlike many malignancies driven primarily by progressive accumulation of somatic mutations, many forms of GTD arise following abnormal fertilisation events associated with altered parental genomic contribution and disruption of genomic imprinting mechanisms (Li et al., 2002). Genomic imprinting is an epigenetic mechanism in which gene expression is determined by parental origin (Barlow and Bartolomei, 2014), and in GTD this process is disrupted, typically resulting in an excess of paternal genetic material that drives abnormal trophoblastic proliferation and increased production of hCG (Li et al., 2002). Studying GTD yields knowledge not only about its own pathophysiology but also broader concepts of tumour growth, metastasis, and immune evasion (Li et al., 2002; Hanahan, 2022)

1.1 Rationale for the Research

The motivation for this research is to further improve outcomes and reduce unnecessary chemotherapy in GTD, particularly in patients with persistent disease. Existing therapies are highly effective, but a key challenge is the inability to predict early which patients will respond to first-line treatment and which will develop resistance or relapse (van Trommel et al., 2006). hCG dynamics play a central role in guiding therapy decisions, yet significant individual variability exists (McMahon et al., 2024). Developing more advanced predictive tools – including mathematical models of hCG kinetics and molecular biomarkers of chemoresistance – could enable more personalised care (Parker et al., 2023). The over-arching hypothesis is that distinct hCG decline profiles may differentiate benign vs. progressive cases, allowing earlier intervention in those who need it and avoiding overtreatment in those who do not.

1.2 Historical Background of GTD and Epidemiology

GTD represents a rare but clinically distinctive spectrum of diseases, notable for their unique biological origin and exceptional responsiveness to therapy. The history of GTD encompasses key milestones in pathology and treatment that have shaped modern management, Including the identification of hCG as a tumour marker, which enabled biochemical monitoring of disease progression, and the subsequent development of effective chemotherapy protocols, notably methotrexate-based regimens and later multi-agent combinations such as EMA/CO, which transformed GTD into one of the most curable solid malignancies (Bagshawe, 1976; Newlands et al., 1986).

Alongside these therapeutic advances, epidemiological research has played a critical role in defining disease incidence, identifying key risk factors such as maternal age, and demonstrating variation between populations (Savage et al., 2010; Seckl et al., 2010). This section reviews the historical development and epidemiological characteristics of GTD, providing essential context for understanding how contemporary management strategies and risk stratification approaches have been shaped by both biological insights and population-level data.

1.2.1 Early Observations of Trophoblastic Disease

The history of GTD dates to ancient times with early descriptions of molar pregnancies in the writings of Hippocrates and other physicians (Lurain, 2010). However, it wasn't until the 19th century that the pathological basis of trophoblastic disease was recognised. In 1827, Madame

Boivin provided one of the earliest detailed descriptions of hydatidiform mole, recognising the abnormal grape-like cystic structures of the chorionic villi that we now associate with this condition (Hunter, 1959). This represented one of the earliest formal recognitions of hydatidiform mole as a separate pathological entity from normal pregnancy. The term “choriocarcinoma” was introduced in the late 19th century following the work of Marchand, who described the malignant counterpart of molar pregnancy and recognised its trophoblastic origin (Cole and Butler, 2015). His description of aggressive, vascular tumours of trophoblastic origin contributed to the distinction between benign and malignant trophoblastic disease. Later, Novak and Te Linde refined the diagnostic criteria for hydatidiform moles, invasive moles and choriocarcinoma, allowing for more accurate disease classification (as reviewed by Cunningham et al., 2018). The introduction of chemotherapy represented a major turning point in GTD management, with Li, Hertz and Bergenstal demonstrating successful treatment of choriocarcinoma using folic acid antagonists, establishing GTD as one of the first metastatic solid tumour shown to be curable with chemotherapy (Li et al., 1958).

1.2.2 Chemotherapy and the Curative Era

Following the demonstration that choriocarcinoma was highly sensitive to chemotherapy, subsequent work focused on optimising treatment protocols, improving risk stratification, and identifying markers of treatment response. The pioneering work of Kenneth Bagshawe in the United Kingdom (UK) was instrumental in advancing the use of methotrexate for the treatment of trophoblastic disease and refining early treatment protocols (Bagshawe, 1963; Bagshawe, 1976).

Early clinical series demonstrated the effectiveness of chemotherapy in trophoblastic disease. In a cohort reported by Bagshawe, 17 out of 23 treated patients achieved complete remission, highlighting the remarkable sensitivity of these tumours to chemotherapeutic agents. Importantly, serial measurement of urinary gonadotrophin levels was shown to provide a sensitive and reliable indicator of treatment response, allowing early identification of therapeutic success or resistance and establishing hCG monitoring as a central component of disease management (Bagshawe, 1963).

The development of centralised specialist centres and systematic hCG monitoring further improved patient outcomes and standardised care pathways, forming the basis of the modern UK GTD service model (Seckl et al., 2010). Building on this, Newlands and colleagues later developed the EMA/CO (etoposide, methotrexate and actinomycin-D alternating with

cyclophosphamide and vincristine) regimen, which remains a cornerstone regimen for high-risk disease and contributed substantially to the modern high cure rates observed in GTN (Newlands et al., 1986). Methotrexate (MTX) and actinomycin-D (ActD) became established as standard single-agent therapies for post-molar gestational trophoblastic neoplasia (GTN), while multi-agent regimens transformed outcomes in high-risk disease. What had once been a frequently fatal diagnosis became one of the most curable malignancies in oncology (Winter, 2021; Goldstein and Berkowitz, 2012). Today, low-risk GTN has an approximately 99–100% survival with appropriate therapy (Winter, 2021), and high-risk metastatic GTN can achieve survival rates approaching 90% with aggressive multi-agent chemotherapy (Goldstein and Berkowitz, 2012). This extraordinary success in cancer treatment is widely attributed to the pioneering developments in GTD, which established principles of chemotherapy sensitivity, biomarker-guided treatment, and centralised care.

1.2.3 Recognition of Rare Subtypes

In the late 20th century, rarer forms of GTD were delineated as distinct entities PSTT was recognised following descriptions of unusual trophoblastic tumours arising from intermediate trophoblast cells, while ETT was later characterised as a separate neoplastic entity with distinct pathological features (Kurman et al., 1976; Shih and Kurman, 1998). Unlike choriocarcinoma, these tumours produce low levels of hCG and tend to be more chemotherapy resistant, often requiring surgical management. Recognition of PSTT and ETT expanded the GTD spectrum and underscored the need for tailored diagnostic and therapeutic approaches (Goldstein and Berkowitz, 2012). It also highlighted that not all GTN fit the classical mould of high hCG and chemotherapy responsiveness – an important consideration in modern management.

1.2.4 Epidemiology and Risk Factors

Though uncommon, GTD has well-described epidemiologic patterns and risk factors. Understanding these helps identify populations in whom GTD is more likely and offers clues to its aetiology.

1.2.4.1 Incidence and Geographic Variation

Hydatidiform moles are an uncommon pregnancy complication worldwide, but incidence varies significantly by geographic region. In North America and Europe, the incidence of complete mole

is roughly 1 in 500–1,000 pregnancies, and partial moles around 1 in 1,000 (Seckl et al., 2010; Savage et al., 2010). In contrast, historically higher rates have been reported in parts of East and Southeast Asia, and some lower-resource settings, with some historical studies reporting substantially higher incidences than in Western countries (Matsui et al., 2007; Candelier, 2016). Earlier epidemiological studies from Japan reported substantially higher rates of molar pregnancy than those observed in Western populations, although incidence declined significantly between the 1970s and 2000s (Matsui et al., 2003).

Reported geographic differences have narrowed over time, potentially reflecting changes in nutrition, socioeconomic factors, diagnostic practices and population-based reporting; however, variation in GTD incidence between populations remains recognised (Matsui et al., 2003; Hui et al., 2017). The aetiology of geographic disparity is not fully understood, with proposed contributions from genetic susceptibility, environmental influences and dietary/nutritional differences (Candelier, 2016; Hui et al., 2017). Interestingly, migrant studies show that women born in high-incidence regions retain a higher risk of molar pregnancy even after moving to lower-incidence regions, suggesting early-life or genetic factors are at play (Savage et al., 2010; Candelier, 2016).

Similar patterns have been observed in UK-based cohorts, including work from the Sheffield centre, where variation in incidence has been associated with ethnicity within a centralised referral population (Tham et al., 2003). Overall, current global incidence ranges roughly from 0.5–1 per 1000 pregnancies in developed regions, with higher rates reported in some Asian and other populations, though precise estimates vary by study methodology and population characteristics (Altieri et al., 2003).

1.2.4.2 Maternal Age

One of the strongest risk factors for molar pregnancy is extremes of maternal age. The risk rises significantly for women at the upper end of reproductive age: women over 35 have a several-fold higher risk, and those over 40 have about a 5–10-fold increased risk of a complete mole compared to women in their 20s (La Vecchia et al., 1985; Savage et al., 2010; Gonzalez et al., 2024). This risk continues to increase with advancing maternal age, particularly beyond 40 years (La Vecchia et al., 1985). Very young maternal age (<20 years) is also associated with a modest increase in risk (Altieri et al., 2003), although the effect is less dramatic than at advanced ages. At age extremes, complete moles also constitute a higher proportion of GTD cases (e.g. >90% of GTDs in women

>45 are complete moles), whereas partial mole incidence appears less dependent on maternal age, with the rise in GTD among older women predominantly attributed to increased complete mole incidence (Savage et al., 2010). The biological mechanisms underlying the age effect are not fully understood but may relate to changes in oocyte quality and increased susceptibility to chromosomal or epigenetic errors that predispose to abnormal fertilisation and molar development (Hui et al., 2017; Candelier, 2016).

1.2.4.3 Prior Molar Pregnancy

A history of GTD greatly increases the risk of a subsequent molar pregnancy. After one molar pregnancy, the risk of another is on the order of 1–2% (about a 5–10-fold increase over baseline risk) (Seckl et al., 2010). If a woman has had two prior molar pregnancies, the risk of a third rises dramatically, estimated at 15–20% (Seckl et al., 2010; Nguyen and Slim, 2014). In such cases, especially if the moles were genetically confirmed complete moles, there may be an underlying genetic predisposition such as mutations in NLRP7, and changing the father/partner does not reduce the recurrence risk (Murdoch et al., 2006; Nguyen and Slim, 2014). Because of this recurrence risk, patients with a history of GTD are advised to have early ultrasound confirmation in future pregnancies to rule out another mole. Despite this increased risk, most women achieve a normal pregnancy outcome following a previous molar pregnancy, although early surveillance is recommended in subsequent pregnancies (Seckl et al., 2010).

1.2.4.4 Nutritional and Environmental Factors

Epidemiological patterns have long suggested a potential association between nutrition and GTD. Diets low in protein, folic acid, or carotene (vitamin A) have been associated with higher rates of molar pregnancy in some populations, although a direct causal relationship remains uncertain (Altieri et al., 2003; Candelier, 2016). Nutritional factors have been proposed to influence oocyte development and epigenetic regulation, potentially contributing to abnormal fertilisation events involved in molar pregnancy development (Candelier, 2016). One hypothesis suggests that nutritional influences during oocyte development may contribute to imprinting abnormalities that predispose to androgenetic fertilisation and mole formation (Coullin et al., 2015; Candelier, 2016). A study examining populations in Morocco and Senegal proposed that maternal nutritional factors, particularly vitamin A and folate status, may contribute to the higher incidence of complete hydatidiform mole observed in these populations, potentially through trans-generational effects (Coullin et al., 2015). While direct causal links remain difficult to establish, declining rates of molar pregnancy in some historically high-incidence regions,

including Japan, have occurred alongside improvements in socioeconomic conditions, healthcare access and nutritional status (Matsui et al., 2003; Hui et al., 2017).

1.2.4.5 Additional Factors

Other possible risk factors have been investigated. Differences associated with ethnicity have also been reported independently of geographic location; for instance, women of Asian ancestry in Western countries still have higher rates of GTD than the general population (Altieri et al., 2003; Tham et al., 2003;). Although the paternal genome plays a central role in the development of androgenetic complete moles, specific paternal factors such as advanced paternal age have not been consistently demonstrated as major independent risk factors (Fisher and Maher, 2021). Earlier studies reported possible associations between molar pregnancy and factors such as parental blood group combinations or previous oral contraceptive use, although findings have been inconsistent and these are not considered established risk factors (Parazzini et al., 1985; La Vecchia et al., 1985).

1.3 Pathophysiology of GTD

GTD arises from the trophoblast, a specialised cell layer that plays a pivotal role in early embryonic development and implantation. The pathophysiology of GTD is characterised by genetic abnormalities, dysregulated trophoblastic cell growth, and alterations in molecular signalling pathways that govern proliferation, differentiation, and invasion (Li et al., 2002; Hui et al., 2017; Fisher and Maher, 2021).

1.3.1 Normal Trophoblastic Development

To understand the pathophysiology of GTD, it is essential to first consider the normal function and development of trophoblastic cells. Following fertilisation, the zygote undergoes sequential cell divisions before forming the blastocyst, with cells of the outer trophectoderm lineage giving rise to trophoblastic cells responsible for implantation and development of the placental structures (Knöfler et al., 2019). Following implantation, trophoblastic cells differentiate into specialised trophoblast lineages:

- Cytotrophoblasts: proliferative progenitor cells that provide the cellular foundation for trophoblast growth and differentiation.

- Syncytiotrophoblasts: highly differentiated multinucleated cells involved in maternal–fetal exchange and hormone production, including secretion of hCG to support pregnancy.

A third population, the intermediate/ extravillous trophoblast lineage, contributes to placental anchoring, invasion into maternal tissue and remodelling of uterine spiral arteries (Knöfler et al., 2019). This balance between proliferation, differentiation and controlled invasion is tightly regulated by genetic, epigenetic and molecular signalling pathways. Dysregulation of these processes contributes to the abnormal trophoblastic proliferation characteristic of GTD (Hui et al., 2017; Fisher and Maher, 2021).

1.3.2 Genetic Basis of Hydatidiform Moles

The genetic abnormalities associated with hydatidiform moles (HM) provide a foundation for understanding GTD. HMs are classified into two distinct types based on genetic origin and pathological features: complete hydatidiform mole (CHM) and partial hydatidiform mole (PHM) (Slim, 2024).

Genetically, CHMs are diploid and characterised by the absence of maternal DNA and exclusive paternal genomic contribution, with approximately 90% demonstrating a 46XX karyotype, and the remaining cases showing a 46XY composition (Fisher and Maher, 2021; Slim,2024). CHMs arise from fertilisation of an anucleate ovum, most commonly by a single sperm that duplicates its chromosomes (monospermic origin). although fertilisation by two sperm (dispermic origin) can also occur (Fisher and Maher, 2021). Monospermic CHMs typically exhibit 46XX karyotypes, while the dispermic cases may result in either 46XX or 46XY. The absence of maternal genomic contribution results in dysregulated trophoblastic proliferation and absence of embryonic development (Kaur, 2024). The mechanism by which the maternal chromosomes is lost is still not fully understood with several proposed mechanisms put forward, including during meiosis, and post-zygotic diploidisation of a triploid conceptus (Fisher and Maher, 2021).

PHMs are triploid, containing one maternal set of chromosomes and two paternal sets, with the most common karyotypes including 69XXX, 69XXY, and 69XYY. This triploidy typically results from dispermic fertilisation of a normal ovum, although rarely may arise from duplication of a single sperm genome (Fisher and Maher, 2021). PHMs may contain embryonic or fetal tissue but are invariably non-viable pregnancies.

A key feature underpinning these genetic abnormalities is disrupted genomic imprinting. In normal placentation, maternally expressed genes act to constrain trophoblastic growth, whereas paternally expressed genes promote proliferation. In CHM, the absence of maternal genomic contribution results in unopposed expression of paternal growth-promoting genes and loss of maternally expressed tumour suppressors, leading to excessive trophoblastic proliferation (Candelier, 2016; Hui et al., 2017)). This is exemplified by loss of expression of the maternally derived cyclin-dependent kinase inhibitor p57^Kip2^ (CDKN1C), which is absent in CHM but retained in PHM (Buza and Hui, 2014).

1.3.3 Molecular Mechanisms of Trophoblastic Proliferation

At the cellular signalling level, hydatidiform moles exhibit dysregulation of pathways that control trophoblast proliferation, differentiation, and invasion. One key pathway involved in normal trophoblast biology is Wnt/β-catenin signalling, which promotes trophoblast differentiation and invasive behaviours (Knöfler and Pollheimer, 2013). Alterations in β-catenin signalling and cell proliferation markers have been reported in gestational trophoblastic lesions, with altered β-catenin localisation and increased Ki-67 expression observed in choriocarcinoma and its premalignant forms compared with normal placental tissue (Wong et al., 2006). These alterations reflect disruption of the normal balance between trophoblastic proliferation and differentiation that characterises GTD.

Regulation of trophoblast invasion is also influenced by transforming growth factor-beta (TGF-β) signalling pathways. In normal trophoblast development, TGF-β signalling contributes to the regulation of trophoblast invasion through downstream pathways including Smad signalling and modulation of extracellular matrix remodelling. Altered expression of TGF-β signalling components, including loss of Smad3 expression, has been identified in malignant trophoblastic cells, suggesting disruption of pathways normally involved in controlling trophoblast behaviour (Xu et al., 2001). Gene expression studies of trophoblast invasion models have further demonstrated the complex regulation of invasion-associated pathways, including factors involved in extracellular matrix remodelling and cellular migration (Bilban et al., 2010).

Beyond individual signalling pathways, hydatidiform moles demonstrate widespread epigenetic dysregulation resulting from abnormal parental genomic contribution. The androgenetic origin of complete hydatidiform moles leads to altered expression of imprinted genes involved in placental development, trophoblast proliferation and differentiation. Aberrant epigenetic

regulation, including altered DNA methylation and dysregulated trophoblast-associated gene expression, contributes to the abnormal proliferative phenotype observed in GTD (Hui et al., 2017; Szabolcsi et al., 2021).

Additionally, hCG and its molecular variants play an important role in GTD biology. Complete hydatidiform moles frequently produce markedly elevated concentrations of hCG, reflecting increased trophoblastic mass and syncytiotrophoblast activity. Hyperglycosylated hCG (HhCG), produced predominantly by invasive trophoblast populations, has been proposed to act as an autocrine regulator promoting trophoblast invasion and growth (Cole, 2012). The combination of abnormal genomic imprinting, altered signalling pathways and changes in trophoblast-associated hormones results in uncontrolled trophoblastic proliferation and provides the biological basis for progression from molar pregnancy to GTN.

1.3.4 High Cure Rates and Unique Malignancy

GTD occupies a unique place in oncology as a highly curable "model malignancy" (Li et al., 1958; Seckl et al., 2010). The reason lies partly in its pathobiology. Choriocarcinoma and persistent GTN, when they arise, retain sensitivity to chemotherapy agents such as MTX and ActD which is thought to relate partly to the biological characteristics of trophoblastic cells, including high proliferative activity and retained sensitivity to apoptosis (Goldstein and Berkowitz, 2012; Winter, 2021). Moreover, because GTN cells secrete hCG, they can be monitored closely – relapse or residual disease is detected at a microscopic level via rising hCG, allowing for early intervention before bulk tumour burden becomes large (Seckl et al., 2010; McMahon et al., 2024). The development of effective chemotherapy regimens, as discussed earlier, has turned GTN from a fatal disease into a curable one in most patients. This creates an imperative to refine treatment so that all patients are cured while minimising exposure to toxic chemotherapy in those who might not need it (Parker et al., 2017).

1.4 Biomarkers and hCG Assay Technology

Biomarkers play a significant role in the diagnosis, monitoring, and management of GTD. hCG is the primary GTD biomarker allowing for disease detection, monitoring, and management (Seckl et al., 2010; McMahon et al., 2024). Its production by trophoblastic cells and role in early pregnancy maintenance are well-documented, making it invaluable in the clinical setting (Cole,

2010). In GTD, the dysregulated production of hCG directly reflects tumour activity, providing a real-time measure of disease progression or remission (McMahon et al., 2024).

1.4.1 hCG as the Central Biomarker

Human chorionic gonadotropin is the cornerstone biomarker in GTD diagnosis and management. It is a heterodimeric glycoprotein hormone comprising an alpha (α) and beta (β) subunit held together by charge and hydrophobic interactions. The α-subunit is shared with other glycoproteins, including luteinising hormone (LH), follicle-stimulating hormone (FSH), and thyroid-stimulating hormone (TSH), while the β-subunit confers specificity to hCG (Cole, 2010). hCG is primarily produced by syncytiotrophoblasts, and its secretion is essential for corpus luteum maintenance and progesterone synthesis during early pregnancy (Cole, 2010). The hCG a-subunit is composed of 92 amino acids and the specific ß-subunit contains 145 amino acids. The hCG glycoprotein contains 8 sugar side chains which can account for up to 40% of the molecular weight of the molecule. The mixture of free and degraded subunits, and the various N – and O – linked oligosaccharide side chains cause significant heterogeneity in hCG structure (Cole, 2012: McMahon et al., 2024).

A combination of intact hCG, free subunits, degraded molecules, molecules with hyperglycosylated N- and O-linked oligosaccharide side chains, and fragments are present in serum, urine and other bodily fluids. These can be detected in normal pregnancy, trophoblastic disease, testicular germ cell malignancies and in some non-trophoblastic neoplasms (Cole and Kohorn, 2006; Cole, 2012: McMahon et al., 2024).

In addition to intact hCG, serum samples contain several important structural variants including free β-subunit hCG, nicked hCG, hCG missing the β-subunit C-terminal peptide and hyperglycosylated hCG. There are also several less common variants: nicked free β-subunit, hyperglycosylated free β-subunit, free β-subunit missing the β-subunit C-terminal peptide and nicked free β-subunit lacking the β-subunit C-terminal peptide (Cole et al., 2012). In urine samples, the same mix of molecules is present plus their terminal breakdown product, the β-core fragment (Cole, 2012; McMahon et al., 2024).

1.4.2 Analytical Platforms for hCG Measurement

Accurate and reproducible measurement of hCG is essential for the clinical management of GTD, where serial changes in concentration are used to identify disease regression, persistence, and

treatment response. Unlike many tumour markers, interpretation of hCG relies not only on absolute concentration but also on longitudinal changes over time, making analytical consistency particularly important (Mitchell et al., 2006).

The biochemical complexity of hCG presents specific analytical challenges. Multiple molecular forms may be present in circulation, including intact hCG, free β-subunit, nicked hCG, hyperglycosylated hCG, and other degradation products. The relative abundance of these variants differs between normal pregnancy and trophoblastic disease, meaning individual immunoassay platforms may produce differing results depending on antibody specificity and epitope recognition (Cole and Kohorn, 2006; Cole, 2012).

Selection of an appropriate assay for GTD monitoring therefore requires broad recognition of clinically relevant hCG isoforms. Assays developed primarily for pregnancy detection may not adequately detect all hCG variants produced by trophoblastic tumours, creating potential risks of underestimation during disease surveillance (Cole and Sutton, 2004). Specialist GTD centres therefore rely on validated methods with demonstrated suitability for longitudinal monitoring.

At Sheffield Teaching Hospitals, hCG monitoring is performed using established assay platforms selected for consistency and detection of clinically relevant hCG variants. Maintaining analytical continuity is particularly important because clinical decisions are based on sequential hCG kinetics rather than isolated measurements. Differences between assay platforms must therefore be considered when comparing results between centres or interpreting historical datasets (Cole and Kohorn, 2006).

1.4.3 Sheffield In-House Radioimmunoassay

The Sheffield hCG assay is a competitive radioimmunoassay (RIA) in which endogenous hCG competes with a radiolabelled tracer for a finite pool of polyclonal anti-hCG antibodies. Signal intensity, measured on a gamma-counter, is therefore inversely proportional to the hCG concentration in the patient sample (McMahon et al., 2024). Calibration is performed with a seven-point master curve traceable to the fifth WHO International Standard (07/364) and is re-verified quarterly with international reference reagents. Urine is the routine matrix, but serum aliquots are analysed whenever heterophilic-antibody interference is suspected ((McMahon et al., 2024); Sheffield Teaching Hospitals, 2023a). The functional detection limit is approximately 2 IU/L, and linearity is maintained up to 180 IU/L, after which dilutions extend the reportable range.

Because the antibody repertoire is polyclonal and targets multiple epitopes across the α–β junction and β-subunit carboxyl terminus, the Sheffield RIA recognises intact hCG, hyper-glycosylated hCG and β-core fragments alike, an advantage during early post-evacuation surveillance when trophoblastic disease when trophoblastic disease may produce varying proportions of different hCG isoforms (Cole and Sutton, 2004; Cole and Kohorn, 2006; McMahon et al., 2024). Daily three-level internal quality controls achieve coefficients of variation below 6 %, and participation in the UK National External Quality Assessment Scheme (UK NEQAS) ensures peer harmonisation.

Nevertheless, the manual nature of tube addition, incubation, and gamma-counting constrains throughput and necessitates strict radiation-safety governance. In addition, comparative studies show the RIA reads 5–10 % higher than IMMULITE in the low-positive region (< 50 IU/L), an analytical bias that must be accounted for when constructing kinetic nomograms (Sheffield Teaching Hospitals, 2023b).

1.4.4 Siemens IMMULITE 2000 chemiluminescent immunoassay

The IMMULITE 2000 is a two-site sandwich immunoassay performed on paramagnetic beads. Monoclonal antibodies specific for the β-subunit carboxyl-terminal peptide and for the α–β junction capture intact hCG and a range of clinically relevant hCG variant molecules; a bead-bound alkaline-phosphatase label then hydrolyses a dioxetane substrate to generate a chemiluminescent signal that is directly proportional to hCG concentration (Siemens Healthcare Diagnostics Inc., 2025; Cole and Sutton, 2004).

The analyser can process up to 200 tests per hour in continuous random-access mode, with a five-hour walk-away capacity when fully loaded. Analytical sensitivity is 0.2 IU/L and linearity extends to 5 000 IU/L, after which the instrument performs automatic 1:16 dilutions, keeping results accurate to about 80 000 IU/L. Calibration is also traceable to the fifth WHO International Standard (07/364) and remains valid for fourteen days or two thousand determinations. Total imprecision is typically below 4 % across the working range (Siemens Healthcare Diagnostics Inc., 2025).

Proprietary heterophilic-antibody blockers reduce false-positive “phantom-hCG” readings, although rare discordant samples continue to mandate serum-versus-urine cross-checks (Cole

and Kohorn, 2006; McMahon et al., 2024). Expert reviews have identified IMMULITE as an appropriate automated platform for GTD follow-up where specialist RIA facilities are unavailable, due to its broad recognition of clinically relevant hCG isoforms (Cole and Sutton, 2004; McMahon et al., 2024).

1.4.5 Global hCG Immunoassays

Human chorionic gonadotrophin is measured worldwide for GTD surveillance predominantly using high-throughput, automated immunoassay platforms, yet there is no true international consensus on which is the optimal assay for monitoring trophoblastic disease. Commonly used commercial systems include platforms from Abbott, Roche, Beckman Coulter, Ortho and Siemens, each using different antibody combinations and assay designs (Cole and Sutton, 2004; Franks et al., 2023).

Although modern hCG assays demonstrate excellent analytical performance for routine pregnancy testing, GTD monitoring presents additional challenges due to the molecular heterogeneity of hCG. Circulating hCG may include intact hCG, free β-subunit, nicked hCG, hyperglycosylated hCG and β-core fragments, with the relative abundance of these forms differing according to disease state. Differences in antibody specificity and epitope recognition mean that recovery of these variants can vary between manufacturers (Cole and Kohorn, 2006; Stenman, 2004).

International standardisation has improved comparability for intact hCG measurement; however, harmonisation remains incomplete because modified hCG variants are not equally recognised across platforms. Consequently, clinically relevant differences may occur when results generated by different assays are compared, particularly during longitudinal monitoring of GTD where small changes in hCG concentration influence clinical interpretation (Grenache et al., 2010; Franks et al., 2023).

These analytical differences highlight the importance of maintaining consistency of assay methodology during GTD follow-up. Switching between platforms, or combining datasets generated using different analytical systems, may introduce assay-dependent variation unrelated to biological change. Therefore, predictive modelling of serial hCG kinetics requires consideration of assay methodology, particularly when comparing cohorts from different centres or healthcare systems (Cole and Sutton, 2004; McMahon et al., 2024).

1.4.6 Immunoassay Interference

All immunoassays are subject to interference that can yield false results. In GTD management, two issues are particularly notable. One is heterophilic antibodies – circulating human anti-animal antibodies that can cross-link assay antibodies and produce a spurious signal. This can cause a persistent low-level positive hCG ("phantom hCG") even when the patient is disease-free (Cole and Kohorn, 2006; ACOG, 2002). If a woman has an unexpected low hCG that doesn't fit clinically, labs will often test a urine hCG sample (as heterophilic IgG antibodies are not excreted in urine), repeat testing using an alternative assay platform, or analyse the sample following addition of heterophilic antibody blocking reagents (ACOG, 2002; McMahon et al., 2024).

The other issue is the high-dose hook effect, which occurs in one-step sandwich assays when hCG levels are extremely high, causing saturation of capture and signal antibodies and resulting in a paradoxically low measurement. In GTN, very high hCG concentrations can occasionally occur, creating the potential for falsely reduced results if assay limitations are not recognised (Grenache et al., 2010). Most modern analysers incorporate mechanisms to detect or reduce this risk, including automated dilution protocols, but vigilance remains required. Dilution of the sample will typically recover the true elevated concentration if a hook effect is present.

Awareness of these technical caveats is critical in GTD, so that therapy decisions are based on accurate data. In practice, any hCG result that doesn't match the clinical scenario (e.g. plateau in a patient doing well, or persistently low-level hCG with no evidence of disease) should prompt confirmatory steps with alternate methods (Joyce et al., 2022; McMahon et al., 2024).

1.4.7 Clinical Utility of hCG in GTD

In GTD management, hCG is used for diagnosis, monitoring, and prognostication. At diagnosis, an elevated hCG concentration after a pregnancy event differentiates GTD from other causes of bleeding (Seckl et al., 2010; Tidy et al., 2021). For instance, in a woman with miscarriage, an hCG level plateau or rise instead of falling suggests retained molar tissue or GTN (Tidy et al., 2021).

The International Federation of Gynaecology and Obstetrics (FIGO) has established criteria for diagnosing post-molar GTN based on hCG dynamics: a plateau of hCG (±10% change) across 4 measurements over 3 weeks, or a >10% rise in hCG over 3 measurements in 2 weeks, or

persistence of any hCG >6 months post-evacuation, or histologic diagnosis of choriocarcinoma. Meeting any of these criteria indicates that active disease remains, and treatment should be considered (Ngan et al., 2018; Ngan et al., 2025). In the UK, treatment decisions are not based solely on FIGO criteria but are guided by centralised specialist centre protocols, including those developed at the Sheffield Trophoblastic Disease Centre.

Once treatment begins, serial hCG measurements (often weekly during chemotherapy) are the primary gauge of response. A normal hCG is the definition of remission, and protocols usually mandate additional consolidation cycles of chemotherapy after the first normal hCG to ensure eradication of residual cells (typically 1–3 extra cycles) (Tidy et al., 2021). Following completion of therapy, hCG monitoring continues for a defined surveillance period to enable early detection of recurrence (Tidy et al., 2021).

Baseline hCG levels also carry prognostic information. The FIGO/WHO scoring system for GTN incorporates the pre-treatment hCG into the overall prognostic risk score (Ngan et al., 2018). This scoring system (see Table 1.1) assigns weighted values to key clinical and biochemical variables to stratify patients with GTN into low-risk (score ≤6) and high-risk (score ≥7) categories, thereby guiding selection of single-agent versus multi-agent chemotherapy. While widely adopted, the FIGO/WHO scoring system provides a population-based risk estimate and may not fully capture individual variation in treatment response, highlighting the need for more dynamic, biomarker-driven prediction approaches.

Factor	**Score 0**	**Score 1**	**Score 2**	**Score 4**
Age	<40	≥40	–	–
Antecedent pregnancy	Mole	Abortion	Term	–
Interval (months)	<4	4–6	7–12	>12
hCG (IU/L)	$<10^3$	10^3–10^4	10^4–10^5	$>10^5$
Tumour size	<3 cm	3–5 cm	>5 cm	–
Metastases site	Lung	Spleen/kidney	GI	Brain/liver
No. metastases	0	1–4	5–8	>8
Previous chemotherapy	None	–	Single drug	≥2 drugs

Table 1.1: FIGO/WHO Prognostic Scoring System for GTN, used for risk stratification and treatment allocation. Source: Adapted from International Federation of Gynaecology and Obstetrics (2002) and updated guidance (Ngan et al., 2018).

An extremely high initial hCG is associated with increased tumour burden, greater risk of metastasis, and higher likelihood of single-agent chemotherapy failure (FIGO Oncology Committee, 2002; Ngan et al., 2018). However, once therapy is appropriately stratified, even very high hCG cases can be cured – so hCG mainly helps in assigning intensity of treatment rather than predicting ultimate survival (which is high across the board) (Lurain, 2011).

In summary, hCG is an indispensable biomarker in GTD. It provides a tumour-specific, quantitative measure that guides every phase of management, from diagnosing persistent disease, to choosing initial therapy, to monitoring remission and detecting relapse. Few solid tumours possess such a sensitive and clinically informative circulating biomarker, which is a major contributor to the favourable outcomes achieved in GTN (Bagshawe, 1992). Ongoing research is examining whether incorporation of hCG kinetics, molecular biomarkers and predictive modelling approaches can further improve personalised risk assessment and earlier identification of treatment resistance.

1.5 Hydatidiform Moles and Low-Risk Gestational Trophoblastic Disease

Hydatidiform moles represent the most common form of gestational trophoblastic disease and include complete and partial hydatidiform moles. Although these lesions arise from abnormal trophoblastic proliferation, the majority follow a benign clinical course and demonstrate spontaneous remission following uterine evacuation. Consequently, most patients are managed conservatively through structured hCG surveillance rather than immediate chemotherapy (Seckl et al., 2010; Tidy et al., 2021).

While most hydatidiform moles resolve without further intervention, a small proportion develop persistent gestational trophoblastic neoplasia (GTN), requiring chemotherapy. The risk of progression differs between complete and partial moles, highlighting the importance of accurate diagnosis, classification and risk-adapted follow-up (Berkowitz and Goldstein, 2013; Ngan et al., 2018).

1.5.1 Pathophysiology and Molecular Mechanisms

Hydatidiform moles develop following specific abnormalities in fertilisation, leading to trophoblastic overgrowth and atypical placental morphology. These changes arise from underlying genetic and epigenetic mechanisms that distinguish complete from partial moles (Hui et al., 2017; Fisher and Maher, 2023).

1.5.1.1 Genetic and Epigenetic Abnormalities

Hydatidiform moles arise from abnormal fertilisation leading to an imbalance of paternal genetic material and disordered genomic imprinting, as outlined in section 1.3.2 (Fisher and Maher, 2021).

The overrepresentation of the paternal genome is a fundamental event that leads to widespread aberrations in genomic imprinting in trophoblastic tissue (Hui et al., 2017; Fisher and Maher, 2021). Normally, imprinted genes from the maternal genome constrain placental growth, while paternally expressed genes promote trophoblastic proliferation. In CHM the lack of any maternal contribution causes unchecked expression of paternally derived growth-promoting genes and silencing of maternal tumour suppressors (Candelier, 2016). As a result, villous trophoblasts proliferate excessively and develop hydropic (oedematous) villi with no viable embryo.

A key imprinting effect is loss of expression of the cell-cycle inhibitor p57Kip2 (CDKN1C), which is maternally expressed; CHM therefore lack p57 protein expression, whereas PHM, which have a maternal genome, retain p57 expression (Hui et al., 2017). These genetic differences underlie the distinct histological and clinical features of complete vs. partial moles (Hui et al., 2017; Fisher and Maher, 2021).

Rare familial recurrent moles with biparental genomes, due to maternal-effect mutations in NLRP7 or KHDC3L, also demonstrate global imprinting failure, highlighting that disrupted epigenetic reprogramming of the maternal genome can lead to the same molar phenotype (Murdoch et al., 2006; Nguyen and Slim, 2014; Savage et al., 2020). Androgenetic diploidy in CHM and diandric triploidy in PHM both cause an imbalance in imprinted gene expression, tipping the balance toward trophoblastic overgrowth and placental tumourigenesis (Hui et al., 2017).

1.5.1.2 Trophoblast Proliferation and Signalling Pathways

As outlined in Section 1.3.3, abnormal trophoblastic proliferation in GTD is driven by disruption of signalling pathways that regulate placental growth, differentiation and invasion. In hydatidiform moles, these molecular abnormalities occur because of abnormal parental genomic contribution and imprinting defects, resulting in excessive villous trophoblast proliferation and abnormal placental development (Hui et al., 2017; Fisher and Maher, 2021).

The predominance of paternal genomic influence promotes a trophoblastic phenotype characterised by enhanced proliferative capacity, altered differentiation and abnormal regulation of invasion. Pathways involved in normal placental development, including Wnt/β-catenin signalling, TGF-β-mediated regulation of invasion and growth-factor signalling pathways, have been implicated in these processes (Xu et al., 2001; Knöfler and Pollheimer, 2013; Fock et al., 2015).

A clinically important consequence of trophoblast expansion is increased production of hCG, particularly from proliferating syncytiotrophoblast. The relationship between trophoblast mass and hCG secretion forms the biological basis for using serial hCG measurements to monitor disease regression or persistence following evacuation (Cole, 2012; Tidy et al., 2021). Therefore, the molecular abnormalities driving hydatidiform mole formation directly underpin the biomarker strategies used in modern GTD surveillance and provide the rationale for modelling hCG kinetics as a predictor of disease behaviour.

1.5.2 Clinical Presentation and Symptoms

Hydatidiform moles often present with signs and symptoms that reflect an abnormal early pregnancy. The clinical features differ in severity between complete and partial moles, but there is overlap. Historically, many patients presented in the late first or early second trimester with classical clinical features; however, the widespread use of early pregnancy ultrasound has resulted in earlier detection, with many cases now diagnosed before significant symptoms develop (Seckl et al., 2010; Tidy et al., 2021; Ngan et al., 2025). Below are the common clinical manifestations.

1.5.2.1 Vaginal Bleeding

Irregular or heavy vaginal bleeding in early pregnancy is the most frequent presenting symptom of hydatidiform mole (Seckl et al., 2010; Tidy et al., 2021). In a complete mole, bleeding is often intermittent and dark brown due to retained blood associated with abnormal villous development although classical presentations are now less frequently observed due to earlier ultrasound detection (Seckl et al., 2010; Tidy et al., 2021). Patients may occasionally pass vesicular tissue per vaginam, described as “grape-like” clusters of hydropic villi, which is highly suggestive of molar pregnancy when present (Seckl et al., 2010).

Partial moles may cause vaginal bleeding, but it may be initially interpreted as non-molar miscarriage. Consequently, many partial moles are recognised only following histopathological examination of products of conception after uterine evacuation (Sebire et al., 2003; Tidy et al., 2021).

1.5.2.2 Excessive Uterine Growth

Historically, discrepancy between uterine size and gestational age was a recognised clinical feature of hydatidiform mole, particularly in complete moles (Berkowitz and Goldstein, 2009; Lurain, 2010). In complete moles, excessive trophoblastic proliferation and hydropic swelling of chorionic villi can result in uterine enlargement beyond that expected for gestational age (Lurain, 2010; Hui et al., 2017).

However, the frequency of classical presentations has declined with the widespread use of early pregnancy ultrasound, which enables diagnosis before the development of advanced clinical features (Berkowitz and Goldstein, 2009). In contrast, partial moles are less commonly associated with marked uterine enlargement and may present similarly to non-molar pregnancy loss due to the presence of abnormal embryonic or fetal tissue, contributing to diagnostic difficulty before histological examination (Sebire et al., 2003; Fisher and Maher, 2021).

1.5.2.3 Hyperemesis Gravidarum

The markedly elevated hCG levels observed in some molar pregnancies can contribute to severe nausea and vomiting (hyperemesis gravidarum), particularly in complete hydatidiform moles (Berkowitz and Goldstein, 2009; Lurain, 2010). This occurs because of excessive trophoblastic proliferation and increased syncytiotrophoblast-derived hCG production, with symptom severity often reflecting the degree of hormonal elevation (Hui et al., 2017; Cole, 2012).

Partial moles generally produce lower concentrations of hCG and are therefore less commonly associated with severe hyperemesis, although symptoms may overlap with those seen in normal early pregnancy or pregnancy loss (Lurain, 2010).

1.5.2.4 Early-Onset Preeclampsia

Preeclampsia (gestational hypertension with proteinuria or end-organ dysfunction) developing before 20 weeks' gestation is a red flag for molar pregnancy (Seckl et al., 2010; Ngan et al., 2025). In a normal gestation, preeclampsia almost never occurs so early; it classically arises after 20 weeks' gestation (Brown et al., 2018). Thus, the appearance of high blood pressure, proteinuria, or severe headache and oedema in the late first or early second trimester should prompt evaluation for GTD (Ngan et al., 2025; Lurain, 2010). Partial moles rarely progress far enough to cause preeclampsia, and if they do, it tends to be milder (Lurain, 2010).

1.5.2.5 Theca Lutein Ovarian Cysts

The ovarian theca cells often become hyper stimulated by the extreme hCG levels in a complete mole, leading to bilateral ovarian theca lutein cysts in ~15–30% of cases (Berkowitz and Goldstein, 2009; Lurain, 2010). These cysts can be quite large (>6 cm) and multicystic (Lurain, 2010; Ngan et al., 2025). On examination or ultrasound, ovaries may appear enlarged with multiple thin-walled cystic structures due to this overstimulation (Berkowitz and Goldstein, 2009). Theca lutein cysts are usually asymptomatic, but in severe cases they can cause pelvic pain or ovarian torsion (Lurain, 2010). They typically regress after the mole is evacuated and hCG levels fall (Berkowitz and Goldstein, 2009). In partial moles, the hCG is usually not high enough to cause significant theca lutein cyst formation (Lurain, 2010).

1.5.2.6 Hyperthyroidism

Owing to molecular mimicry between hCG and TSH, very high hCG levels can activate the thyroid gland (Cole, 2009). Clinical hyperthyroidism (thyrotoxicosis) is observed in a small percentage (~1–5%) of patients with complete moles, especially those with extremely elevated hCG (Lurain, 2010; Ngan et al., 2025). Signs can include tachycardia, warm skin, tremors, and suppressed TSH levels (Lurain, 2010). This too is rarely seen in partial moles. Most often, if present, it occurs in advanced or untreated moles (hCG >100,000 IU/L), and it typically resolves after molar evacuation (Seckl et al., 2010; Lurain, 2010). Routine thyroid function testing is not necessary in

all molar pregnancies, but should be considered if the patient has symptoms of thyrotoxicosis or very high hCG (Tidy et al., 2021).

1.5.2.7 Absence of Fetal Activity (Complete Mole)

By definition, a complete mole has no fetus. Thus, there are no fetal heart tones detectable on Doppler, and no fetal parts palpable (Ngan et al., 2025; Hui et al., 2017). Often by the end of the first trimester, the absence of fetal heartbeat combined with other signs (bleeding, large uterus) may raise suspicion for a molar pregnancy (Seckl et al., 2010). In a partial mole, there may be a fetus or fetal tissue present, although it is typically abnormal and not viable (Hui et al., 2017; Fisher and Maher, 2021). In some cases of PHM, a fetal heartbeat might be detected early on, but the fetus is usually growth-restricted or has multiple anomalies and often dies in utero (Hui et al., 2017). The presence of a co-twin can complicate this picture (rarely, a viable co-twin can coexist with a complete mole in a twin pregnancy, but that is a separate scenario) (Vejerslev, 1991).

1.5.3 Diagnosis and Classification

Accurate diagnosis and classification of hydatidiform moles are fundamental to ensuring appropriate clinical management, surveillance, and prognostication. The distinction between complete and partial moles relies on a combination of morphological, immunohistochemical, and genetic features (Tidy et al., 2021). While many cases can be recognised by characteristic histopathological appearances, early gestational specimens or atypical presentations may require ancillary investigations to confirm the diagnosis and guide risk stratification. This section outlines the key diagnostic modalities used in current clinical practice and the distinguishing features of each mole subtype (Fisher and Maher, 2021).

1.5.3.1 Histopathology and Morphology

Definitive diagnosis of a hydatidiform mole is made by histopathologic examination of uterine contents after evacuation (Kaur, 2024). Complete and partial moles have distinct microscopic features as discussed.

1.5.3.1.1 Complete Hydatidiform Mole

The morphological features of CHM reflect the underlying genetic mechanisms described in Section 1.3.2. On gross examination, a CHM often appears as a cluster of translucent vesicles (“bunch of grapes”) due to swelling of chorionic villi (Kaur, 2024). Microscopy shows diffuse

hydropic degeneration of virtually all villi: chorionic villi are enlarged, oedematous, and avascular, with a central cistern of fluid in larger villi (Hui et al., 2017; Kaur, 2024). There is circumferential trophoblastic proliferation around villi, with both cytotrophoblast and syncytiotrophoblast cells proliferation, often with atypia (Hui et al., 2017). Trophoblastic inclusions and scalloping of villi may be present (Fisher and Maher, 2021). No embryonic or fetal tissues are present, which is a key distinguishing feature (Genest and Berkowitz, 2022). Because all genetic material is paternal, p57 (a maternally expressed cell-cycle inhibitor) is absent in the villous cytotrophoblast and stromal cells of a complete mole (Buza and Hui, 2014). Histologically, the absence of p57 staining together with the diffuse pattern of villous oedema and trophoblastic hyperplasia confirms the diagnosis of CHM (Buza and Hui, 2014; Kaur, 2024).

1.5.3.1.2 Partial Hydatidiform Mole

The morphological features of PHM similarly reflect the underlying triploid genetic origin described in Section 1.3.2. Grossly, a partial mole may show fewer and smaller grape-like villi, often mixed with normal placental tissue (Banet et al., 2014; Kaur, 2024). Microscopic exam reveals a mixture of hydropic and normal villi. Some chorionic villi are enlarged and oedematous, but others are normal-sized or fibrotic (Banet et al., 2014). The villi often show irregular shapes with scalloped borders and trophoblastic inclusions (Banet et al., 2014). Fetal or embryonic tissue is frequently present in a partial mole – one may see nucleated fetal red blood cells in villous capillaries, identifiable fetal parts, or even a formed but nonviable fetus if the pregnancy progressed enough (Banet et al., 2014; Kaur, 2024). Trophoblastic proliferation in PHM is focal and mild: there are regions of villi with syncytiotrophoblastic hyperplasia (often described as “syncytial knots”), but it is not circumferential and severe as in CHM (Banet et al., 2014). Cytologic atypia of trophoblast is minimal. Importantly, p57 immunostaining is positive in partial moles (because the maternal genome is present), with p57 expression in villous stromal and cytotrophoblast cells similar to a normal placenta (Buza and Hui, 2014; Banet et al., 2014). This contrasts sharply with complete moles. The combination of p57-positive immunohistochemistry and identification of fetal tissue or only focal villous oedema is diagnostic of a partial mole and essentially excludes a complete mole (Buza and Hui, 2014; Fisher and Maher, 2021).

Diagnosing early first-trimester moles can be challenging based on morphology alone, as the villous changes may be subtle. There is known inter-observer variability among pathologists in distinguishing partial mole from hydropic (non-molar) miscarriage in early samples (Banet et al., 2014). Thus, ancillary techniques are important to confirm the diagnosis in equivocal cases.

Immunohistochemistry for p57 is a widely used adjunct: a negative p57 strongly supports complete mole, whereas a positive p57 favours partial mole or a non-molar hydropic abortus (Buza and Hui, 2014; Banet et al., 2014). This helps avoid misclassification that could lead to inappropriate management. In difficult cases, molecular genotyping is considered the diagnostic gold standard for classification (Murphy et al., 2009; Fisher and Maher, 2021).

1.5.3.2 Genetic Testing (Molecular Classification)

DNA-based analysis of molar tissue can definitively determine ploidy and parental origin, thereby confirming the genetic mechanisms described in Section 1.3.2 (Banet et al., 2014; Fisher and Maher, 2021).

Techniques such as short tandem repeat genotyping or single-nucleotide polymorphism arrays can identify androgenetic diploidy (indicative of CHM), diandric triploidy (PHM), and biparental diploidy (consistent with a non-molar miscarriage) (Murphy et al., 2009; Fisher and Maher, 2021). In practice, an algorithmic approach is used, integrating histopathology, p57 immunohistochemistry, and if results are discordant or ambiguous, PCR-based molecular genotyping (Banet et al., 2014). This combined approach of morphology, immunohistochemistry, and genotyping represents the current best practice for precise diagnosis of hydatidiform moles (Kaur, 2024; Fisher and Maher, 2021).

1.5.3.3 Imaging Findings

Ultrasonography is the key imaging modality for initial evaluation when a molar pregnancy is suspected (Ngan et al., 2018; Tidy et al., 2021). In a complete mole, ultrasound classically shows a diffuse echogenic intrauterine mass with innumerable cystic spaces, but no identifiable fetus. This is often described as a “snowstorm” or “cluster of grapes” appearance, reflecting the hydropic villi within the uterus (Kirk et al., 2007; Shaaban et al., 2017). The mass is heterogeneous and vesicular, filling the uterine cavity. There is absence of a fetus or amniotic fluid in a pure CHM. The uterus may appear enlarged on scan, and bilateral ovarian cysts (theca lutein cysts) may be visualised as adnexal masses due to ovarian hyperstimulation (Shaaban et al., 2017; Ngan et al., 2018). Colour Doppler ultrasound of a complete mole shows a hypervascular lesion with low-resistance, high-flow blood signals within the molar tissue, owing to the rich trophoblastic blood supply (Shaaban et al., 2017).

In the first trimester, however, a complete mole's ultrasound appearance can sometimes be mistaken for a missed miscarriage or blighted ovum, especially before the classic cystic changes become prominent (Kirk et al., 2007). Fewer than half of complete moles are definitively recognised by ultrasound in the first trimester, which is why correlation with hCG levels and clinical signs is important (Kirk et al., 2007), in keeping with UK national guidance which emphasises that ultrasound findings alone are insufficient for definitive diagnosis, requiring histopathological confirmation (Tidy et al., 2021).

A partial mole is more difficult to diagnose on imaging. Ultrasound may reveal an abnormally formed fetus or embryonic tissue coexisting with a thick, multicystic placenta (Kirk et al., 2007; Shaaban et al., 2017). Described findings include an enlarged placenta with focal cystic spaces – a "molar" appearance of the placenta – along with an amniotic sac that may contain a fetus or fetal parts (Shaaban et al., 2017). The fetus, if present, is often growth-restricted or malformed and may show oligohydramnios (Shaaban et al., 2017). In many cases of PHM, the ultrasound looks like a missed abortion with a disproportionally large, echogenic placenta (Kirk et al., 2007; Shaaban et al., 2017). One clue favouring partial mole over a simple missed abortion is the presence of abnormal cystic placental changes rather than normal trophoblastic appearances (Kirk et al., 2007). However, some partial moles can appear very similar to complete moles on ultrasound (if the fetus is small or already resorbed) or can appear like an uncomplicated miscarriage (Kirk et al., 2007).

Given these overlapping features, ultrasound alone is not diagnostic for PHM; many partial moles are only confirmed after evacuation and pathology (Kirk et al., 2007; Tidy et al., 2021). In practice, most partial moles are diagnosed retrospectively from histopathology after evacuation performed for presumed miscarriage (Tidy et al., 2021).

The diagnostic workup for a suspected molar pregnancy therefore integrates clinical assessment, ultrasound findings, and biochemical evaluation with quantitative hCG measurement, but definitive diagnosis relies on histopathological examination of uterine contents (Tidy et al., 2021). Accurate classification of complete versus partial mole is essential, as this determines the risk of progression to gestational trophoblastic neoplasia and guides subsequent follow-up and management strategies.

1.5.4 Management Strategies

Management of hydatidiform mole aims to safely remove molar tissue, preserve fertility where possible, and identify patients who develop persistent gestational trophoblastic neoplasia through structured surveillance. Unlike many premalignant conditions, initial management is usually conservative, with surgical evacuation followed by serial hCG monitoring rather than immediate systemic therapy, as most complete and partial moles resolve spontaneously after evacuation (Seckl et al., 2010; Tidy et al., 2021). The highly sensitive relationship between trophoblastic activity and circulating hCG allows early detection of persistent disease, enabling treatment to be initiated promptly in the small proportion of patients who progress to GTN (Ngan et al., 2018; Tidy et al., 2021). Management strategies therefore focus on three key areas: effective uterine evacuation, appropriate use of surgery in selected cases, and long-term biochemical surveillance.

1.5.4.1 Uterine Evacuation

The primary treatment for a hydatidiform mole is prompt uterine evacuation of the molar tissue. The method of choice is suction curettage, which effectively removes the abnormal trophoblastic tissue while preserving fertility and minimising subsequent treatment requirements (Tidy et al., 2000; Tidy et al., 2021). Guidelines recommend using suction evacuation, performed by an experienced clinician, with ultrasound guidance considered where appropriate to reduce the risk of incomplete evacuation or complications (Tidy et al., 2021; Ngan et al., 2025). The procedure is generally performed under anaesthesia. Given the increased vascularity of molar tissue, evacuation carries a risk of haemorrhage, particularly in patients with enlarged uteri or high volume disease, and appropriate facilities for managing bleeding should therefore be available (Ngan et al., 2025; Tidy et al., 2021). Rh immunoglobulin is administered at the time of evacuation to prevent Rh sensitisation, in line with guidance for early pregnancy events. Anti-D prophylaxis is recommended in practice, particularly where a partial molar pregnancy cannot be excluded at the time of evacuation (Tidy et al., 2021).

1.5.4.2 Hysterectomy

In certain cases, hysterectomy may be considered for managing a hydatidiform mole. This is usually reserved for women who have completed childbearing or where fertility preservation is not required, particularly in older patients who have an increased risk of developing persistent GTN (Tidy et al., 2021; Ngan et al., 2025). Performing a hysterectomy removes the molar tissue

and eliminates the primary uterine site of trophoblastic proliferation, reducing the risk of local persistent disease (Tidy et al., 2000). For example, an older patient with a complete mole and no desire for future fertility might elect hysterectomy, although this approach is not routinely required because most molar pregnancies resolve following suction evacuation (Tidy et al., 2021).

However, continued hCG surveillance is still essential after hysterectomy because trophoblastic cells could have metastasised prior to removal of the uterus (Ngan et al., 2025). Hysterectomy therefore reduces but does not eliminate the need for future chemotherapy (Tidy et al., 2000; Sita-Lumsden et al., 2012). Similar findings have also been reported in European GTN cohorts, where hysterectomy was associated with reduced chemotherapy exposure and improved local disease control in selected patients with trophoblastic disease (Eysbouts et al., 2017).

In summary, suction evacuation remains the standard treatment for most, while hysterectomy represents an alternative for carefully selected patients or those who want definitive treatment at the cost of fertility. Other methods of uterine evacuation, including medical induction or hysterotomy, are generally avoided because of increased risks of incomplete evacuation, haemorrhage, and complications compared with suction evacuation (Tidy et al., 2021; Ngan et al., 2025).

1.5.4.3 Post-Evacuation hCG Monitoring

Careful follow-up after mole evacuation is critical, because hydatidiform moles can persist or turn into malignancy (GTN) even after removal (Seckl et al., 2010; Ngan et al., 2025). The cornerstone of follow-up is serial quantitative hCG measurements, since hCG is a sensitive tumour marker for residual trophoblastic tissue (Tidy et al., 2021; Ngan et al., 2025). After evacuation, serum hCG is measured weekly (or every 1–2 weeks) until it falls to within the normal non-pregnant range (Ngan et al., 2018; Tidy et al., 2021). The expectation is that hCG will decline exponentially after the mole is removed. For a complete mole, the median time to normalisation is around 8–14 weeks; for a partial mole it is often faster, within 8 weeks or less, because the initial levels are lower (Lurain, 2011; Eysbouts et al., 2017).

Once hCG has normalised, protocols historically recommended continued monitoring monthly for 6 months (for complete mole) or for a shorter duration for partial mole (Seckl et al., 2010). This surveillance is to ensure that any resurgence of hCG (indicating regrowth of trophoblastic tissue)

is detected early. During the surveillance period, pregnancy must be avoided as a new pregnancy would make interpretation of hCG surveillance difficult, so reliable contraception is advised (Tidy et al., 2021; Ngan et al., 2025). Oral contraceptives are commonly used and are safe in this context (Tidy et al., 2021).

It is worth noting that follow-up recommendations have updated recently. Recent evidence suggests that prolonged monitoring may not be necessary for all patients, especially those with partial moles given their low risk of recurrence (Schmitt et al., 2013; Albright et al., 2020). The 2018 FIGO guidelines and some national protocols (like UK guidelines) now tailor follow-up duration to mole type and hCG kinetics (Ngan et al., 2018; Tidy et al., 2021). For example, in the UK, after a partial mole, once hCG has normalised, patients just provide one more confirmatory normal sample 4 weeks later and then can be discharged from follow-up (Tidy et al., 2021). After a complete mole, if hCG normalises within 8 weeks post-evacuation, they follow for 6 months from the time of evacuation; if hCG takes longer than 8 weeks to normalise, they follow for 6 months from the first normal value (Tidy et al., 2021). These adjusted schedules aim to reduce unnecessary monitoring in low-risk cases while still catching the vast majority of persistent GTN, which usually declares itself within the first few months (Schmitt et al., 2013; Albright et al., 2020).

The key point is that persistent gestational trophoblastic disease almost always presents as a plateau or rise in hCG levels during follow-up. The criteria for diagnosing post-molar GTN (per FIGO) are: a plateau of hCG (±10% change) across 3 weeks, a >10% rise in hCG over 2 consecutive weeks, histologic diagnosis of choriocarcinoma, or persistence of detectable hCG beyond 6 months (although prolonged detectable hCG alone is no longer universally applied as an absolute indication for treatment in all modern protocols) (Ngan et al., 2018; Ngan et al., 2025). If any of these occur, it indicates that molar tissue remains and is proliferating, and chemotherapy is indicated.

1.5.5 Risk Factors for Persistent GTD

Certain clinical features present before or at the time of evacuation predict a higher likelihood that a mole will progress to invasive disease. Patients with these high-risk features may warrant closer monitoring or even prophylactic treatment (Goldstein and Berkowitz, 2012; Ngan et al., 2025).

Characteristics that approximately double the risk of post-molar persistence include:

- Extremes of maternal age: Age >35 (especially >40) or <20 years (Altieri et al., 2003; Ngan et al., 2025).
- Very high pre-evacuation hCG: >100,000 IU/L before evacuation (extremely elevated hCG correlates with large tumour burden) (Seckl et al., 2010; Goldstein and Berkowitz, 2012).
- Large uterine size or mass: Uterus larger than expected for dates, or a molar uterine mass > uterine half-size on imaging (Goldstein and Berkowitz, 2012; Ngan et al., 2025).
- Large theca lutein ovarian cysts: Ovarian cysts >6 cm in diameter (Goldstein and Berkowitz, 2012; Ngan et al., 2025).
- Clinical complications: Hyperemesis gravidarum, early preeclampsia, hyperthyroidism, or trophoblastic embolization to lungs/DIC – all signs of a very active mole (Seckl et al., 2010; Tidy et al., 2021).

A complete mole presenting with, for example, a 16-week size uterus, hCG of 500,000 IU/L, and large ovarian cysts would be considered high-risk for persistent disease. In contrast, a partial mole in a young patient with modest hCG has a very low risk. These factors are sometimes incorporated into scoring systems (like the WHO/FIGO prognostic score) if the patient eventually needs chemotherapy (Ngan et al., 2018). Clinically, knowing the risk factors helps identify who might benefit from prophylactic chemotherapy or intensified surveillance (Ngan et al., 2025).

Although historical studies have investigated prophylactic chemotherapy in patients considered at high risk of developing GTN, this approach is not routinely adopted within contemporary UK practice due to the effectiveness of structured hCG surveillance and the potential for unnecessary treatment in patients who would otherwise undergo spontaneous disease resolution (Wang et al., 2017; Tidy et al., 2021). Consequently, accurate identification of patients who will develop persistent disease remains an important clinical objective

1.5.5.1 Management of GTN following Diagnosis

Despite optimal initial management, about 10–20% of complete moles and ~1–5% of partial moles will progress to persistent GTD requiring chemotherapy (Seckl et al., 2010; Ngan et al., 2018). Once the criteria for GTN are met (plateauing or rising hCG after evacuation), the disease is classified and staged as GTN and treated accordingly (Ngan et al., 2018). Fortunately, virtually all cases of low-risk GTN (FIGO score 0–6, usually non-metastatic or limited mets) are highly curable with single-agent chemotherapy (Lurain, 2011; Tidy et al., 2021).

The two most commonly used regimens are methotrexate with folinic acid rescue or actinomycin-D as a single agent, given in pulses until hCG normalisation and then a couple additional cycles (Lurain, 2011; Sita-Lumsden et al., 2012). The initial remission rates are very high, and even those who do not respond to one agent are almost always cured by switching to the other (Ngan et al., 2018). Overall cure rates exceed 99% for low-risk GTN, with preservation of fertility in most cases (Sita-Lumsden et al., 2012; Tidy et al., 2021). Therefore, although progression to post-molar GTN requires chemotherapy, treatment outcomes are excellent when managed within specialist trophoblastic disease services (Sita-Lumsden et al., 2012; Tidy et al., 2021) .

1.5.6 Prognosis and Outcomes

Hydatidiform moles are associated with excellent clinical outcomes when identified early and managed through appropriate evacuation and structured hCG surveillance. Although a proportion of patients develop persistent trophoblastic disease requiring chemotherapy, the majority achieve spontaneous remission following removal of molar tissue. Long-term prognosis is largely determined by the risk of progression to gestational trophoblastic neoplasia (GTN), early identification through hCG monitoring, and timely initiation of treatment where required (Seckl et al., 2010; Tidy et al., 2021).

1.5.6.1 Spontaneous Resolution Rates

The majority of hydatidiform moles are benign and self-limited after appropriate treatment. Approximately 80–85% of complete moles regress completely after uterine evacuation, with hCG returning to normal and no further therapy needed (Seckl et al., 2010; Savage et al., 2013). Partial moles have an even higher rate of cure with just evacuation – about 95–99% of PHM cases do not require additional treatment beyond this (Savage et al., 2013; Tidy et al., 2021). In large series, only 10–20% of CHMs and 0.5–5% of PHMs progress to invasive GTD or choriocarcinoma that necessitates chemotherapy (Seckl et al., 2010; Savage et al., 2013). UK national cohort data similarly demonstrate persistent GTD rates of approximately 15% following complete mole and around 1% following partial mole (Savage et al., 2013; Tidy et al., 2021).

Thus, the prognosis of a molar pregnancy in terms of likelihood of malignancy is excellent, especially for partial mole. Even those moles that do become persistent are highly curable; low-risk GTN has ~99% cure rate (Sita-Lumsden et al., 2012; Tidy et al., 2021). Therefore, within

structured surveillance programmes, patients with molar pregnancy have excellent outcomes, with early identification and treatment of the small proportion who develop persistent disease.

1.5.6.2 Factors Influencing Progression to Persistent GTD

While any molar pregnancy can develop into GTN, certain factors increase that risk. As noted, complete moles carry a much higher risk of persistence than partial moles (on the order of a 10- to 30-fold difference) (Coyle et al., 2018). The purely paternal genomic constitution of CHM is thought to be a driver of its greater neoplastic potential – androgenetic cells may be more prone to abnormal proliferation (Hui et al., 2017; Fisher and Maher, 2021).

Within complete moles, those with high-risk clinical features (large uterine size, very high hCG, theca lutein cysts, etc.) have an elevated chance of requiring chemotherapy, often quoted around 30–50% if multiple risk factors are present (Goldstein and Berkowitz, 2012; Ngan et al., 2018). For example, a CHM with pre-evacuation hCG >100,000 IU/L has roughly double the risk of persistence compared to one with lower hCG (Sita-Lumsden et al., 2012; Ngan et al., 2018). Similarly, if a patient had preeclampsia or hyperthyroidism with the mole, it often signifies a heavy tumour burden and higher malignant potential (Lurain, 2011; Tidy et al., 2021).

Clinicians also use the post-evacuation hCG trajectory as an indicator of disease behaviour, as delayed normalisation or persistent elevation of hCG is strongly associated with the development of GTN requiring chemotherapy (Agarwal et al., 2012; Eysbouts et al., 2017).

1.5.6.3 Malignant Transformation and Metastasis

By definition, a hydatidiform mole is a pre-malignant condition. If it becomes invasive, it is then termed a persistent GTN (or invasive mole as it is a localised uterine tumour) or, if it differentiates into a frankly malignant phenotype with potential for widespread metastasis, choriocarcinoma (Ngan et al., 2018; Hancock and Tidy, 2021). The risk of progression to choriocarcinoma is quite low for moles – roughly 2–5% of complete moles and far less for partial moles (~0.1–0.5%) (Ngan et al., 2018; Tidy et al., 2021). Invasive mole is more common than metastatic choriocarcinoma. About 15% of CHMs need chemotherapy (mostly for persistent disease), whereas only a small subset of those are true choriocarcinomas (Coyle et al., 2018). The prognosis even in those cases is very good if appropriately treated, especially when identified early via hCG surveillance (Sita-Lumsden et al., 2012; Tidy et al., 2021).

1.5.6.4 Risk of Recurrence

Women who have had one molar pregnancy are at increased risk of having another in the future, although the absolute risk remains low. The chance of a second molar pregnancy is about 1% (1 in 100), which is roughly a tenfold increase over the baseline risk in the general obstetric population (Sebire et al., 2003; Eagles et al., 2015). This recurrence risk applies to both complete and partial moles, but interestingly the data suggest it is mainly driven by repeat complete moles (Sebire et al., 2003; Eagles et al., 2015).

If a patient has had two molar pregnancies, the risk of a third rises to as high as 10–20%. Such cases raise suspicion for an underlying genetic predisposition, such as recessive mutations in NLRP7 causing familial recurrent moles (Murdoch et al., 2006; Nguyen and Slim, 2014). In familial recurrent mole syndrome, the moles are often complete but with biparental genome due to maternal imprinting defects (Murdoch et al., 2006; Nguyen and Slim, 2014).

Because recurrence is possible, future pregnancies after a mole are managed with early ultrasound (6–8 weeks) and early hCG measurement (Ngan et al., 2018; Tidy et al., 2021). Pathology of the placenta in subsequent pregnancies is often advised to detect any unrecognised molar changes. Despite the increased risk, it bears emphasizing that 90–99% of women will not have a repeat mole and can expect normal outcomes (Sebire et al., 2003; Eagles et al., 2015).

1.5.6.5 Long-Term Fertility and Reproductive Outcomes

The long-term prognosis for fertility after a molar pregnancy is generally excellent, although this may be influenced by factors such as treatment intensity and patient age. Simply having had a mole does not usually impair a woman's ability to conceive or carry future pregnancies, provided her uterus is preserved (Sebire et al., 2003; Eagles et al., 2015).

Following completion of surveillance or treatment, reproductive outcomes are generally favourable, with most women achieving successful pregnancies and no significant increase in adverse obstetric outcomes compared with the general population (Blagden et al., 2002; Eagles et al., 2015). Even for the subset of patients who require chemotherapy for GTN, most can still have children afterward – the chemotherapy regimens used (methotrexate or actinomycin D, and even multi-agent regimens if needed) have a high success rate with generally minimal long-term effect on ovarian function in young women (Blagden et al., 2002; Lok et al., 2003).

Small increases in the risk of earlier menopause have been reported following multi-agent chemotherapy, particularly with combination regimens used for high-risk disease (Savage et al., 2015). However, single-agent chemotherapy used for low-risk GTN has not been associated with significant long-term impairment of fertility (Savage et al., 2015; Tidy et al., 2021).

Current guidelines advise patients to avoid conception until the recommended hCG surveillance period has been completed, to prevent a new pregnancy obscuring interpretation of hCG monitoring (Ngan et al., 2018; Tidy et al., 2021). Following completion of surveillance, patients can be reassured that future fertility outcomes are excellent, with the majority achieving normal pregnancies and requiring only early assessment in subsequent pregnancies because of the small increased recurrence risk (Sebire et al., 2003; Eagles et al., 2015).

1.5.7 Recent Advances and Research Directions

Advances in the understanding of gestational trophoblastic disease have increasingly shifted research from purely diagnostic classification towards improved prediction of disease behaviour and personalised patient management. Although current surveillance strategies based on serial hCG monitoring achieve excellent clinical outcomes, they require prolonged follow-up for large numbers of patients who will ultimately undergo spontaneous remission, while a smaller subset will progress to persistent GTD requiring chemotherapy (Agarwal et al., 2012; Coyle et al., 2018). Consequently, recent research has focused on identifying molecular, biochemical, and computational approaches that can improve early risk stratification, reduce unnecessary monitoring, and minimise treatment burden without compromising the high cure rates achieved in specialist GTD centres (Ngan et al., 2018; Tidy et al., 2021).

These developments include advances in understanding the molecular mechanisms underlying trophoblastic transformation, improved diagnostic classification through immunohistochemical and genomic techniques, evaluation of novel biomarkers, and the application of predictive models based on longitudinal hCG dynamics. Together, these approaches reflect a movement towards precision medicine in GTD, where management decisions may increasingly be guided by individual tumour biology and early disease behaviour rather than fixed population-based thresholds.

1.5.7.1 Recent Molecular Insights into GTD Pathogenesis

Ongoing research has expanded the understanding of molecular and epigenetic consequences of hydatidiform mole development and progression. While the fundamental genetic basis of complete and partial moles has been described in Sections 1.3.2 and 1.5.1.1, more recent studies have focused on the downstream molecular consequences of abnormal genomic imprinting and paternal genomic excess. Genome-wide methylation studies have identified dysregulation of numerous imprinted genes, altered DNA methylation profiles, and disruption of trophoblastic differentiation pathways in complete moles (Xue et al., 2004; Hui et al., 2017). These findings support the concept that molar pregnancies represent disorders of epigenetic programming as much as chromosomal imbalance.

Molecular studies have demonstrated that these epigenetic alterations influence pathways involved in trophoblast proliferation, differentiation, and malignant transformation, providing insight into mechanisms through which some molar pregnancies progress to GTN. Comparative molecular analyses between hydatidiform moles, choriocarcinoma, and normal placental tissue have identified abnormalities in pathways involved in cell-cycle regulation, invasion, and tumour-associated signalling pathways (Xue et al., 2004; Hui et al., 2017).

Another area of progress is understanding genetic predisposition to GTD. Mutations involving genes such as NLRP7 and KHDC3L have been shown to disrupt establishment of normal maternal imprinting during oogenesis, resulting in recurrent molar pregnancies despite biparental inheritance (Murdoch et al., 2006; Nguyen and Slim, 2014). These discoveries have significantly advanced understanding of the role of epigenetic regulation in trophoblastic development and have enabled improved genetic counselling and diagnostic assessment in affected families.

Research has also examined the signalling pathways regulating trophoblast proliferation and differentiation. Experimental studies suggest that abnormal activation of Wnt/β-catenin signalling contributes to sustained cytotrophoblast proliferation and impaired trophoblastic differentiation in molar tissue (Knöfler and Pollheimer, 2013). Similarly, overexpression of epidermal growth factor receptor has been demonstrated in gestational trophoblastic disease and may contribute to trophoblastic hyperplasia through enhanced proliferative signalling (Balaram et al., 1997). These molecular abnormalities may represent potential therapeutic targets in rare cases of chemotherapy-resistant GTN.

The role of immune regulation in GTD has also become an area of increasing interest. Trophoblastic tissue normally exhibits immune-modulatory properties that facilitate maternal tolerance during pregnancy. Alterations in these pathways within molar tissue and GTN are now being investigated as potential contributors to tumour progression and immune escape (Veras et al., 2017). Recognition of immune checkpoint pathway involvement has also contributed to the development of novel treatment approaches for resistant GTN, including immune checkpoint inhibition (Ghorani et al., 2017). Collectively, these advances continue to refine understanding of GTD biology and may ultimately contribute to improved molecular classification, risk stratification, and targeted therapeutic approaches.

1.5.7.2 Novel Biomarkers for Risk Stratification

A current research goal is to identify biomarkers that can predict which molar pregnancies will progress to persistent disease, allowing more personalised management. One promising marker is hyperglycosylated hCG (HhCG). As discussed, HhCG is disproportionately elevated in more aggressive trophoblastic disease reflecting its association with invasive trophoblast behaviour (Cole, 2009). Although HhCG has been investigated as a potential marker of persistent GTD and malignant transformation, its clinical role remains supplementary rather than replacing conventional serial hCG surveillance (Cole, 2009; Iles et al., 2010).

Emerging molecular approaches, including analysis of circulating tumour-derived material and trophoblast-specific molecular markers, have also been investigated as potential methods of improving early identification of patients at risk of persistent disease. However, these approaches remain largely experimental and have not yet replaced hCG-based monitoring in routine clinical practice.

One particularly useful diagnostic advance that has already impacted practice is the combined use of p57 immunostaining and molecular genotyping, as detailed earlier. While these are diagnostic rather than predictive biomarkers, they improve classification and thus indirectly aid risk stratification (Hui et al., 2017). Despite advances in molecular biomarker discovery, serial quantitative hCG measurement remains the central biomarker in GTD management. Recent research has therefore focused not only on identifying new biomarkers, but also on improving interpretation of existing hCG trends through optimised surveillance strategies and predictive modelling approaches (Coyle et al., 2018).

1.5.7.3 Refinements in hCG Surveillance Protocols

Recent research has increasingly focused on refining hCG surveillance protocols to improve earlier identification of patients at risk of persistent GTD while reducing unnecessary prolonged monitoring in patients with spontaneously resolving disease. Traditional surveillance strategies rely predominantly on fixed FIGO diagnostic criteria and serial hCG interpretation using predefined thresholds (Ngan et al., 2018; Tidy et al., 2021). However, emerging evidence suggests that early hCG kinetics may provide additional prognostic information before formal diagnostic criteria for GTN are reached.

Several studies have explored the use of hCG decline curves, nomograms, and predictive modelling approaches to identify abnormal disease trajectories earlier in follow-up (Eysbouts et al., 2017; Coyle et al., 2018; Hoeijmakers et al., 2021). These approaches aim to move beyond binary interpretation of hCG results by incorporating dynamic changes in hCG decline over time. Such work has provided the conceptual basis for the predictive modelling approaches explored within this thesis.

Despite these advances, most proposed models remain limited by relatively small cohort sizes, retrospective design, and lack of external validation. Furthermore, variability in surveillance schedules and assay methodology between centres continues to present challenges for broader implementation. Consequently, standardised hCG surveillance remains the cornerstone of post-molar follow-up, with predictive modelling currently representing a complementary rather than replacement approach.

1.5.7.4 Minimising Chemotherapy Exposure

Given that nearly 80–90% of molar patients do not need chemotherapy, a key goal is to treat fewer patients with unnecessary chemotherapy without missing anyone who truly needs it. Current research therefore focuses on improving risk stratification to identify patients who require intervention while avoiding overtreatment in those likely to undergo spontaneous resolution.

In selected high-risk cases, prophylactic chemotherapy using single-agent methotrexate or actinomycin-D has been investigated as a strategy to reduce progression to post-molar GTN. Evidence from systematic reviews suggests that prophylactic chemotherapy can reduce the incidence of GTN development in high-risk complete moles; however, concerns regarding unnecessary chemotherapy exposure, treatment toxicity, and the effectiveness of modern hCG

surveillance mean that it is not routinely adopted where reliable follow-up is available (Wang et al., 2017; Tidy et al., 2021).

Another approach aimed at reducing chemotherapy exposure is second uterine evacuation in carefully selected patients with low-risk, non-metastatic GTN. The GOG-242 study demonstrated that repeat curettage resulted in hCG remission in a proportion of patients, allowing some individuals to avoid chemotherapy entirely (Osborne et al., 2016). However, patient selection remains critical, and this approach is generally considered only in appropriately selected low-risk cases managed within specialist centres (Osborne et al., 2016; Tidy et al., 2021).

Targeted therapies have also emerged for patients with resistant GTN. Although rarely required in low-risk disease due to the effectiveness of conventional chemotherapy, advances in understanding tumour biology have led to investigation of immune checkpoint inhibitors, with pembrolizumab demonstrating activity in drug-resistant GTN (Ghorani et al., 2017).

Overall, recent advances in GTD management increasingly aim to maintain excellent cure rates while reducing unnecessary treatment burden. Improved interpretation of hCG kinetics, biomarker development, and personalised risk assessment provide opportunities to refine patient management and represent the foundation for predictive approaches explored within this thesis.

1.6 Organisation of GTD Services in the UK and Dataset Context

The organisation of GTD services in the United Kingdom is distinct from that of many other healthcare systems and has important implications for both clinical management and research datasets. Unlike more decentralised models seen internationally, the UK operates a nationally coordinated system for GTD diagnosis, treatment, and follow-up, delivered through specialist trophoblastic disease centres (Tidy et al., 2021). The development of this model originated from the establishment of the UK hydatidiform mole registration programme, which demonstrated the clinical value of centralised registration, structured follow-up, and specialist management of trophoblastic disease (Bagshawe et al., 1986). This approach has subsequently been formalised through national guidance and ensures consistent application of diagnostic criteria, treatment protocols, and longitudinal surveillance strategies (Seckl et al., 2010; Tidy et al., 2021).

Within this model, patient registration is a fundamental component of care. Patients diagnosed with molar pregnancy in the UK are referred for registration with a specialist trophoblastic centre, enabling structured hCG surveillance and longitudinal follow-up. Early registration following uterine evacuation ensures prompt initiation of monitoring and enables detailed assessment of longitudinal hCG changes (Tidy et al., 2021). These early hCG trajectories provide a valuable research opportunity for investigating whether patterns of hCG decline can predict subsequent disease behaviour, including spontaneous resolution, persistence requiring treatment, and response to chemotherapy.

This nationally coordinated approach differs from healthcare systems in countries such as France, the Netherlands, and the United States, where GTD management is often decentralised and may lack mandatory national registration, resulting in variability in follow-up practices and potential loss to longitudinal data capture (Seckl et al., 2010; Ngan et al., 2018).

A defining feature of the UK system is the use of serial hCG measurement as the cornerstone of disease monitoring, with standardised protocols governing testing frequency and duration. Centralised follow-up has been associated with high cure rates, typically exceeding 98%, and enables early detection of disease persistence or relapse (Seckl et al., 2010; Lurain, 2011). The consistency of surveillance protocols and near-complete patient capture contribute to the generation of high-quality longitudinal datasets, which are particularly valuable for investigating disease kinetics, treatment response, and predictive modelling approaches.

The Sheffield Trophoblastic Disease Centre operates within this national framework and contributes to one of the largest longitudinal GTD datasets globally. A distinguishing characteristic of the Sheffield programme is the routine use of urinary hCG monitoring, specifically the hCG/creatinine ratio, as a major component of post-molar surveillance. This differs from Charing Cross, where surveillance predominantly relies on serum hCG measurement, while Dundee incorporates urine-based monitoring approaches within its follow-up strategy. The availability of extensive, standardised urine-based data within the Sheffield cohort provides a unique opportunity to evaluate hCG kinetics using a non-invasive and patient-convenient monitoring method.

The centralised nature of the UK system also has important implications for the timing and completeness of data collection. Early registration and structured follow-up ensure that initial hCG trajectories are consistently captured, which is essential when investigating early markers of disease behaviour. In decentralised systems, delays in referral or heterogeneity in monitoring

practices may obscure early kinetic patterns, thereby limiting the development and validation of predictive models (Savage et al., 2015; Ngan et al., 2018).

While the strengths of the UK model lie in its standardisation and completeness of follow-up, these characteristics also have implications for generalisability. The findings generated within this thesis are derived from a healthcare system with coordinated specialist oversight, standardised monitoring protocols, and high levels of patient follow-up compliance. Therefore, caution is required when extrapolating predictive models to healthcare settings where differences exist in registration pathways, monitoring frequency, assay methodology, or thresholds for initiating treatment.

In summary, the UK's centralised GTD service model provides a uniquely robust platform for both clinical care and research. The integration of specialist registration, standardised hCG surveillance, and comprehensive longitudinal follow-up distinguishes UK datasets from those generated within more fragmented healthcare systems. These factors are fundamental to understanding the structure and quality of the Sheffield dataset and explain why this cohort provides a rare opportunity to investigate early hCG kinetics as predictors of spontaneous disease resolution, treatment requirement, and chemotherapy response.

1.7 Malignant Gestational Trophoblastic Neoplasia (GTN)

Following the diagnosis and management of hydatidiform mole, the principal clinical challenge is identifying the minority of patients who develop persistent or malignant disease requiring systemic treatment. Although modern surveillance and chemotherapy protocols have resulted in excellent outcomes for patients with GTN, considerable variation exists in tumour behaviour, treatment response, and the requirement for therapy escalation (Ngan et al., 2018; Tidy et al., 2021).

Understanding the mechanisms underlying malignant transformation, clinical presentation, diagnostic assessment, and prognostic classification of GTN is therefore essential for optimising patient management. This section explores the biological and clinical features of GTN, providing the foundation for subsequent discussion of treatment strategies and the role of hCG-based predictive modelling.

1.7.1 Pathophysiology and Molecular Mechanisms

The molecular and epigenetic abnormalities underlying molar trophoblastic proliferation have been discussed in Sections 1.3 and 1.5. This section focuses specifically on the mechanisms associated with malignant transformation, invasion, metastasis, and immune escape in GTN.

Malignant GTNs arise from placental trophoblastic tissue and include persistent GTN, choriocarcinoma, PSTT, and ETT (Seckl et al., 2010; Hancock and Tidy, 2021). Persistent disease and choriocarcinoma frequently develop following complete hydatidiform moles and retain many of the underlying epigenetic abnormalities associated with androgenetic trophoblastic proliferation (Xue et al., 2004; Seckl et al., 2010). However, malignant GTN is characterised by additional acquisition of invasive capacity, vascular dissemination, resistance to apoptotic signalling, and immune evasion.

A key feature of malignant GTN is dysregulation of signalling pathways involved in trophoblast invasion and proliferation. Molecular studies have identified alterations in pathways regulating cellular proliferation, invasion, angiogenesis, and trophoblastic differentiation, which may contribute to malignant progression in GTN (Xue et al., 2004).

GTNs also demonstrate subtype-specific molecular characteristics reflecting their trophoblastic cell of origin. PSTT and ETT arise from intermediate trophoblast and exhibit distinct immunophenotypic profiles compared with choriocarcinoma. Differences in expression of markers including p63, human placental lactogen (hPL), and other trophoblastic differentiation markers assist pathological classification and reflect their underlying trophoblastic lineage (Froeling et al., 2019; Hancock and Tidy, 2021).

Although GTNs generally demonstrate relatively low tumour mutational burden compared with many other solid tumours, molecular studies have identified abnormalities associated with proliferation, invasion, and immune regulation.

Another hallmark of GTN is immune evasion. Trophoblastic tumours frequently express high levels of programmed death ligand-1, which likely contributes to suppression of anti-tumour immune responses and persistence of malignant trophoblastic cells (Veras et al., 2017). Recognition of this mechanism has led to increasing interest in immune checkpoint inhibitors for

chemotherapy-resistant GTN, with early studies demonstrating promising responses in selected refractory cases (Ghorani et al., 2017).

Overall, malignant GTN arises through the interaction of abnormal trophoblastic proliferation, dysregulated invasive signalling pathways, resistance to apoptosis, and immune escape mechanisms. These processes contribute to the unique clinical behaviour of GTN, characterised by rapid vascular dissemination yet marked sensitivity to chemotherapy.

1.7.2 Epidemiology and Risk Factors

Gestational trophoblastic neoplasia most commonly develops following hydatidiform mole, although it may also arise after miscarriage, ectopic pregnancy, or term pregnancy (Seckl et al., 2010; Lurain, 2011). The likelihood of progression varies according to the antecedent pregnancy type, with complete hydatidiform moles carrying a substantially greater risk of malignant transformation than partial moles due to their androgenetic origin and increased trophoblastic proliferation (Ngan et al., 2018; Tidy et al., 2021).

Persistent GTN and choriocarcinoma represent the most common malignant forms of GTN. Choriocarcinoma may occur following any gestational event, although antecedent complete mole remains the most frequent precursor lesion (Seckl et al., 2010). In contrast, PSTT and ETT are rare tumours arising from intermediate trophoblast and account for a small minority of GTN cases (Froeling et al., 2019). These subtypes typically present later after the antecedent pregnancy and exhibit lower hCG production relative to tumour burden.

Following diagnosis of GTN, prognosis and treatment selection are determined using the FIGO/WHO prognostic scoring system rather than individual post-molar risk factors. This system incorporates clinical variables including antecedent pregnancy, interval from pregnancy, pretreatment hCG concentration, tumour size, metastatic burden, metastatic site, and previous chemotherapy exposure (FIGO, 2002; Ngan et al., 2018). Patients with low-risk disease generally respond well to single-agent chemotherapy, whereas high-risk GTN requires multi-agent treatment approaches.

PSTT and ETT differ clinically from choriocarcinoma and persistent GTN in several important respects. These tumours are less chemosensitive, tend to metastasise later, and are more strongly associated with prolonged interval from antecedent pregnancy (Froeling et al., 2019;

Hancock and Tidy, 2021). Consequently, hysterectomy plays a more prominent role in management of localised PSTT and ETT compared with other forms of GTN.

Despite the aggressive potential of malignant GTN, outcomes remain highly favourable when disease is identified and treated appropriately within specialist centres. Centralised management and structured hCG surveillance have contributed significantly to the excellent cure rates achieved in modern GTD practice (Seckl et al., 2010; Tidy et al., 2021).

1.7.3 Clinical Presentation and Symptoms

GTN demonstrates a wide spectrum of clinical presentations, ranging from asymptomatic biochemical persistence detected through hCG surveillance to symptomatic metastatic disease. Within organised follow-up programmes, most post-molar GTN is identified through abnormal hCG regression before significant clinical symptoms develop (Seckl et al., 2010; Tidy et al., 2021). However, GTN should be considered in patients presenting with persistent or unexplained vaginal bleeding following any pregnancy event, including molar pregnancy, miscarriage, or term delivery (Lurain, 2011; Ngan et al., 2018).

1.7.3.1 Persistent GTN

Persistent GTN following molar pregnancy is most commonly identified by plateauing or rising hCG concentrations during surveillance. This biochemical pattern reflects ongoing active trophoblastic proliferation and forms the basis of internationally recognised diagnostic criteria for post-molar GTN (Ngan et al., 2018; Tidy et al., 2021).

Clinically, patients may present with persistent vaginal bleeding, abnormal discharge, or symptoms associated with local uterine invasion. In cases with metastatic disease, symptoms reflect the site of involvement. The lungs represent the most frequent metastatic site and may present with cough, dyspnoea, chest pain, or haemoptysis, while less common cerebral or hepatic metastases may result in neurological symptoms or abdominal complications (Seckl et al., 2010; Lurain, 2011).

Despite the potential for aggressive clinical presentation, centralised surveillance programmes mean that most post-molar GTN cases are identified early, allowing treatment initiation before the development of extensive metastatic disease (Tidy et al., 2021).

1.7.3.2 PSTT and ETT

PSTT and ETT demonstrate distinct clinical characteristics compared with persistent GTN and choriocarcinoma. These tumours arise from intermediate trophoblast and often present months or years after the antecedent pregnancy, commonly with irregular vaginal bleeding or abnormal uterine findings (Feng et al., 2019; Froeling et al., 2019; Hancock and Tidy, 2021).

Unlike choriocarcinoma, PSTT and ETT typically produce lower concentrations of hCG relative to tumour burden due to their intermediate trophoblastic origin, with PSTT demonstrating greater expression of markers associated with implantation-site trophoblast differentiation, including hPL (Feng et al., 2019; Hancock and Tidy, 2021).

These tumours are generally less chemosensitive than other forms of GTN and prognosis is strongly influenced by disease stage and interval from antecedent pregnancy. Early recognition remains important, as localised PSTT and ETT can often be effectively managed surgically, whereas advanced disease requires more intensive treatment approaches (Froeling et al., 2019; Hancock and Tidy, 2021).

1.7.4 Diagnostic Approaches

Diagnosis of GTN requires integration of clinical history, biochemical assessment, imaging findings, and, where available, histopathological evaluation. Unlike hydatidiform mole, where diagnosis is primarily established through examination of evacuated pregnancy tissue, many cases of post-molar GTN are diagnosed clinically based on abnormal hCG regression during surveillance rather than through tissue confirmation (Seckl et al., 2013; Ngan et al., 2018).

Accurate diagnosis involves not only confirming persistent trophoblastic disease but also determining disease extent, GTN subtype, and prognostic risk category. Quantitative hCG measurement remains central to diagnosis and monitoring; however, interpretation must consider tumour biology, particularly in PSTT and ETT, where hCG production may be disproportionately low relative to tumour burden (Froeling et al., 2019; Hancock and Tidy, 2021).

Histopathological examination, immunohistochemical assessment, and molecular techniques provide additional information in diagnostically challenging cases, particularly for differentiating rare GTN subtypes and confirming gestational origin (Hui et al., 2017; Shih, 2007). Imaging is subsequently used to evaluate local invasion, identify metastatic disease, and guide FIGO staging and risk classification, which directly determines subsequent treatment strategy (Seckl et al., 2013; Ngan et al., 2018).

1.7.4.1 Laboratory and Biomarkers

The cornerstone of GTN diagnosis is the hCG test (Seckl et al., 2013; Tidy et al., 2021). Quantitative hCG measurement is an extremely sensitive marker for trophoblastic tissue and is used to both detect and monitor GTNs. Following evacuation of a molar pregnancy, persistent or rising hCG concentrations indicate ongoing trophoblastic proliferation and form the basis of internationally recognised diagnostic criteria for post-molar GTN (Ngan et al., 2018; Tidy et al., 2021).

Beyond its diagnostic role, serial hCG monitoring provides important information regarding disease behaviour, as the pattern and rate of hCG decline following evacuation can identify patients at increased risk of requiring subsequent treatment. Large cohort analyses have demonstrated that hCG kinetics during surveillance can provide clinically useful prognostic information and support more individualised approaches to follow-up (Coyle et al., 2018; Tidy et al., 2021).

Serum hCG concentration also contributes to GTN risk stratification, with pretreatment hCG incorporated into the FIGO/WHO prognostic scoring system alongside other clinical factors including antecedent pregnancy, disease interval, metastatic burden, and previous treatment exposure (Seckl et al., 2013; Ngan et al., 2018).

However, interpretation of hCG concentration must consider GTN subtype. Choriocarcinoma and persistent GTN are typically associated with marked hCG elevation, whereas PSTT and ETT may produce relatively low concentrations of hCG despite clinically significant tumour burden (Froeling et al., 2019; Hancock and Tidy, 2021).

This difference reflects their origin from intermediate trophoblast rather than villous trophoblast. Human placental lactogen, although not routinely used as a circulating diagnostic biomarker,

may assist histopathological assessment because expression patterns reflect trophoblastic differentiation and can support the diagnosis of PSTT (Shih, 2007; Feng et al., 2019; Hancock and Tidy, 2021).

1.7.4.2 Histopathology

Definitive histological classification of GTN depends on examination of tumour tissue where available, although many cases of post-molar GTN are diagnosed clinically based on hCG surveillance findings rather than surgical specimens. The histological features of CHM and PHM have been described previously (Section 1.5.3.1); however, malignant transformation is associated with invasive growth patterns, loss of normal placental architecture, and subtype-specific pathological characteristics.

Histopathological assessment remains important for differentiating GTN subtypes, particularly where clinical presentation or hCG patterns are atypical. Ancillary techniques including immunohistochemistry and molecular approaches have improved diagnostic accuracy, particularly in cases where morphology alone is insufficient (Lai et al., 2004; Hui et al., 2017; Ronnett, 2018).

Persistent GTN is characterised by persistence of abnormal chorionic villi with trophoblastic proliferation and invasion into the myometrium. The presence of chorionic villi remains the key pathological feature distinguishing persistent GTN from choriocarcinoma, which consists of malignant trophoblastic proliferation without villous structures (Shih, 2007; Soper, 2021).

Choriocarcinoma demonstrates highly atypical cytotrophoblastic and syncytiotrophoblastic proliferation, typically accompanied by extensive haemorrhage, necrosis, and vascular invasion. Due to the vascular nature of these tumours, biopsy is often avoided when the diagnosis can be established clinically through characteristic presentation, markedly elevated hCG concentrations, and imaging findings (Seckl et al., 2013; Soper, 2021).

PSTT and ETT demonstrate distinct histopathological features reflecting their origin from intermediate trophoblast. Immunohistochemical assessment assists differentiation between these entities, with markers including p63, p40, hPL, and Ki-67 contributing to diagnostic classification (Shih and Kurman, 2001; Allison et al., 2006; McCarthy et al., 2018). Accurate classification is clinically important because PSTT and ETT differ from other GTN subtypes

through lower hCG production, reduced chemosensitivity, and a greater role for surgical management (Froeling et al., 2019; Hancock and Tidy, 2021).

1.7.4.3 Imaging and Disease Staging

Imaging plays an important role in the assessment of GTN, although its purpose differs from the initial diagnosis of molar pregnancy described previously (Section 1.5.3.3). In established GTN, imaging is primarily used to determine disease extent, identify metastatic involvement, and support accurate FIGO staging and risk stratification (Seckl et al., 2013; Ngan et al., 2018).

Pelvic ultrasound is commonly used as the initial imaging modality to assess uterine disease, particularly in patients presenting with abnormal bleeding or abnormal hCG trends following pregnancy. Persistent GTN may appear as a focal intrauterine lesion or demonstrate increased vascularity on Doppler assessment, although findings are not sufficiently specific to establish diagnosis without correlation with hCG trends and clinical features (Allen et al., 2006; Dhanda et al., 2014). MRI provides superior assessment of local tumour extent, including depth of myometrial invasion and involvement of adjacent pelvic structures, and may be particularly useful in complex cases or when surgical management is being considered (Dhanda et al., 2014; Shaaban et al., 2017).

Assessment for metastatic disease forms an essential component of GTN staging. The lungs represent the most common site of metastatic spread; therefore, chest imaging is routinely performed during evaluation. Chest radiography remains incorporated within FIGO staging systems, although CT provides increased sensitivity for detecting smaller pulmonary lesions. However, interpretation of small CT-detected abnormalities requires clinical judgement, as detection of very small pulmonary nodules may influence staging without necessarily altering clinical outcome (Seckl et al., 2013; Ngan et al., 2018).

Further cross-sectional imaging with CT or MRI is guided by disease risk, clinical presentation, and evidence of metastatic involvement. High-risk GTN may require assessment for liver, brain, or other distant metastases because these findings influence prognosis and treatment intensity (Seckl et al., 2013; Shaaban et al., 2017). PET-CT is not routinely required for initial diagnosis but may have a role in selected patients with recurrent, resistant, or unclear disease sites where conventional imaging is inconclusive (Dhanda et al., 2014; Joyce et al., 2022).

Overall, imaging complements biochemical monitoring rather than replacing it. The integration of serial hCG measurement, imaging findings, and FIGO risk assessment enables accurate classification of GTN and guides selection of appropriate treatment strategies (Ngan et al., 2018; Tidy et al., 2021).

1.7.5 Treatment Strategies

Malignant GTN is unique among solid malignancies because of its exceptional sensitivity to systemic therapy, with cure achievable even in many patients presenting with metastatic disease. Modern outcomes have been significantly improved through the introduction of specialist treatment centres, standardised risk assessment, and risk-adapted chemotherapy approaches (Seckl et al., 2013; Brown et al., 2017). Treatment selection is primarily guided by the FIGO/WHO prognostic scoring system, which classifies patients into low-risk and high-risk groups based on clinical and biochemical factors including antecedent pregnancy, interval from pregnancy, pretreatment hCG concentration, metastatic burden, and previous chemotherapy exposure (Ngan et al., 2018). This approach allows treatment intensity to be tailored according to predicted disease behaviour, maximising cure rates while minimising unnecessary treatment exposure.

1.7.5.1 Risk-Adapted Chemotherapy

Chemotherapy remains the cornerstone of treatment for persistent GTN and choriocarcinoma. Patients classified as low-risk GTN (FIGO score 0–6) are typically treated with single-agent chemotherapy, most commonly MTX or ActD. Both agents demonstrate high efficacy, although treatment response varies between individuals and some patients require transition to alternative therapy because of drug resistance or inadequate hCG decline (Lawrie et al., 2016). Despite this, sequential use of available therapies results in cure rates approaching 100% for low-risk disease (Seckl et al., 2013; Tidy et al., 2021).

Assessment of treatment response relies heavily on serial hCG measurement, with plateauing or rising concentrations indicating inadequate response and the need for treatment modification. While current management strategies are highly successful, the requirement to identify treatment failure through ongoing hCG monitoring means some patients may receive ineffective chemotherapy cycles before escalation. This limitation has driven interest in earlier prediction of treatment response using hCG kinetics and other prognostic approaches.

High-risk GTN (FIGO score ≥7) requires combination chemotherapy due to increased risk of widespread disease and treatment resistance. EMA/CO remains the most widely established first-line regimen and has resulted in substantial improvements in survival for patients with metastatic GTN (Alifrangis et al., 2013). Patients with ultra-high-risk features, including very high FIGO scores, extensive metastatic disease, or brain and liver involvement, have poorer outcomes and may require modified treatment approaches, including induction chemotherapy to reduce early treatment-related complications (Bolze et al., 2016).

Overall survival has improved considerably over recent decades as a result of earlier diagnosis, specialist management, improved risk stratification, and development of effective salvage therapies for resistant disease (Brown et al., 2017; Joyce et al., 2022).

1.7.5.2 Surgical and Adjunct Treatment Approaches

Although chemotherapy represents the primary treatment modality for most forms of GTN, surgery remains important in selected clinical scenarios. Unlike persistent GTN and choriocarcinoma, PSTT and ETT demonstrate reduced chemosensitivity and are therefore managed differently. Surgical resection, most commonly hysterectomy for localised disease, plays a central role in treatment and may be curative when disease is confined to the uterus (Froeling et al., 2019; Hancock and Tidy, 2021).

Surgery may also be considered in selected cases of chemotherapy-resistant GTN, persistent localised disease, or complications such as uncontrolled haemorrhage. However, the requirement for surgical intervention has reduced substantially with improvements in chemotherapy effectiveness and specialist multidisciplinary management (Seckl et al., 2013; Joyce et al., 2022).

1.7.5.3 Emerging Therapeutic Approaches

Although conventional chemotherapy successfully cures the majority of patients with GTN, a small subset develop multi-drug resistant disease requiring alternative therapeutic approaches. Advances in understanding tumour immune regulation have identified immune checkpoint inhibition as a promising strategy, particularly because trophoblastic tumours frequently demonstrate high expression of PD-L1, allowing evasion of host immune responses (Veras et al., 2017).

Clinical experience with immune checkpoint inhibitors has demonstrated encouraging results in resistant GTN. Pembrolizumab has produced durable responses in patients with disease resistant to multiple chemotherapy regimens, including cases previously considered difficult to treat (Ghorani et al., 2017). Similarly, the TROPHIMMUN phase II trial demonstrated that avelumab could achieve sustained remission in a proportion of patients with GTN resistant to single-agent chemotherapy (You et al., 2020).

While immunotherapy currently remains reserved for selected patients with resistant or relapsed disease, these developments demonstrate how improved understanding of GTN biology may continue to refine treatment approaches. Nevertheless, conventional chemotherapy remains the foundation of GTN management, with emerging therapies providing additional options for patients who do not respond to established treatment pathways.

1.7.6 Prognosis and Outcomes

Following successful treatment of GTN, continued surveillance is required to confirm sustained remission and identify the small proportion of patients who develop recurrent disease. Serial hCG measurement remains the cornerstone of post-treatment monitoring because rising concentrations provide an early and sensitive indicator of persistent or recurrent trophoblastic activity (Seckl et al., 2013; Tidy et al., 2021).

The frequency and duration of follow-up depend on the GTN subtype, treatment received, and individual patient risk factors. Most relapses following treatment for persistent GTN and choriocarcinoma occur within the first year after completion of therapy, supporting the use of structured hCG monitoring during this period (Seckl et al., 2013). Early identification of abnormal hCG trends allows prompt investigation and initiation of further treatment when required.

Follow-up considerations differ for PSTT and ETT because of their distinct biological behaviour. Compared with other forms of GTN, these tumours may demonstrate slower progression, lower hCG production, and the potential for later recurrence. Therefore, prolonged clinical surveillance may be required, particularly in patients with higher-risk features or disease extending beyond the uterus (Froeling et al., 2019; Hancock and Tidy, 2021).

Long-term outcomes following successful GTN treatment are generally favourable, with most patients returning to normal health after completion of therapy. However, survivorship

considerations remain important, particularly following intensive multi-agent chemotherapy. Long-term follow-up studies have demonstrated that single-agent chemotherapy is associated with minimal long-term morbidity, while combination chemotherapy may be associated with increased risks of earlier menopause and secondary malignancy in a small proportion of patients (Savage et al., 2015).

Overall, surveillance following GTN treatment relies on accurate longitudinal assessment of hCG dynamics. The ability of serial hCG measurement to detect persistent or recurrent disease highlights its central role not only in clinical follow-up, but also as a valuable tool for investigating disease behaviour and developing predictive approaches to optimise patient management.

1.8 Chemotherapy in GTD

Chemotherapy has transformed the management of malignant GTD, converting a group of historically fatal conditions into some of the most curable solid tumours in modern oncology. The introduction of effective cytotoxic agents, particularly MTX and ActD, demonstrated that even widely metastatic trophoblastic malignancies could achieve durable remission with systemic treatment (Chabner and Roberts, 2005; Seckl et al., 2013).

MTX and ActD remain the principal first-line agents used in low-risk GTN because of their high efficacy and favourable long-term outcomes. Although overall cure rates are excellent, variation in individual treatment response remains clinically important, as patients who develop drug resistance may receive multiple ineffective treatment cycles before therapy escalation. Understanding the molecular mechanisms responsible for chemotherapy response and resistance is therefore essential for improving treatment stratification and developing predictive approaches capable of identifying ineffective therapy earlier.

1.8.1 Molecular Mechanisms of Action

The effectiveness of chemotherapy in GTD is closely linked to the biological characteristics of trophoblastic cells, particularly their rapid proliferation, high metabolic activity, and dependence on tightly regulated DNA synthesis pathways. Although GTN can demonstrate aggressive clinical behaviour with early vascular invasion and metastatic potential, these same proliferative characteristics contribute to the marked sensitivity of many trophoblastic tumours to cytotoxic chemotherapy.

The two principal agents used in first-line treatment of low-risk GTN, MTX and ActD, achieve their therapeutic effects through distinct mechanisms. MTX primarily disrupts folate-dependent nucleotide synthesis, thereby impairing DNA replication, whereas ActD directly interferes with DNA transcription and RNA synthesis. These differing mechanisms help explain why resistance to one agent does not necessarily predict resistance to another and provide the biological rationale for sequential treatment approaches in GTN.

1.8.1.1 MTX mechanisms of action

MTX is an antifolate chemotherapy agent that has played a central role in cancer therapy since the development of antifolate-based treatment approaches, exerting its cytotoxic effect through disruption of folate-dependent nucleotide synthesis (Chabner and Roberts, 2005; Visentin et al., 2012). Cellular uptake occurs primarily through folate transport systems, particularly the reduced folate carrier pathway, after which MTX undergoes intracellular polyglutamation, increasing cellular retention and enhancing inhibition of folate-dependent enzymatic pathways (Matherly and Goldman, 2003; Visentin et al., 2012).

The primary target of MTX is dihydrofolate reductase (DHFR), an enzyme responsible for reducing dihydrofolate to tetrahydrofolate (Gonen and Assaraf, 2012; Visentin et al., 2012). Inhibition of DHFR reduces the availability of tetrahydrofolate cofactors required for thymidylate and purine synthesis, resulting in impaired DNA replication and repair (Gonen and Assaraf, 2012; Visentin et al., 2012). Rapidly proliferating trophoblastic cells are particularly susceptible to this disruption because of their high requirement for nucleotide synthesis during cellular division (Visentin et al., 2012). Folinic acid rescue is administered following MTX treatment to reduce toxicity in normal tissues by bypassing DHFR inhibition, while maintaining selective cytotoxic effects against rapidly proliferating trophoblastic cells (Visentin et al., 2012).

The effectiveness of MTX in GTN reflects the highly proliferative nature of trophoblastic tissue and its dependence on tightly regulated DNA synthesis pathways. However, treatment response is influenced by multiple factors including intracellular drug accumulation, folate pathway regulation, and tumour cell adaptation. Alterations affecting MTX transport, retention, or target enzymes can reduce antifolate sensitivity and contribute to the development of treatment resistance (Zhao and Goldman, 2003; Assaraf, 2007).

Although these molecular mechanisms provide insight into MTX sensitivity and resistance, they are not routinely measurable in clinical practice. Consequently, treatment response continues to be assessed primarily through serial hCG monitoring, highlighting the importance of identifying early biochemical patterns capable of predicting ineffective treatment before prolonged exposure to unsuccessful chemotherapy.

1.8.1.2 ActD Mechanism of Action

ActD is a cytotoxic antibiotic that exerts its anti-tumour activity through direct interaction with DNA, leading primarily to inhibition of transcription and disruption of nucleic acid synthesis (Koba and Konopa, 2005). Unlike MTX, which indirectly affects DNA synthesis through inhibition of folate metabolism, ActD binds directly to double-stranded DNA and prevents the expression of genes required for continued cellular proliferation (Sobell, 1985; Koba and Konopa, 2005).

The principal mechanism of ActD involves intercalation between adjacent guanine-cytosine base pairs within DNA. This interaction stabilises the DNA–drug complex and physically prevents movement of RNA polymerase along the DNA template, resulting in suppression of messenger RNA synthesis and subsequent impairment of protein production (Goldberg et al., 1962; Sobell, 1985). Cells with high rates of transcription and proliferation, such as malignant trophoblastic cells, are therefore particularly susceptible to ActD-mediated cytotoxicity.

In addition to inhibition of RNA synthesis, ActD has been shown to induce broader cellular stress responses, including interference with DNA-associated enzymatic processes and activation of pathways leading to cell-cycle arrest and apoptosis (Koba and Konopa, 2005; Nitiss, 2009). These effects contribute to tumour cell death and explain the activity of ActD across a range of rapidly proliferating malignancies.

The distinct mechanisms of MTX and ActD provide the biological rationale for sequential treatment approaches in low-risk GTN. Because ActD acts directly on DNA transcription rather than folate metabolism, resistance mechanisms affecting MTX sensitivity do not necessarily confer resistance to ActD. This lack of complete cross-resistance explains why patients demonstrating inadequate biochemical response to MTX may still achieve remission following transition to ActD therapy.

1.8.2 Clinical Applications in GTD

The introduction of effective chemotherapy transformed GTD management, with modern treatment strategies achieving exceptionally high cure rates while preserving fertility in most patients. Current therapeutic approaches are based on risk-adapted treatment, whereby chemotherapy intensity is determined using the FIGO staging system and modified WHO prognostic scoring system (Seckl et al., 2013; Ngan et al., 2018).

Although survival outcomes are excellent, optimisation of treatment remains clinically important. A key challenge is identifying patients who are unlikely to respond to initial therapy, as ineffective chemotherapy may result in additional treatment cycles, increased toxicity, and delayed transition to alternative regimens (Braga et al., 2021a). Therefore, understanding chemotherapy selection, treatment response, and mechanisms of resistance is central to improving personalised management strategies in GTD.

1.8.2.1 Single-Agent Chemotherapy in Low-Risk GTN

Low-risk GTN is typically managed with single-agent chemotherapy using either MTX or ActD, with treatment selection guided by disease characteristics, patient factors, toxicity profiles, and institutional protocols (Seckl et al., 2013; Ngan et al., 2018). This approach maintains excellent cure rates while avoiding the increased toxicity associated with multi-agent chemotherapy.

MTX, commonly administered with folinic acid rescue, remains widely used as first-line treatment due to extensive clinical experience, established efficacy, and a favourable toxicity profile. Commonly used regimens include multi-day MTX protocols with folinic acid rescue and weekly intramuscular MTX schedules, although treatment protocols vary between specialist centres (Seckl et al., 2013; Ngan et al., 2018). ActD represents an effective alternative first-line agent and is frequently used following inadequate biochemical response or development of MTX resistance.

Comparative studies have demonstrated differences in primary remission rates between MTX and ActD. A randomised phase III trial demonstrated higher initial complete response rates with pulsed ActD compared with weekly MTX; however, overall cure rates remained excellent because patients with resistant disease could usually be successfully treated with alternative chemotherapy (Osborne et al., 2011). Similarly, systematic review evidence has shown that ActD is associated with higher primary remission rates compared with some MTX regimens, although

treatment decisions require consideration of toxicity, administration route, and patient-specific factors (Lawrie et al., 2016).

In clinical practice, MTX is often favoured initially because of its tolerability and lower incidence of adverse effects such as alopecia, whereas ActD may be associated with increased nausea, vomiting, and hair loss (Osborne et al., 2011). However, ActD remains highly effective as second-line therapy following MTX failure, with most patients achieving remission after appropriate treatment escalation.

Despite excellent outcomes, predicting which patients will develop resistance to first-line chemotherapy remains an important limitation of current management. Existing prognostic scoring systems predict overall outcome but have more limited ability to determine individual treatment response. Multicentre evidence has demonstrated that even patients within similar FIGO risk categories may have substantially different probabilities of developing resistance to single-agent chemotherapy, supporting the need for additional predictive approaches (Braga et al., 2021b).

Earlier recognition of ineffective treatment could reduce unnecessary chemotherapy exposure, minimise treatment burden, and allow more rapid transition to effective therapy. This provides the clinical rationale for investigating early biomarkers of response, including dynamic changes in hCG kinetics during treatment.

1.8.3 Resistance Mechanisms

Although chemotherapy achieves excellent cure rates in GTD, a proportion of patients demonstrate inadequate response to first-line treatment and require escalation to alternative therapeutic regimens. In low-risk GTN, resistance to single-agent chemotherapy remains clinically important because it can prolong exposure to ineffective treatment before biochemical criteria for escalation are met (Braga et al., 2021b).

Resistance may develop through multiple mechanisms, including reduced intracellular drug accumulation, altered drug retention, modification of molecular targets, and activation of broader cellular survival pathways. These processes are often multifactorial rather than attributable to a single molecular abnormality, which makes prospective prediction of treatment failure challenging (Assaraf et al., 2019). Understanding these mechanisms provides important

context for interpreting hCG response patterns and for developing approaches capable of identifying patients at risk of treatment failure earlier during their clinical course.

1.8.3.1 Methotrexate Resistance

Tumour resistance to methotrexate can arise through several well-characterised mechanisms, often reflecting adaptive changes in folate metabolism pathways (Matherly and Goldman, 2003; Zhao and Goldman, 2003; Assaraf, 2007; Gonen and Assaraf, 2012). These mechanisms broadly include reduced intracellular drug accumulation through impaired transport or increased efflux, decreased intracellular retention via altered polyglutamation, and target-based resistance involving changes in DHFR expression or function. Collectively, these adaptations reduce effective intracellular MTX activity and permit continued nucleotide synthesis despite antifolate therapy (Zhao and Goldman, 2003; Assaraf, 2007).

1.8.3.1.1 Reduced drug uptake

Reduced intracellular accumulation represents an important mechanism of MTX resistance. MTX uptake depends on functional folate transport pathways, particularly the reduced folate carrier system, and reduced transporter expression or activity may limit intracellular drug availability (Matherly and Goldman, 2003). In addition, increased expression of drug efflux transporters can promote removal of cytotoxic compounds from tumour cells, reducing intracellular exposure and contributing to multidrug resistance phenotypes (Assaraf et al., 2019).

1.8.3.1.2 Decreased MTX retention

Following cellular uptake, MTX undergoes intracellular polyglutamation by folylpolyglutamate synthase, which increases intracellular retention and enhances inhibition of folate-dependent enzymes. Polyglutamated MTX derivatives demonstrate prolonged intracellular persistence and greater inhibitory activity against targets including DHFR and thymidylate synthase (Allegra et al., 1985; Visentin et al., 2012).

Alterations affecting polyglutamation pathways can therefore reduce MTX sensitivity by decreasing intracellular drug retention and shortening the duration of antifolate activity. Disruption of this process enables tumour cells to maintain folate-dependent nucleotide synthesis despite MTX exposure, contributing to therapeutic resistance (Zhao and Goldman, 2003; Gonen and Assaraf, 2012).

1.8.3.1.3 DHFR overexpression

Increased expression of the target enzyme DHFR represents one of the earliest described mechanisms of antifolate resistance. Tumour cells may amplify the DHFR gene or increase DHFR expression, producing sufficient enzyme activity to overcome competitive inhibition by MTX and maintain tetrahydrofolate production (Schimke et al., 1978; Cowan et al., 1982).

This mechanism has been observed across multiple MTX-resistant tumour models and remains one of the most extensively characterised examples of drug resistance mediated through target amplification (Schimke et al., 1978; Gonen and Assaraf, 2012). However, resistance mechanisms in trophoblastic tumours appear multifactorial, and DHFR expression alone may not fully predict MTX sensitivity, suggesting that additional pathways including transport, retention, and cellular adaptation contribute to treatment failure (Han et al., 2010).

1.8.3.1.4 Altered DHFR affinity

Structural alteration of DHFR represents an additional mechanism through which tumour cells may evade antifolate activity. Mutations affecting the MTX-binding region of DHFR can reduce drug affinity while preserving enzymatic function, allowing continued folate metabolism despite exposure to therapeutic concentrations of MTX (Assaraf, 2007; Gonen and Assaraf, 2012).

Although individual resistance mechanisms are often described separately, MTX resistance usually reflects the combined influence of multiple cellular adaptations rather than a single isolated abnormality. Studies using choriocarcinoma models have demonstrated that MTX-resistant trophoblastic cells can develop alterations affecting antifolate sensitivity, supporting the relevance of these mechanisms within GTN biology (Sakai et al., 1989).

The clinical consequence of MTX resistance in GTN is inadequate biochemical response, typically identified through plateauing or rising hCG concentrations during treatment. Because molecular resistance mechanisms cannot currently be assessed routinely in clinical practice, serial hCG monitoring remains the principal method of identifying ineffective therapy. This highlights the importance of understanding early hCG kinetics and developing predictive approaches capable of identifying patients unlikely to respond before prolonged exposure to ineffective chemotherapy.

1.8.3.2 Actinomycin D Resistance

Resistance to ActD is less frequently encountered than MTX resistance in GTN; however, resistant disease may occur, particularly following multiple lines of chemotherapy exposure. Unlike MTX, which targets folate-dependent nucleotide synthesis, ActD interacts directly with DNA and inhibits transcription, meaning mechanisms of resistance are typically associated with reduced intracellular drug accumulation, altered cellular stress responses, or broader multidrug resistance pathways (Koba and Konopa, 2005; Assaraf et al., 2019).

One of the principal mechanisms implicated in resistance to DNA-targeting cytotoxic agents is increased expression of ATP-binding cassette transporters. These membrane transport proteins actively export a range of chemotherapeutic compounds from tumour cells, reducing intracellular drug concentration and limiting cytotoxic activity. Upregulation of multidrug resistance pathways, including drug efflux mechanisms, therefore represents an important mechanism by which tumour cells may acquire reduced sensitivity to ActD and other cytotoxic therapies (Assaraf et al., 2019).

Additional mechanisms may involve alterations in cellular responses to DNA damage. Although ActD primarily acts through inhibition of RNA synthesis, disruption of DNA-associated processes can activate pathways leading to cell-cycle arrest and apoptosis. Changes affecting DNA damage recognition, repair mechanisms, or apoptotic signalling may therefore influence cellular sensitivity to treatment, although their specific clinical importance in GTN remains less clearly defined (Koba and Konopa, 2005; Nitiss, 2009).

Clinically significant ActD resistance remains uncommon in low-risk GTN because most patients achieve remission following sequential single-agent chemotherapy. The different mechanisms of action between MTX and ActD explain why patients with inadequate response to MTX frequently remain sensitive to ActD, supporting its use as an effective second-line agent. However, a small subset of patients develop chemotherapy-resistant disease requiring alternative therapeutic approaches. Recent investigation of immune checkpoint inhibition, including avelumab, has demonstrated promising activity in selected patients with GTN resistant to conventional single-agent chemotherapy (You et al., 2020).

Overall, resistance to MTX and ActD reflects distinct but overlapping mechanisms involving altered drug handling, molecular adaptation, and tumour cell survival pathways. As these

mechanisms cannot currently be predicted routinely before or during early treatment, serial assessment of hCG response remains central to identifying ineffective therapy and guiding timely treatment modification.

1.8.4 Methotrexate Pharmacokinetics

The clinical effectiveness and toxicity profile of MTX are influenced by its pharmacokinetic properties, including absorption, distribution, metabolism, and elimination. Although the cytotoxic activity of MTX is mediated through inhibition of folate-dependent pathways, variation in drug exposure, intracellular retention, and clearance may influence therapeutic response and treatment tolerability.

In GTN, MTX is commonly administered through parenteral routes, including intramuscular administration for low-risk disease and intravenous administration as part of multi-agent chemotherapy protocols for high-risk disease. Low-risk regimens commonly include repeated-dose MTX with folinic acid rescue, whereas high-risk treatment protocols incorporate MTX within combination regimens such as EMA/CO (Seckl et al., 2013; Ngan et al., 2018). Parenteral administration avoids variability associated with oral absorption and provides more predictable systemic exposure.

Following administration, MTX is transported into cells through folate transport pathways, including the reduced folate carrier system. Once inside the cell, MTX undergoes polyglutamation, a process that increases intracellular retention and enhances inhibition of folate-dependent enzymes involved in nucleotide synthesis (Matherly and Goldman, 2003; Visentin et al., 2012). Polyglutamated MTX metabolites persist intracellularly for longer periods than circulating MTX, contributing to prolonged biological activity after plasma drug concentrations decline (Allegra et al., 1985; Visentin et al., 2012).

MTX undergoes limited hepatic metabolism, including conversion to 7-hydroxy-methotrexate. Elimination occurs predominantly through renal excretion via glomerular filtration and active tubular secretion, making renal function a major determinant of systemic drug clearance. Impaired renal elimination can result in delayed MTX clearance, prolonged exposure, and increased risk of toxicity (Widemann and Adamson, 2006; Howard et al., 2016).

Delayed clearance is particularly relevant following higher-dose MTX administration, where hydration, urine alkalinisation, and folinic acid rescue strategies are used to minimise toxicity and protect normal tissues. These approaches reduce treatment-related complications while maintaining the therapeutic effect of MTX against rapidly proliferating tumour cells (Widemann and Adamson, 2006; Howard et al., 2016).

The pharmacokinetic behaviour of MTX therefore reflects a balance between systemic exposure, intracellular accumulation, and elimination. While molecular mechanisms of resistance cannot currently be routinely assessed during GTN treatment, changes in tumour response are reflected through serial hCG measurement. Understanding the pharmacological basis of MTX activity provides important context for investigating early hCG kinetics as a marker of treatment effectiveness and potential resistance.

1.8.5 Actinomycin D Pharmacokinetics

The pharmacokinetic properties of ActD contribute to its effectiveness as both a first-line treatment option and an alternative therapy following inadequate response to MTX in low-risk GTN. Unlike MTX, which targets folate-dependent nucleotide synthesis, ActD exerts its cytotoxic effect through direct interaction with DNA, resulting in inhibition of RNA synthesis and disruption of tumour cell proliferation (Koba and Konopa, 2005).

ActD is administered intravenously, with commonly used low-risk GTN protocols including pulsed administration or multi-day treatment regimens. It is also incorporated within combination chemotherapy approaches used for high-risk disease, including EMA/CO (Seckl et al., 2013; Ngan et al., 2018). Intravenous administration is required due to its pharmacological properties and careful administration is necessary because ActD is a vesicant agent capable of causing local tissue injury if extravasation occurs.

Following administration, ActD rapidly distributes into tissues and binds strongly to DNA within nucleated cells. This interaction results in prolonged intracellular activity despite declining circulating drug concentrations. Pharmacokinetic studies have demonstrated substantial interpatient variability in ActD exposure, reflecting differences in drug distribution and clearance that may contribute to variability in toxicity profiles (Veal et al., 2005).

Unlike MTX, ActD does not require intracellular metabolic activation through polyglutamation and undergoes limited metabolism. Elimination occurs predominantly through hepatobiliary pathways, with renal clearance representing a smaller component of overall drug elimination. As a result, hepatic function has an important influence on systemic exposure and potential toxicity (Veal et al., 2005).

The distinct pharmacological characteristics of MTX and ActD provide the rationale for sequential treatment approaches in low-risk GTN. Patients who demonstrate inadequate response to MTX may remain sensitive to ActD because the two agents act through different molecular mechanisms and are influenced by different resistance pathways (Koba and Konopa, 2005; Seckl et al., 2013). Consequently, monitoring biochemical response through serial hCG measurement remains central to identifying ineffective treatment and guiding timely transition between therapeutic regimens.

1.9 Global Evidence on hCG Kinetics in Low-Risk GTD

Reliable early-phase prediction of MTX failure in FIGO low-risk GTD depends on how sensitively, and, just as importantly, how comparably, laboratories can track serial serum hCG. To inform the present investigation, a structured search of PubMed, Embase and Web of Science (January 2000 – May 2025) combined disease terms (gestational trophoblastic neoplasia/disease, low-risk) with kinetic keywords (trajectory, clearance, percent drop, half-life, prediction model) and platform trade-names (ARCHITECT, cobas, Access, VITROS, Atellica, AIA). Only peer-reviewed primary studies or systematic reviews enrolling FIGO score ≤ 6 patients and reporting at least two post-diagnosis serum hCG values were retained. Forty-three articles met inclusion, twenty-seven of which developed or validated kinetic prediction tools. Risk of bias, high in most studies because of analytical heterogeneity and small sample sizes, limitations that persist despite two decades of research.

A first, unavoidable finding from this literature is the challenge of hCG assay harmonisation, as discussed previously. Despite calibration against international reference standards, commercially available hCG immunoassays demonstrate clinically relevant variation due to differences in antibody specificity and recognition of hCG-related molecular variants, including intact hCG, free β-subunit, nicked forms, and β-core fragment (de Souza et al., 2017; McMahon et al., 2024). These analytical differences are particularly relevant in GTD, where clinical decisions rely on small changes in serial hCG concentration rather than isolated measurements.

Direct comparison of commercially available platforms has demonstrated that assay-dependent variation can influence interpretation during GTD monitoring, highlighting the importance of consistent assay use during longitudinal follow-up (de Souza et al., 2017; McMahon et al., 2024). Consequently, multicentre prediction studies must account for analytical variability, either through assay standardisation, data transformation, or validation across different platforms. Consequently, every multi-centre study in the corpus either restricted patients to a single analyser or transformed raw results (e.g., $\log_{10}$ units or multiples of the assay-specific upper reference limit) before modelling.

1.9.1 Early Descriptive Kinetics (2000 – 2010)

The earliest post-2000 literature centred on simple half-life calculations. Early investigations into hCG kinetics demonstrated that the rate of hCG decline during treatment provided clinically useful information beyond baseline concentration alone. Analysis of hCG clearance patterns showed that slower reduction in serum hCG was associated with increased likelihood of MTX resistance, establishing the principle that treatment response is better characterised by dynamic biochemical behaviour than by single measurements (You et al., 2010). Fixed time-point rules soon followed. Kerkmeijer et al. validated a single serum hCG threshold of 3 000 IU/L at week 4, predicting MTX resistance with 97.5 % specificity but only 52 % sensitivity in an independent cohort analysed on Roche cobas e-601(Kerkmeijer et al., 2009). Similar week-four rules using different cut-offs (> 20 000 IU/L on cobas; > 3 500 IU/L on Access) reproduced high specificity, but variable sensitivity once exported to a new platform, underscoring the fragility of absolute concentration thresholds.

1.9.2 Parametric Population-Kinetic Models (2010 – 2015)

A decisive methodological shift occurred when groups began modelling the entire decline curve. You *et al.* fitted a biexponential decay to serial values and extracted hCGres, the modelled residual daily production after six weeks. In London and Lyon datasets analysed on different assays, hCGres predicted MTX failure with > 90 % sensitivity and 83 % specificity, illustrating that shape-based metrics travel better than raw value (You et al., 2013). A subsequent analysis of 154 low-risk patients confirmed that clearance-rate constants outperformed baseline hCG as independent risk factors for escalation (Phianpiset et al., 2022). Nevertheless, these parametric tools still required ≥ 4 post-diagnosis samples, limiting their usefulness for clinicians who often prefer decisions sooner.

1.9.3 Nomograms and Machine-Learning (2016 – present)

Leveraging larger electronic registries, recent studies have aimed for earlier prediction horizons. Hoeijmakers and colleagues built a logistic nomogram combining a single week-four hCG value with baseline clinical variables; at 97.5 % specificity it identified two-thirds of post-molar progressors a fortnight sooner than conventional FIGO criteria (Hoeijmakers et al., 2021). However, external validation on Siemens Atellica data required recalibration of the intercept, again reflecting between-platform bias. Machine-learning (ML) approaches promise to capture non-linear patterns. A 2022 Chinese study applied a random-forest algorithm to the first three weekly hCGs and achieved an AUC of 0.92, surpassing logistic regression in the same cohort. Decision-tree models incorporating pre-treatment hCG concentration and the day-7/day-0 ratio show similar discrimination (Jin et al., 2024). Yet screening reveals a high risk of over-fitting in these ML reports: sample sizes rarely exceed 150 patients, internal cross-validation is the norm, and no study has yet demonstrated impact on real-time clinical decision-making. In addition, approaches that require four or more serial hCG measurements to achieve reliable discrimination inherently delay prediction, thereby delaying actionable prediction beyond the early post-evacuation window when intervention decisions are most clinically valuable.

1.9.4 Synthesis and Implications for the Present Thesis

Three messages recur across this 25-year window. First, trajectory outperforms static values. Parameters encoding early velocity, percent drop, half-life, hCGres, consistently surpass fixed week-four thresholds at discriminating future MTX failure. Second, assay heterogeneity blunts generalisability. Even refined models drift when transplanted to a platform with a different cross-reactivity profile, unless values are transformed or otherwise normalised. Third, prospective evidence is scarce. None of the 27 modelling studies identified had yet been tested in a randomised decision-impact design; the field remains largely retrospective and single-centre. The present thesis addresses these gaps by modelling the full serial serum hCG trajectory within a uniquely harmonised U.K. cohort measured on a single calibration chain (RIA / IMMULITE 2000), then gauging how soon robust predictions can be issued. By eliminating the dominant source of analytical noise, platform drift, this work provides a stringent benchmark for trajectory-based and ML frameworks before wider multi-platform deployment.

1.10 Hypothesis and Aims

The central hypothesis of this research is that distinct hCG profiles exist among patients with GTD and GTN, which can be used to predict disease behaviour and treatment requirements at an earlier stage, thereby supporting more individualised patient management. This overarching hypothesis gives rise to the following specific hypotheses and aims:

1.10.1 Hypothesis 1

Patients with naturally resolving GTD follow a characteristic hCG decline profile that distinguishes them from those requiring treatment.

Aim: To define and characterise hCG levels and trajectory in patients with naturally resolving disease, and to validate these profiles using testing and validation cohorts to ensure consistency within the studied population.

1.10.2 Hypothesis 2

hCG profiles can differentiate patients who require MTX from those with self-resolving disease.

Aim: To define and characterise hCG profiles in patients requiring MTX and assess their predictive value for treatment necessity by analysing the rate of change, curve characteristics, and potential thresholds. The distribution of hCG-derived features will be examined to determine whether patient groups demonstrate clear separation suitable for threshold-based stratification or whether probabilistic modelling approaches are required.

1.10.3 Hypothesis 3

Patients who demonstrate inadequate response to MTX exhibit distinct hCG trajectories that can predict the requirement for ActD.

Aim: To define and characterise the hCG profiles of patients requiring escalation from MTX to ActD, and to validate these findings using independent validation cohorts. The distribution of predictive features will be assessed to determine whether clear threshold-based stratification is achievable or whether overlapping distributions require probabilistic modelling approaches

1.10.4 Hypothesis 4

A predictive model based on hCG trajectories can enhance early identification of patients requiring MTX and ActD.

Aim: To develop predictive models for (a) identifying patients requiring MTX treatment and (b) predicting inadequate MTX response requiring ActD escalation. Model performance will be assessed using positive predictive value (PPV), area under the curve (AUC) analysis, sensitivity, specificity, and other clinically relevant measures of predictive performance.

2.0 Methods Chapter

The methodological strategy was structured to address four central hypotheses concerning the early biochemical behavior of GTD and its relationship to treatment outcome. Two complementary datasets were analysed: a urine dataset consisting of log-transformed hCG/Creatinine ratios from patients monitored post-evacuation, and a serum hCG dataset comprising sequential early measurements from patients treated with MTX. **Hypothesis 1**, which proposed that patients with naturally resolving GTD follow a characteristic hCG trajectory, was addressed by developing logistic regression, gradient boosting, and random forest classifiers on the urine dataset to distinguish patients with normalising versus abnormal patterns of decline (see Section 2.1.1). **Hypotheses 2 and 3** were tested using the serum dataset, which included both MTX responders and patients who escalated to Actinomycin D. These hypotheses evaluated whether early hCG kinetics could differentiate treatment responders from those requiring escalation. **Hypothesis 4** proposed that predictive modelling of early hCG trajectories, using supervised machine learning, could enhance early identification of treatment escalation risk. To test this, the same classification models (logistic regression, gradient boosting, and random forest) were applied consistently across both urine and serum datasets. Model performance was evaluated using standard metrics including accuracy, AUC, precision, recall, F1 score, and negative predictive value. All data preparation and model development were conducted in Python using the Google Colab platform, with separate pipelines for each dataset.

2.1 Data Sources

This study utilised two retrospective clinical datasets obtained from the Sheffield Trophoblastic Disease Centre at Sheffield Teaching Hospitals NHS Foundation Trust (STH). The datasets represented distinct stages of the GTD clinical pathway and were selected to address the specific hypotheses of this thesis. The urine hCG dataset was used to investigate early hCG regression patterns following molar pregnancy surveillance, while the serum hCG dataset was used to evaluate treatment response and identify predictors of MTX resistance requiring escalation to ActD.

Both datasets consisted of routinely collected clinical monitoring data generated as part of standard GTD management. Prior to analysis, data underwent cleaning, preprocessing, and feature engineering to create structured datasets suitable for statistical analysis and machine learning model development.

2.1.1 Urine Dataset (Hypothesis 1 and 2)

The urine hCG dataset initially contained 12,972 anonymised patient records obtained from the clinical archives at Sheffield Teaching Hospitals NHS Foundation Trust. Each record corresponded to a case of suspected or confirmed GTD, and contained serial urinary hCG measurements, urine creatinine measurements, and a novel hCG/ creatinine normalised ratio. These data were originally collected as part of routine post-molar pregnancy surveillance and reflect the natural history of hCG elimination in a predominantly conservatively managed cohort. This dataset originates from the Sheffield national GTD referral centre, which operates within the centralised UK trophoblastic disease surveillance framework (Tidy et al., 2021). A distinctive feature of the Sheffield programme is the routine use of urinary hCG monitoring, with hCG/creatinine ratios used as a primary biomarker for post-molar follow-up. This differs from centres where serum hCG is the principal monitoring modality, and results in a uniquely detailed longitudinal urine dataset. While this enables evaluation of early hCG kinetics in a minimally invasive monitoring context, it also reflects centre-specific practice, which should be considered when interpreting model generalisability. Given the variability in timing, completeness, and follow-up intervals inherent to real-world clinical data, substantial preprocessing was necessary to identify a cohort appropriate for hypothesis-driven modelling. Initially, all rows with missing values for creatinine, hCG, or hCG/creatinine ratio were excluded. This cleaning step resulted in a refined dataset comprising 156,886 individual urine hCG ratio measurements derived from these 12,972 patients. To capture only the relevant early trajectory of hCG change, which was critical for distinguishing cases likely to resolve spontaneously from those requiring treatment, data for each patient were further filtered to retain only the first uninterrupted sequence of results. Specifically, any monitoring episode that included a gap of more than 30 days between sequential samples was truncated at the point of interruption. This approach aimed to isolate the initial active phase of surveillance, excluding late follow-up results that typically occur after remission and are not predictive of early treatment outcomes. To ensure adequate temporal resolution for decline pattern analysis, patients with fewer than three results in their initial monitoring episode were excluded. This criterion removed 4,036 individuals, leaving 8,936 patients with sufficient data to assess early hCG dynamics. Patients were then stratified into two groups based on the final hCG ratio recorded within their initial monitoring sequence. A patient was classified as normal (i.e., presumed spontaneous resolution) if their final ratio was ≤ 0.35, aligning with established local thresholds for biochemical remission. In contrast, patients whose final ratio remained > 0.35 were considered abnormal, suggesting persistent disease activity or incomplete regression within the episode studied. Following this classification, 7,276 patients

(81.4%) met criteria for the normal group, while the remaining 1,660 (18.6%) were classified as abnormal. These two groups provide the basis for testing Hypothesis 1, which proposes that patients whose disease resolves without the need for chemotherapeutic intervention follow a characteristically different hCG decline trajectory compared to those with persistent or escalating disease.

2.1.1.1 Urine Dataset: Feature Engineering

Following the cohort definition described above, a second phase of data preparation was undertaken to create analytic features capable of quantifying the shape, speed and magnitude of early urinary hCG decline. As only the first three observations (t_1...t_3) were extracted to balance practical clinic scheduling (weekly visits) with the clinical aim of early-stage risk stratification. Patients whose t_1 ratio was already ≤ 0.35 and who remained ≤ 0.35 at t_2 and t_3 were excluded at this stage (n = 4 086) because they manifest no discernible decline trajectory; retaining them would artificially inflate specificity without interrogating the hypothesis. It was then this dataset that would be used for modelling. The first process involved manually generating a structured feature set in Excel that quantitatively captured the direction, rate, and consistency of change in the $\log_{10}$(hCG/Creatinine) ratio over the first three observed time points in each patient's initial monitoring episode.

1. Raw Ratio Values

- 'Ratio_1': First $\log_{10}$(hCG/Creatinine) ratio recorded after diagnosis.
- 'Ratio_2': Second ratio, typically recorded approximately one week after the first.
- 'Ratio_3': Third ratio, typically recorded two weeks after the first.

These values reflect the absolute biochemical burden (after urine dilution correction) at the start of follow-up and are retained for direct modelling or feature transformation.

2. Kinetic Features (Rate of Change)

- 'Slope_1_2': Change in log-ratio between Ratio_1 and Ratio_2.
- 'Slope_2_3': Change in log-ratio between Ratio_2 and Ratio_3.
- 'Slope_1_3': Overall change in log-ratio from Ratio_1 to Ratio_3, divided by the number of days between them (to account for variable intervals).
- 'Mean_Slope': Average of the two individual slopes above: (Slope_1_2 + Slope_2_3) / 2.

These slopes reflect whether urinary hCG/Creatinine is rising, falling, or plateauing across the early follow-up period.

3. Trajectory Shape Features

- 'Curvature': Difference between Slope_2_3 and Slope_1_2. This captures whether the trajectory is accelerating or decelerating.
- 'Slope_Change_Sign': A binary flag (0 or 1) indicating whether the direction of slope reversed between time intervals, i.e., if Slope_1_2 and Slope_2_3 have opposite signs.

These features help distinguish monotonic declines from erratic or rebound profiles.

4. Descriptive Summary Features

- 'Mean_Ratio': Arithmetic mean of the three ratio values.
- 'Min_Ratio': Lowest ratio value observed among the three.
- 'Max_Ratio': Highest ratio value among the three.
- 'Delta_MaxMin': Difference between Max_Ratio and Min_Ratio, capturing fluctuation range.

These statistics summarise the magnitude and variability of the patient's early trajectory.

5. Timing Features

- 'Days_1_2': Number of days between the first and second urine collections.
- 'Days_2_3': Number of days between the second and third urine collections.
- 'Days_1_3': Total number of days between the first and third samples.

These timing variables were used to contextualise slope and pattern behaviour in patients with irregular sample spacing.

6. Outcome Variable

- 'Escalated': A binary target variable indicating whether the patient ultimately required further treatment.
    - 0 = The patient resolved spontaneously (final ratio ≤ 0.35).

- 1 = The patient's ratio never dropped below 0.35 or rebounded before the end of the first monitoring sequence.

2.1.2 Serum Dataset (Hypothesis 3 and 4)

The serum dataset comprised 332 anonymised GTD patient records obtained from STH clinical archives. Each patient was initially treated with MTX and either responded successfully or required escalation to ActD. Only patients with complete serial hCG data from Day 0 to at least Day 28 were included. All patients had quantitative serum hCG values measured at predefined intervals, typically Day 0 (start of treatment), Day 14, and Day 28.
Patients were divided into two categories based on treatment outcome:

- MTX Cure (Escalated = 0): Patients whose hCG levels fell satisfactorily, requiring no further treatment beyond MTX.
- Escalated to ActD (Escalated = 1): Patients whose hCG levels plateaued, rebounded, or rose despite MTX, prompting second-line intervention

This curated serum dataset served as a structured comparator for the urine hCG dataset, allowing hypothesis testing around early biochemical indicators of treatment failure and the potential to predict therapeutic escalation based on hCG trajectory.

2.1.2.1 Serum Dataset: Feature Engineering

To support predictive modelling, a set of derived variables was engineered from the serum hCG data. Initial feature generation was performed manually in Microsoft Excel, allowing for iterative review and domain-informed adjustments. These features were subsequently validated, standardised, and restructured for analysis in Python. Each patient record was ultimately summarised using thirteen features, organised into the following groups:

1. Raw hCG Values:

- 'Day0_hCG': Quantitative serum hCG on the day MTX was initiated.
- 'Day14_hCG': Value after the second treatment at 14 days.
- 'Day28_hCG': Value after the third treatment at 28 days.

2. Kinetic Features (Rate of Change):

- Slope_0_14: Slope between Day 0 and Day 14.
- Slope_14_28: Slope between Day 14 and Day 28.
- Slope_0_28: Overall slope across all three time points.

These slopes approximate the rate at which hCG changes and are useful in detecting stagnation or rebound patterns.

3. Percentage Drop Features:

- 'Percent_Drop_0_14': Percentage drop from Day 0 to Day 14.
- 'Percent_Drop_14_28': Percentage drop from Day 14 to Day 28.
- 'Percent_Drop_0_28': Overall percentage drop from Day 0 to Day 28.

These features account for proportional decline and help identify patients with incomplete metabolic response.

4. Ratios:

- 'Ratio_0_to_14': Day 14 value divided by Day 0.
- 'Ratio_0_to_28': Day 28 value divided by Day 0.

These normalised ratios highlight the magnitude of change without requiring the scale of absolute hCG values.

5. Binary Rebound Flag:

- 'Rebound_14_28': Boolean feature (1 or 0) indicating whether Day 28 hCG exceeded Day 14.

6. Minimum hCG:

- 'Min_hCG_0_28': Minimum value among Day 0, 14, and 28 hCG.

7. Outcome Variable:

A binary outcome variable is necessary for modelling so the 'Escalated' column was created. This variable indicated whether a patient needed further treatment beyond the initial course of MTX. If a patient responded well to MTX and did not require additional therapy, they were given a

value of 0. If the patient's hCG levels failed to drop as expected and they went on to receive ActD, they were given a value of 1. This classification was used as the target for all predictive analysis, allowing models to learn patterns associated with MTX treatment success or ActD escalation.

2.2 Data Acquisition and Computational Environment

The modelling was performed using Google Colaboratory (Colab), a cloud-based environment that allows Python code to be executed without requiring local installation of any software. This was selected as most of the data modelling was conducted on an NHS Trust Laptop where installation of software and repositories requires several layers of IT authorisation and can cause time delays. It was determined that a cloud-based environment would streamline the process without any negative consequences. A private Google Colab environment was used, and only anonymised datasets were uploaded, providing an additional level of data security.

1. Go to: https://colab.research.google.com/

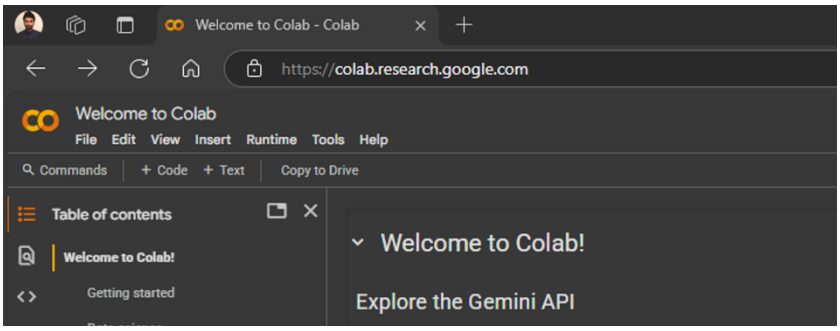


Figure 2.1: Google Colaboratory domain interface shown at initial access (https://colab.research.google.com). Source: Author's own screenshot.

2. Click New Notebook in Drive from the File menu
3. The following code was inserted into the new notebook to facilitate the upload of the refined hCG Dataset:

from google.colab import files *(Import Colab's file handling function)*

files.upload() *(Opens a browser allowing manual file selection)*

Important note:

Throughout the remainder of the methods section, each line of Python code will be presented in **blue bold**, with an explanation in *green bracketed italics* beside it. Screenshots of the code

as it appears in Python, along with preceding and/or following cells (where relevant), will also be included to show how the code is displayed in Python.

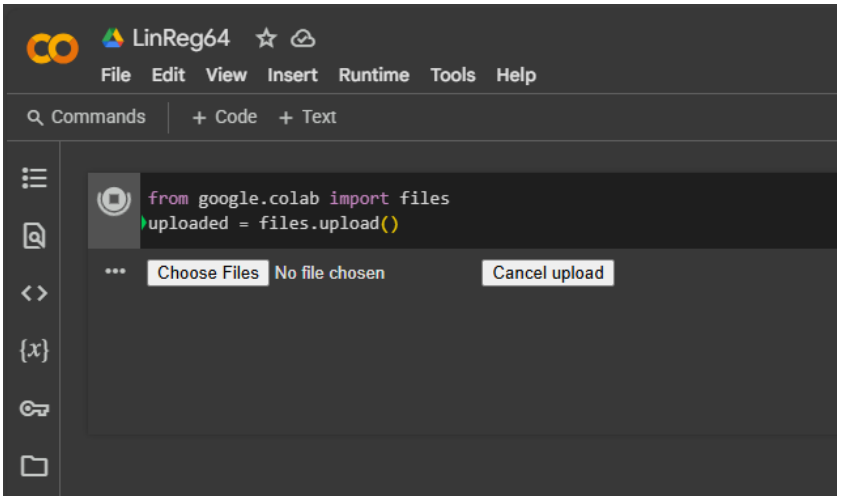


Figure 2.2: Uploading files to Google Colab using files.upload() to manually select the dataset. Source: Author's own Python code executed in Google Colab.

4. The hCG dataset was then chosen, either the urine hCG data set, or the serum hCG dataset depending on the modelling, which was then uploaded to the Google Colab database

2.2.1 Data Loading and Initial Inspection

Once uploaded, the dataset was imported into Python using the Pandas library and stored in a structure known as a DataFrame, using the following code:

```
import pandas as pd          (Imports the pandas library for data handling)
df = pd.read_csv('...')      (Reads the uploaded CSV file into a DataFrame)
df.head()                    (Displays the first five rows of the loaded DataFrame)
```

The Pandas library format is used for working with tabular data and offers a user friendly, spreadsheet-like structure that simplifies both inspection and analysis. Using a DataFrame made it easier to explore the dataset, identify potential formatting issues, and prepare the data for later modelling steps.

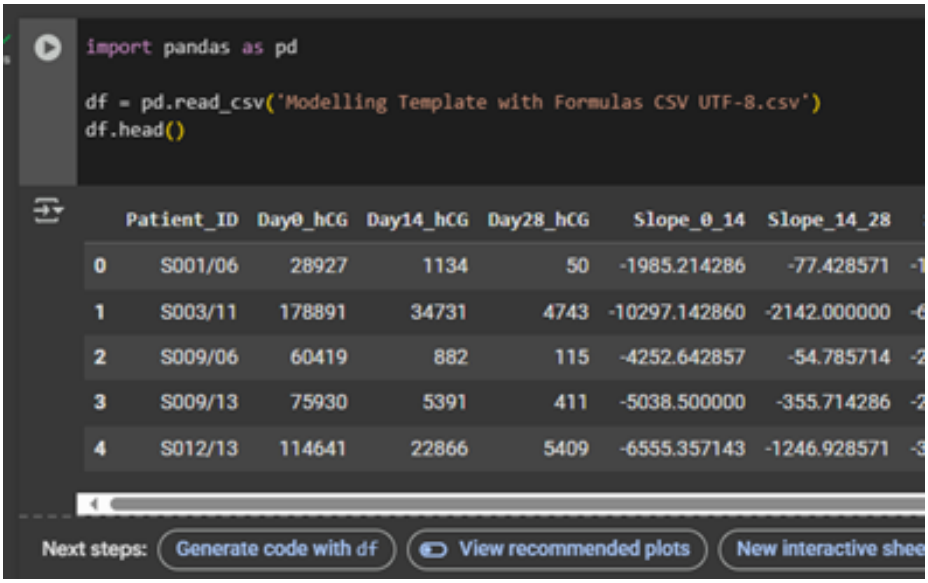


Figure 2.3: Initial dataset preview using df.head() to verify successful import of hCG data. This dataset would differ depending on if the serum dataset is used, or the urine dataset.
Source: Author's own Python code.

Pandas was chosen because it provides a range of efficient tools for cleaning, manipulating, and analysing structured datasets, supporting rapid exploration and preparation of large tabular datasets (McKinney, 2010). For example, it allowed for quick filtering of rows, selection of relevant columns, and identification of missing or inconsistent values that could be missed in excel due to the volume of patient data. Importantly, the DataFrame format integrates effectively with machine learning libraries such as scikit-learn, allowing structured datasets to be efficiently processed and converted into formats suitable for model development (Pedregosa et al., 2011). By using simple commands such as .head(), it was possible to view the first few records and confirm that the data had been loaded correctly, which was important as one of the downsides to modelling in Google Collab is there is no visual confirmation of data once uploaded. Overall, loading the dataset into a Pandas DataFrame was an essential first step that allowed for clear organisation of the data and formed the foundation for all subsequent stages of analysis across all models.

2.3 Logistic Regression Modelling – Serum Dataset

This section describes the development of a logistic regression model aimed at predicting escalation from MTX to ActD in patients with GTD. Logistic regression was selected because the outcomes of interest, escalation or non-escalation, was a binary outcome (yes/no). Logistic regression models are well-suited to binary classification problems and allow interpretation of how each predictor influences the odds of the outcome occurring. Rather than selecting variables manually based on clinical intuition alone, an objective feature selection process was performed to identify predictors with statistically meaningful contributions to model performance. This approach was intended to enhance scientific rigour and avoid bias. It was anticipated that clinically important variables, particularly those reflecting early hCG levels and rates of decline, would emerge as significant predictors during feature selection. If confirmed, this would support the clinical goal of enabling earlier risk stratification and intervention based on initial treatment response data.

2.3.1 Data Upload in Google Colab

As per 2.2 above

2.3.2 Data Loading and Inspection

As per 2.2.1 above

2.3.3 Predictor and Target Definition

All available numerical variables were initially retained, while text-based or identifier columns were excluded using the following code:

```
[6]  df_model = df.drop(columns=['Patient_ID', 'Trajectory_Type']).dropna()

     X = df_model.drop(columns=['Escalated'])  # Predictor variables
     y = df_model['Escalated']  # Target variable
```

Figure 2.4: Code dropping non-predictive columns and separating predictors X and target y. Model: Serum Logistic Regression. Source: Author's own Python code.

The columns dropped were 'Patient_ID' and 'Trajectory_Type' as they are not suitable as numerical predictors (one is an identifier, the other categorical text) – without this step, the model continued to crash. The predictors (X) are all the variables that could help the model make a prediction, as discussed in section 2.1.2.1 above. The target (y) is whether the patient escalated to ActD (1) or remained on MTX (0), this seperation was necessary to ensure that machine modelling was conducted on the correct patient subset.

2.3.4 Train–Test Split

Before developing any predictive model, it is essential to divide the available data into two separate parts: a training set and a testing set. The training set is used to build the model by allowing it to learn patterns from known outcomes. The testing set, by contrast, is held back and used only to evaluate how well the model performs on data it has never seen – which is critical when considering how the model might behave in a real-world clinical setting.

The dataset was split into training and testing subsets in an 80/20 ratio within Python using the following code:

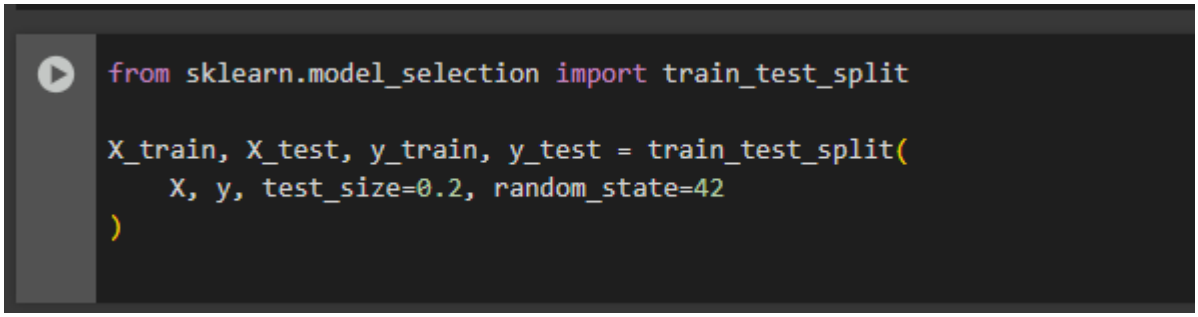
from sklearn.model_selection import train_test_split *(Imports the `train_test_split` function from the `sklearn` (scikit-learn) library)*

X_train, X_test, y_train, y_test = train_test_split(

X, y, test_size=0.2, random_state=42 *(These line of code runs the split. The X and y variables were defined in 2.3.3 above. The model test size is specified as 0.2 (20%). The training size is therefore 0.8 (80%))*

In clinical data modelling, where overfitting is a common risk and generalisability is essential, the integrity of this separation becomes particularly important. Reserving part of the data exclusively for testing ensures that the model's performance is not only optimised for retrospective accuracy but also holds potential value when applied prospectively to new patient cases. The partitioning step, while technically straightforward, plays a critical role in establishing both methodological credibility and practical relevance. An 80/20 split was used, that is, 80% of the data was allocated for training and the remaining 20% reserved for testing. This ratio is commonly used in machine learning because it provides a good balance between allowing the model sufficient data to learn from while still retaining enough unseen data to test generalisability (Hastie et al., 2009; Pedregosa et al., 2011). To ensure reproducibility, a fixed random seed (random_state=42) was

used during the split. Setting a seed means that the same data will be allocated to the training and testing sets each time the process is repeated. While the number 42 is often used in data science as a reference to The Hitchhiker's Guide to the Galaxy (Nafis, 2021), the specific number is arbitrary, what really matters is that it remains consistent (Pedregosa et al., 2011).



```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Figure 2.5: Code separating Training and Testing Groups.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.5 Predictor Scaling

Before training the model, all predictor variables (X) were standardised using a method called StandardScaler. This technique adjusts each variable so that it has a mean of zero and a standard deviation of one, which helps ensure that all variables are on a similar scale (Hastie et al., 2009).

This was performed using the following code:

from sklearn.preprocessing import StandardScaler *(Imports the ` StandardScaler` tool from the ` sklearn` (scikit-learn) library)*

scaler = StandardScaler() *(This line sets up the scaler, preparing it for use)*

X_train_scaled = scaler.fit_transform(X_train) *(scaler.fit(X_train) calculates mean and SD of each feature in the training set. transform(X_train)` uses those values to scale the training data)*

X_test_scaled = scaler.transform(X_test) *(Does the same for the testing data set)*

This StandardScaler step is important for several reasons. First, logistic regression (and many other machine learning methods) works best when all input variables are roughly within the same range. If some values, such as hCG measurements, are much larger than others, the model can become biased and focus too heavily on those features (Hastie et al., 2009).

Standardising the variables also helps the model learn more efficiently. It allows the algorithm to move through the data more smoothly during training, making the optimisation process faster and more stable. Crucially, scaling is applied only to the training data first, and then the same

transformation is applied to the test data. This avoids what's known as data leakage, where information from the test set might accidentally influence the model during training (Pedregosa et al., 2011).

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 2.6: Standardisation of numerical features using StandardScaler. Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.6 Baseline Logistic Regression

Once the predictor variables had been standardised, a logistic regression model was trained on the scaled dataset using the following code:

from sklearn.linear_model import LogisticRegression *(This line imports the logistic regression model from scikit-learn's linear_model module)*

logreg = LogisticRegression(max_iter=1000, random_state=42) *(Logistic regression model is created with a maximum of 1000 training steps. Usullay this is set to 100, but as this is complex data, increasing iterations alows the model to learn proprerly and settle on a good solution)*

logreg.fit(X_train_scaled, y_train) *(This line trains the data on the standardised training data set. The .fit() method applies the logistic regression algorithm, using the input features (X_train_scaled) to estimate the relationship with the target labels (y_train), which indicate whether each patient was escalated or not)*

Logistic regression is suited to binary classification problems and was chosen here because of its simplicity, interpretability and its established use in clinical prediction tasks (Steyerberg, 2019).

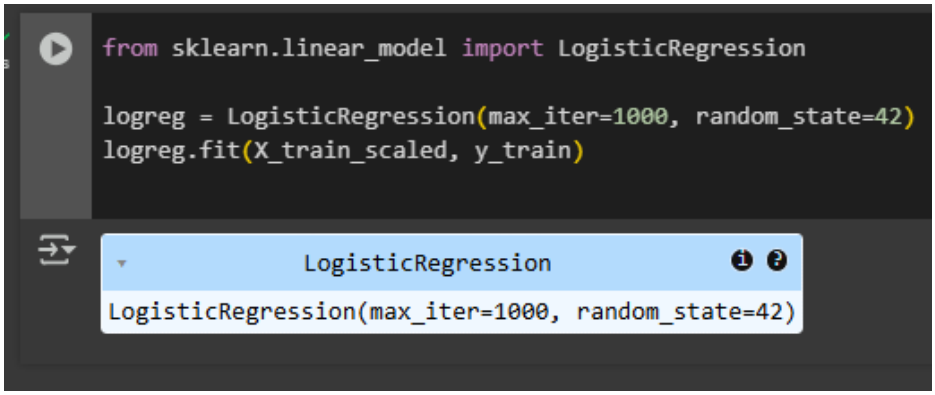


Figure 2.7: Code training the model on the scaled dataset.

Model: Serum Logistic Regression. Source: Author's own Python code.

The method works by modelling the relationship between the input variables and the outcome (escalation or not) using the log-odds scale. This allows the contribution of each predictor to be interpreted in terms of its relative effect on the likelihood of escalation, which is particularly useful in a clinical research setting (Schober and Vetter, 2021). The parameter max_iter=1000 was specified to ensure that the algorithm had sufficient opportunity to converge. While convergence is more reliable following standardisation, increasing the maximum iteration limit adds a layer of robustness and avoids premature termination of the optimisation process. This is especially important when working with multiple predictors that may exhibit subtle correlations or when attempting to capture signal in a dataset with moderate complexity.

2.3.7 Coefficient-Based Feature Selection

To improve model interpretability and reduce unnecessary complexity, a Feature Selection option was applied, and guided by the magnitude of each variable's coefficient in the fitted logistic regression model. In this context, variables with larger absolute coefficients were assumed to exert greater influence on the predicted outcome. A cut-off value of 0.01 was applied, whereby any feature with a coefficient below this threshold was considered to have minimal practical impact and was excluded from the final model. The following Python code was run to implement this feature:

import numpy as np *(Imports NumPy, a library used for working with numbers and arrays in Python. In this case, it helps take the absolute values of the model coefficients)*

coefs = np.abs(logreg.coef_[0]) *(This line extracts the logistic regression coefficients and converts them into absolute values. Logreg.coef contains the model's learned coefficients for*

each feature, the '0' accesses the coefficient array directly, and np.abs() ensures we're only interested in the size of each effect, not whether it is positive or negative)

threshold = 0.01 *(cut-off value of 0.01 is defined. Features with absolute coefficients greater than 0.01 will be kept; others are considered too small to be meaningful)*

important_features = [feature for feature, coef in zip(X.columns, coefs) if coef > threshold]

(This creates a list of features that have a strong enough influence on the model, zip(X.columns, coefs) pairs each feature name with its coefficient. The list comprehension checks whether each coefficient is above the threshold, and if so, includes the feature in the final list)

print("Selected Features:", important_features) *(this line prints out the names of the features that were selected based on the threshold)*

This approach offers a straightforward way to identify and retain only those predictors that meaningfully contribute to the model's decisions. By removing variables with negligible effect sizes, the resulting model becomes simpler, more transparent, and easier to interpret, which is particularly important in clinical applications where clarity of reasoning is essential (Bagherzadeh-Khiabani et al., 2016).

```
import numpy as np

coefs = np.abs(logreg.coef_[0])

threshold = 0.01
important_features = [feature for feature, coef in zip(X.columns, coefs) if coef > threshold]

print("Selected Features:", important_features)
```

```
Selected Features: ['Day0_hCG', 'Day28_hCG', 'Slope_0_14', 'Slope_14_28', 'Slope_0_28', 'Perc
```

Figure 2.8: Code to generate Feature Selection.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.8 Refined Logistic Regression

Following Feature Selection, the logistic regression model was retrained using only the subset of predictors that had demonstrated meaningful influence in the initial model using the following Python Code:

X_train_imp = scaler.fit_transform(X_train[important_features]) *(This line scales only the selected important features from the training set. X_train[important_features] selects the reduced set of columns and scaler.fit_transform() standardises the data by first calculating the mean and standard deviation, then applying the transformation)*

X_test_imp = scaler.transform(X_test[important_features]) *(This applies the same scaling to the test data, using the values (mean and SD) learned from the training data. It ensures consistency between training and testing without introducing data leakage)*

logreg_reduced = LogisticRegression(max_iter=1000, random_state=42) *(This creates a new logistic regression model, with max_iter=1000 (explained previously and random_state=42 (explained previously))*

logreg_reduced.fit(X_train_imp, y_train) *(This trains the model using only the selected features from the scaled training data. The model learns how these important variables relate to the outcome (e.g., escalation or not))*

This step ensures that the final model is informed solely by variables with substantive predictive value, improving its interpretability and reducing unnecessary complexity. By excluding features with negligible impact, the risk of overfitting is reduced, and the model becomes more transparent and easier to communicate which is an important consideration when applying the results in a clinical context (Chowdhury and Turin, 2020).

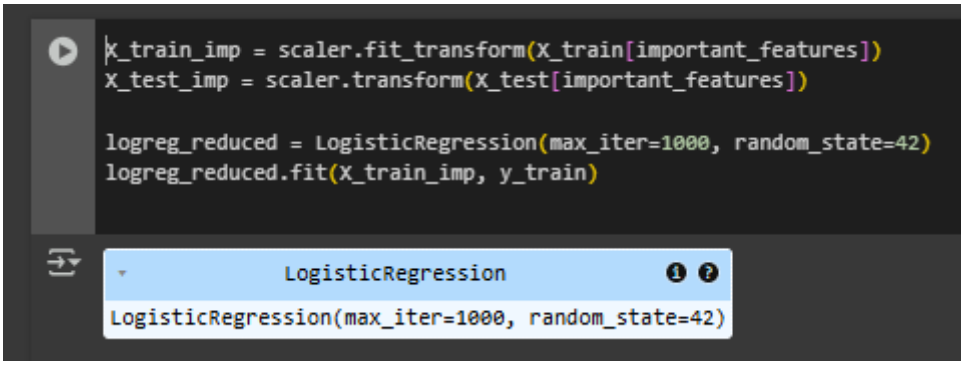


Figure 2.9: Code retraining the model based on Feature Selection.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.9 Evaluation Metrics

The performance of the final logistic regression model was evaluated using a range of metrics. Accuracy was used as a basic measure of overall correctness, representing the proportion of all predictions (both positive and negative) that matched the actual outcomes. However, given the potential clinical impact of false positives and false negatives, additional metrics were included

to provide a more nuanced assessment. The area under the receiver operating characteristic curve (AUC) was calculated to quantify the model's ability to distinguish between patients who required escalation and those who did not. AUC reflects the probability that the model will correctly rank a randomly chosen escalated case above a non-escalated one. Precision was reported to assess the reliability of positive predictions, specifically, the proportion of patients predicted to require escalation who did. Recall (or sensitivity) was also calculated to determine how well the model identified true escalation cases among all patients who required further treatment. Finally, the F1 score was included to provide a balanced summary of both precision and recall. This harmonic mean is particularly useful in scenarios where an imbalance between false positives and false negatives could influence clinical decision-making. Together, these metrics offer a more comprehensive view of model performance than accuracy alone and better reflect the practical consequences of prediction errors in a clinical setting (Saito and Rehmsmeier, 2015; Steyerberg, 2019).

To obtain the above metrics, the following Python code was run:

from sklearn.metrics import accuracy_score, roc_auc_score, precision_score, recall_score, f1_score, confusion_matrix *(This line imports several commonly used evaluation metrics from scikit-learn)*

y_pred = logreg_reduced.predict(X_test_imp) *(Generates the predicted class labels (0 or 1) for the test data. These are the model's binary decisions: whether each patient is predicted to require escalation or not)*

y_probs = logreg_reduced.predict_proba(X_test_imp)[:, 1] *(This inputs the predicted probability of each patient being in class 1 (i.e., requiring escalation). The model generates a probability for both classes, and [:, 1] selects the values for the positive class)*

accuracy = accuracy_score(y_test, y_pred) *(This calculates the accuracy, the proportion of all predictions (both 0 and 1) that the model got right)*

auc = roc_auc_score(y_test, y_probs) *(This calculates AUC of the ROC curve, which reflects the model's ability to rank true positives above true negatives across different classification thresholds)*

precision = precision_score(y_test, y_pred) *(This measures precision, the proportion of predicted escalations that were actually correct. It reflects how trustworthy the model's positive predictions are)*

recall = recall_score(y_test, y_pred) *(Calculates recall, also known as sensitivity — the proportion of actual escalations that the model correctly identified)*

f1 = f1_score(y_test, y_pred) *(Calculates the F1 score, which combines precision and recall into a single value using their harmonic mean. It's helpful when you want a balance between avoiding false positives and false negatives)*

cm = confusion_matrix(y_test, y_pred) *(This creates a confusion matrix, which breaks down the prediction outcomes into: TN, FP, FN, and TP)*

tn, fp, fn, tp = cm.ravel() *(This line unpacks the four numbers in the confusion matrix into individual variables so they can be used in calculations)*

npv = tn / (tn + fn) *(This calculates the Negative Predictive Value (NPV) — the proportion of predicted non-escalations that were correct. It's useful in clinical settings where reassurance about low-risk cases is important)*

print(f"Accuracy: {accuracy:.3f}")

print(f"AUC: {auc:.3f}")

print(f"Precision: {precision:.3f}")

print(f"Recall (Sensitivity): {recall:.3f}")

print(f"F1 Score: {f1:.3f}")

print(f"Negative Predictive Value (NPV): {npv:.3f}")

(These above lines display the performance metrics with results rounded to three decimal places, making it easy to read and interpret the model's behaviour)

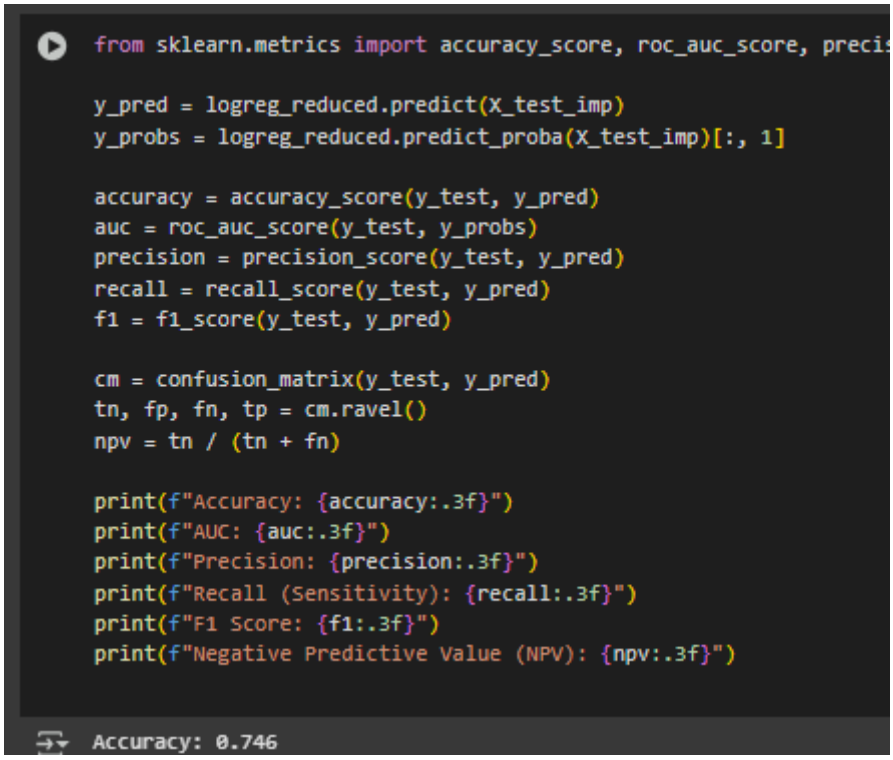


Figure 2.10: Code to generate the Model Evaluation.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.10 Confusion Matrix Visualisation

The model's performance was further assessed using a confusion matrix, which was both computed and visualised to provide a clear breakdown of prediction outcomes using the following lines of Python code:

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay *(This line imports two tools from scikit-learn, confusion_matrix that calculates the matrix of predicted vs actual outcomes, and ConfusionMatrixDisplay which provides an easy way to visualise the matrix as a plot within Python)*

cm = confusion_matrix(y_test, y_pred) *(This generates the confusion matrix based on the actual labels (y_test) and the model's predictions (y_pred))*

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['No Escalation', 'Escalation']) *(This prepares the visual display of the matrix. display_labels sets custom axis labels)*

disp.plot(cmap='Blues')

plt.title('Confusion Matrix - Logistic Regression Reduced Model')

plt.grid(False)

plt.show()

(The above code uses the blue colour scheme, adds a title, removes gridlines, and then displays in the output cell in Google Colab)

The confusion matrix summarises model predictions by comparing predicted and observed outcomes, categorising results into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Visualisation of the confusion matrix provides an intuitive representation of classification performance, allowing clear identification of correct predictions and error types across classes (Steyerberg, 2019; Saito and Rehmsmeier, 2015).

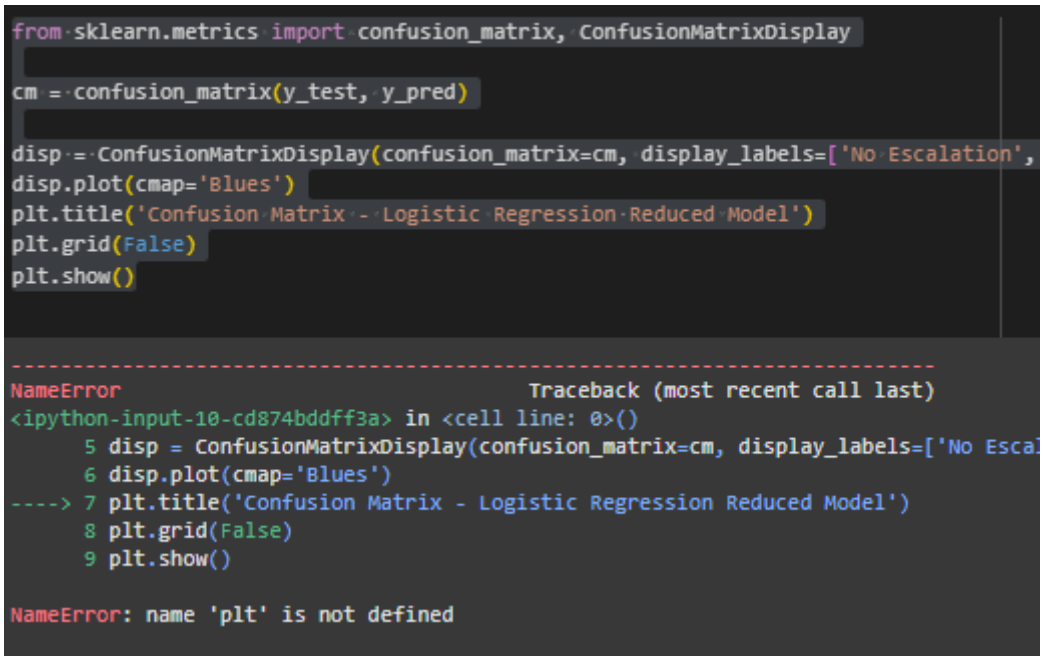


Figure 2.11: Code to display the Confusion Matrix.
Model: Serum Logistic Regression. Source: Author's own Python code.

A minor implementation issue related to missing library import (matplotlib.pyplot) was identified during visualisation; however, this did not affect the underlying confusion matrix or any model performance metrics. All required libraries were subsequently explicitly imported to ensure reproducibility.

2.3.11 ROC Curve

To further assess the model's ability to discriminate between patients who required escalation and those who did not, a ROC curve was plotted using the following Python code:

from sklearn.metrics import roc_curve *(This imports the roc_curve function from scikit-learn.)*

import matplotlib.pyplot as plt *(This imports the matplotlib.pyplot module for creating plots. It's commonly abbreviated as plt and used to control figure layout, titles, labels, and other visual elements)*

fpr, tpr, thresholds = roc_curve(y_test, y_probs) *(This line calculates the data points needed to draw the ROC curve. fpr = false positive rate (x-axis), tpr = true positive rate or sensitivity (y-axis), and thresholds = the probability cut-off values used to compute each point on the curve)*

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkblue', lw=2, label='ROC curve (AUC = {:.2f})'.format(auc))

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve - Logistic Regression Reduced Model')

plt.legend(loc="lower right")

plt.grid(True) *(The above code draws the curve, labels the axis, adds a title, as well as a dashed line for to represent no discriminatory power)*

plt.show() *(Displays a grid on the plot, making it easier to read values and compare performance at different thresholds)*

The ROC curve illustrates the trade-off between sensitivity (true positive rate) and 1 – specificity (false positive rate) across a range of classification thresholds. It provides a visual summary of the model's performance in ranking positive cases above negative ones, independent of any fixed decision threshold (Mandrekar, 2010).

The AUC was also reported, offering a single scalar value that quantifies overall discriminative ability. An AUC of 0.5 indicates performance no better than chance, while an AUC closer to 1.0 reflects excellent separation between the two outcome groups. In clinical settings, this analysis is especially valuable, as it highlights how well the model distinguishes at-risk patients even

before a fixed cut-off is applied, supporting more flexible or threshold-adjusted decision-making (Hajian-Tilaki, 2013).

```python
from sklearn.metrics import roc_curve
import matplotlib.pyplot as plt

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkblue', lw=2, label='ROC curve (AUC = {:.2f})'.format(auc))
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - Logistic Regression Reduced Model')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

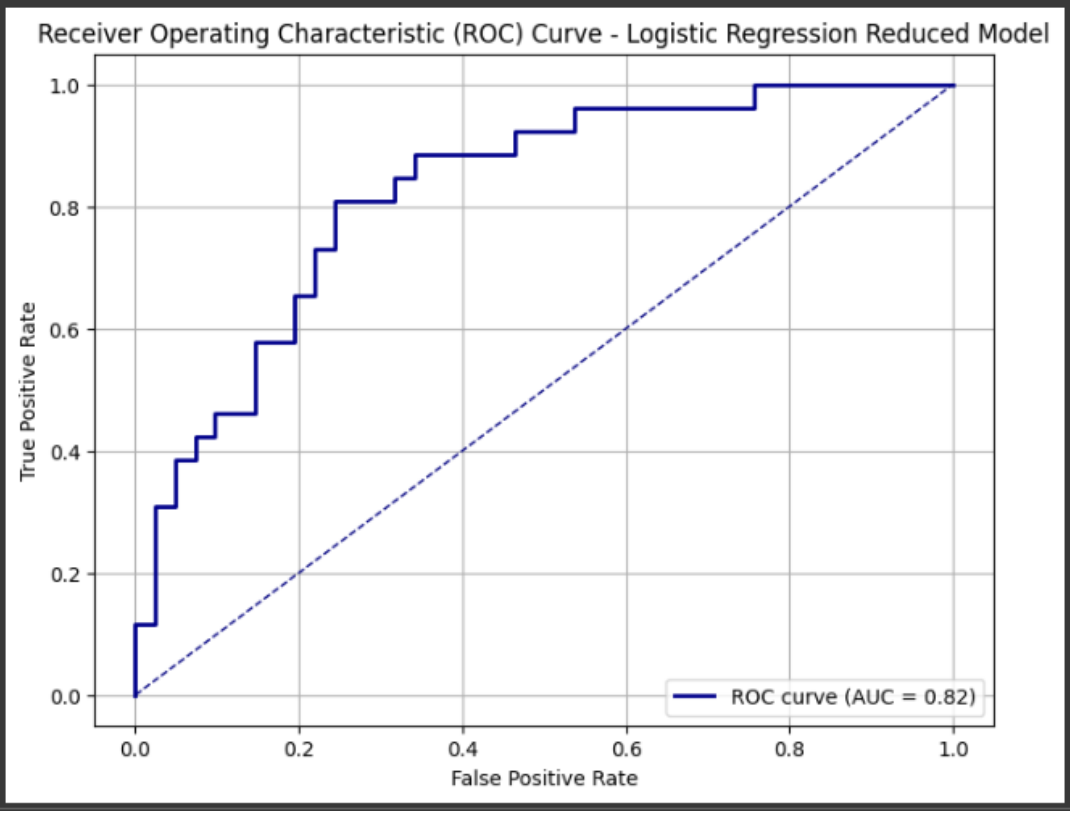


Figure 2.12: Code to display the ROC curve.
Model: Serum Logistic Regression. Source: Author's own Python code.

Note: the ROC curve is shown here as it appears in Google Colab after entering the Python code, future codes will only display the code as the curve is available in the results section.

2.3.12 Cross-Validation

To further assess the model's robustness and generalisability, 5-fold cross-validation was applied to the training dataset. This approach involves dividing the training data into five equal subsets, or "folds." In each round of validation, the model is trained on four of the folds and evaluated on the remaining one. This process is repeated five times, with each fold serving as the validation set exactly once. This was performed using the following Python code:

```
from sklearn.model_selection import cross_val_score
```
(This imports the cross_val_score function from scikit-learn. It allows you to perform cross-validation and automatically return a list of scores for each fold)

```
cross_val_auc_scores = cross_val_score(
    logreg_reduced,
    X_train_imp,
    y_train,
    cv=5,
    scoring='roc_auc')
```

(The above code performs the 5-fold cross-validation on the reduced logistic regression model. logreg_reduced is the model to be evaluated, X_train_imp is the input training features (only the important ones, already scaled), y_train is the true target values, cv=5 confirms the use of 5-fold cross-validation (split data into 5 equal parts), and scoring='roc_auc' uses AUC as the evaluation metric for each fold)

```
print(f"Mean Cross-Validated AUC: {cross_val_auc_scores.mean():.3f}")
```
(This calculates and prints the average AUC across all 5 folds, rounded to three decimal places. It gives a single summary value that reflects the model's typical discriminative performance when trained and tested on different subsets of the training data)

Cross-validation provides a more reliable estimate of model performance by ensuring that every data point in the training set contributes to both model development and evaluation. By repeatedly evaluating the model on different subsets of the data, cross-validation provides an estimate of performance that is less dependent on any single partition, thereby improving robustness. This process also better approximates how the model is likely to perform on unseen data, enhancing its generalisability to independent patient populations (Hastie et al., 2009;

Steyerberg, 2019). Unlike a single train/test split, which may produce overly optimistic or pessimistic results depending on how the data was divided, cross-validation helps reduce variance in performance estimates. This is particularly important in clinical modelling, where consistency across different patient subsets is crucial (Wilimitis and Walsh, 2023; Collins et al., 2024)

If the model achieves a consistently high AUC across all folds, this supports the stability of its predictions and strengthens confidence that the observed performance is not an artefact of a particular data partitioning.

```
from sklearn.model_selection import cross_val_score

cross_val_auc_scores = cross_val_score(
    logreg_reduced,
    X_train_imp,
    y_train,
    cv=5,
    scoring='roc_auc'
)

print(f"Mean Cross-Validated AUC: {cross_val_auc_scores.mean():.3f}")
```

```
Mean Cross-Validated AUC: 0.780
```

Figure 2.13: Code to generate Cross Validation AUC score.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.13 Feature Count Reporting

The final number of predictors retained after feature selection was recorded and reported as part of the modelling process using the following Python code:

print(f"Total features before selection: {len(X.columns)}") *(Displays total features before selection)*

print(f"Features retained after selection: {len(important_features)}") *(Displays total features after selection)*

This step highlights that the model underwent deliberate simplification based on objective statistical criteria, even though only one feature was removed based on the criteria selected. Documenting this reduction in dimensionality not only supports transparency and reproducibility but also signals that the model was refined with interpretability and generalisability in mind.

Feature reduction is particularly important in clinical modelling where datasets may be limited in size and were overly complex models risk overfitting and poor external validity. By retaining only

the most informative variables, the final model becomes easier to interpret and explain, a crucial consideration when applying predictive tools in real-world healthcare settings, where clinical decision-making often depends on understanding how and why a model reaches its conclusions.

```
print(f"Total features before selection: {len(X.columns)}")
print(f"Features retained after selection: {len(important_features)}")
```

```
Total features before selection: 13
Features retained after selection: 12
```

Figure 2.14: Code to generate Feature Selection.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.14 Coefficient Table

To assess the relative influence of each predictor within the model, the logistic regression coefficients were extracted using the following Python code:

for feature, coef in zip(X.columns, logreg.coef_[0]): *(This line starts a loop that goes through each feature name and its corresponding coefficient in the trained logistic regression model. X.columns gives the names of the features used in the model. logreg.coef_[0] accesses the list of learned coefficients, and zip(...) combines the two lists (feature names and coefficients) so they can be processed together)*

print(f"{feature}: {coef}") *(prints each feature name alongside its corresponding coefficient value)*

These coefficients represent the change in the log-odds of the outcome (i.e. escalation to second-line treatment) associated with a one-unit increase in each predictor, holding all other variables constant. By reviewing the size and direction (positive or negative) of the coefficients, it becomes possible to understand how individual features contribute to the model's predictions (Harrell, 2015). As outlined previously in section 2.3.7, an initial feature selection step was carried out by retaining variables with coefficients greater than 0.05 in absolute value. However, as demonstrated in section 2.3.13, this threshold resulted in the removal of only a single feature. Considering this minimal reduction, the full set of coefficients was reviewed to explore whether further manual refinement or simplification could be justified. Presenting this information in a structured coefficient table supports transparency and facilitates deeper interpretation of the model's behaviour.

```
for feature, coef in zip(X.columns, logreg.coef_[0]):
    print(f"{feature}: {coef}")
```

```
Day0_hCG: 0.3030719093056577
Day14_hCG: -0.005680410034555294
Day28_hCG: 0.18515236866351553
Slope_0_14: -0.5906139618369928
Slope_14_28: 0.2299313363750998
Slope_0_28: -0.30320531817651136
Percent_Drop_0_14: -0.057480565655294316
Percent_Drop_14_28: -0.5206784016686719
Percent_Drop_0_28: -0.48348743179509396
Ratio_0_to_14: 0.05040141371830274
Ratio_0_to_28: -0.4635595583933
Rebound_14_28: 0.4961656725774698
Min_hCG_0_28: 0.2017047215951417
```

Figure 2.15: Code to present Coefficient Table.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.15 Odds Ratio Presentation

For clinical relevance and easier interpretability, the extracted logistic regression coefficients were exponentiated to create odds ratios using the following Python code:

import numpy as np *(This imports NumPy whose function has been explained previously)*

odds_ratios = np.exp(logreg.coef_[0]) *(This line converts the logistic regression coefficients into odds ratios by exponentiating them. logreg.coef_[0] retrieves the list of coefficients from the fitted logistic regression model and np.exp() applies the exponential function to each value, turning log-odds into standard odds ratios. This makes the model's outputs more interpretable in a clinical context)*

for feature, odds in zip(X.columns, odds_ratios): *(This code starts a loop that pairs each feature name with its corresponding odds ratio. X.columns gives the names of all predictors used in the model and the odds_ratios is the list of odds ratios for each predictor. zip(...) combines the two lists into pairs so they can be printed together)*

print(f"{feature}: {odds:.3f}") *(This prints the name of each feature along with its odds ratio, rounded to three decimal places)*

While coefficients operate on a log-odds scale that can be abstract, odds ratios offer a more intuitive interpretation. They indicate how many times more (or less) likely the outcome becomes for every unit increase in each variable. An odds ratio above 1 suggests a positive association with escalation risk, whereas a value below 1 indicates a protective or inverse effect. Presenting odds ratios alongside confidence intervals allows for an appreciation of both effect size and uncertainty, making them a valuable tool for translating statistical results into clinical meaning (Harrell, 2015: Norton et al., 2018).

```
Rebound_14_28: 0.4961656725774698
Min_hCG_0_28: 0.2017047215951417

import numpy as np

odds_ratios = np.exp(logreg.coef_[0])

for feature, odds in zip(X.columns, odds_ratios):
    print(f"{feature}: {odds:.3f}")

Day0_hCG: 1.354
Day14_hCG: 0.994
Day28_hCG: 1.203
Slope_0_14: 0.554
Slope_14_28: 1.259
Slope_0_28: 0.738
Percent_Drop_0_14: 0.944
Percent_Drop_14_28: 0.594
Percent_Drop_0_28: 0.617
Ratio_0_to_14: 1.052
Ratio_0_to_28: 0.629
Rebound_14_28: 1.642
Min_hCG_0_28: 1.223
```

Figure 2.16: Code to present Odds Ratios.
Model: Serum Logistic Regression. Source: Author's own Python code.

2.3.16 Parsimonious Model Development

To explore whether the model could be further simplified without substantial loss of performance, all features were ranked by the absolute magnitude of their logistic regression coefficients.

The five predictors with the largest absolute values were:

- Slope_0_14
- Percent_Drop_14_28
- Rebound_14_28

- Percent_Drop_0_28
- Ratio_0_to_28

These were selected for further evaluation. A new logistic regression model was trained using only this reduced subset of features. This process aimed to test whether a more compact model, focused on the strongest contributors, could retain similar discriminative power. Simplified models are often more interpretable, less prone to overfitting, and more likely to be adopted in clinical practice, particularly when working with limited datasets (Chowdhury and Turin, 2020: Steyerberg, 2019). Performance metrics for the reduced model, including accuracy, AUC, precision, recall, F1 score, and negative predictive value, were recalculated using the below Python code for comparison with the full model to assess the impact of this targeted feature reduction:

top_5_features = ['Slope_0_14', 'Percent_Drop_14_28', 'Rebound_14_28', 'Percent_Drop_0_28', 'Ratio_0_to_28'] *(This defines a list of the top five predictors previously identified based on their coefficient magnitudes. These are assumed to have the greatest impact on model performance)*

scaler = StandardScaler() *(Creates a standard scaler object from scikit-learn to perform z-score normalisation. Standardising ensures that each feature contributes equally during model training, which is especially important for logistic regression)*

X_train_reduced = scaler.fit_transform(X_train[top_5_features]) *(Fits the scaler to the selected features from the training set, and then scales them. The mean and standard deviation of each feature are calculated from the training data and used for standardisation)*

X_test_reduced = scaler.transform(X_test[top_5_features]) *(Applies the same transformation to the test set using the parameters derived from the training data. This prevents data leakage and ensures consistency between training and test features)*

logreg_reduced_5 = LogisticRegression(max_iter=1000, random_state=42) *(Creates a new logistic regression model instance, allowing up to 1000 iterations for convergence and setting a fixed random seed (42) for reproducibility – both previously explained in depth)*

logreg_reduced_5.fit(X_train_reduced, y_train) *(Fits (trains) the model using only the scaled top five features from the training data)*

y_pred_reduced = logreg_reduced_5.predict(X_test_reduced) *(y_pred_reduced predicts the class labels (0 or 1) for the test set)*

y_probs_reduced = logreg_reduced_5.predict_proba(X_test_reduced)[:, 1] (y_probs_reduced predicts the probability of escalation (class 1) for each test case. The [:, 1] selects the second column of probabilities (positive class))

from sklearn.metrics import accuracy_score, roc_auc_score, precision_score, recall_score, f1_score, confusion_matrix *(This line imports the performance metrics used to assess the model, including accuracy, AUC, precision, recall, F1 score, and confusion matrix)*

accuracy_reduced = accuracy_score(y_test, y_pred_reduced)

auc_reduced = roc_auc_score(y_test, y_probs_reduced)

precision_reduced = precision_score(y_test, y_pred_reduced)

recall_reduced = recall_score(y_test, y_pred_reduced)

f1_reduced = f1_score(y_test, y_pred_reduced)

(The above 5 lines calculate the metrics for the 5-feature model)

cm_reduced = confusion_matrix(y_test, y_pred_reduced)

tn_reduced, fp_reduced, fn_reduced, tp_reduced = cm_reduced.ravel()

npv_reduced = tn_reduced / (tn_reduced + fn_reduced)

(The above 3 lines compute the confusion matrix, which breaks down the model's predictions into true/false positives and negatives. Then extracts those individual values and calculates NPV)

print(f"5-Feature Model Results:")

print(f"Accuracy: {accuracy_reduced:.3f}")

print(f"AUC: {auc_reduced:.3f}")

print(f"Precision: {precision_reduced:.3f}")

print(f"Recall (Sensitivity): {recall_reduced:.3f}")

print(f"F1 Score: {f1_reduced:.3f}")

print(f"Negative Predictive Value (NPV): {npv_reduced:.3f}")

(The above 7 lines of code print a summary of all the evaluation metrics for the reduced model, rounded to three decimal places for clarity and easy comparison with the full model)

```python
top_5_features = ['Slope_0_14', 'Percent_Drop_14_28', 'Rebound_14_28', 'Percent_Drop_0_

scaler = StandardScaler()
X_train_reduced = scaler.fit_transform(X_train[top_5_features])
X_test_reduced = scaler.transform(X_test[top_5_features])

logreg_reduced_5 = LogisticRegression(max_iter=1000, random_state=42)
logreg_reduced_5.fit(X_train_reduced, y_train)

y_pred_reduced = logreg_reduced_5.predict(X_test_reduced)
y_probs_reduced = logreg_reduced_5.predict_proba(X_test_reduced)[:, 1]

from sklearn.metrics import accuracy_score, roc_auc_score, precision_score, recall_scor

accuracy_reduced = accuracy_score(y_test, y_pred_reduced)
auc_reduced = roc_auc_score(y_test, y_probs_reduced)
precision_reduced = precision_score(y_test, y_pred_reduced)
recall_reduced = recall_score(y_test, y_pred_reduced)
f1_reduced = f1_score(y_test, y_pred_reduced)
cm_reduced = confusion_matrix(y_test, y_pred_reduced)
tn_reduced, fp_reduced, fn_reduced, tp_reduced = cm_reduced.ravel()
npv_reduced = tn_reduced / (tn_reduced + fn_reduced)

print(f"5-Feature Model Results:")
print(f"Accuracy: {accuracy_reduced:.3f}")
print(f"AUC: {auc_reduced:.3f}")
print(f"Precision: {precision_reduced:.3f}")
print(f"Recall (Sensitivity): {recall_reduced:.3f}")
print(f"F1 Score: {f1_reduced:.3f}")
print(f"Negative Predictive Value (NPV): {npv_reduced:.3f}")
```

```
5-Feature Model Results:
Accuracy: 0.731
AUC: 0.829
Precision: 0.682
Recall (Sensitivity): 0.577
F1 Score: 0.625
Negative Predictive Value (NPV): 0.756
```

Figure 2.17: Code to generate the Model Evaluation.
Model: Serum Logistic Regression (Reduced). Source: Author's own Python code.

2.3.17 Inclusion of Baseline hCG (Day 0)

Following coefficient ranking, it was observed that Day0_hCG, representing the initial serum hCG level prior to first treatment, had a moderately large coefficient (+0.3024) but narrowly missed inclusion in the top five features. Despite this, Day0_hCG was manually retained in the reduced model due to its well-established clinical significance as a marker of disease burden in GTD. Initial hCG values are routinely used in prognostic scoring systems and treatment decisions, and excluding such a key variable risked undermining the clinical relevance of the model.

This brought the total number of selected predictors to six:

- Slope_0_14
- Percent_Drop_14_28
- Rebound_14_28
- Percent_Drop_0_28
- Ratio_0_to_28
- Day0_hCG (manually added)

Retaining clinically important variables that may not strictly meet statistical thresholds reflects best practice in predictive modelling. Model development in healthcare requires more than algorithmic optimisation, it must be informed by domain expertise and grounded in clinical context (Steyerberg, 2019; Chowdhury & Turin, 2020). This approach enhances both interpretability and acceptability in real-world settings.

The same Python code as 2.3.17 was used, modifying it to include Day0_hCG:

```python
top_5_plus_day0hcg = top_5_features + ['Day0_hCG']

top_5_plus_day0hcg = list(set(top_5_plus_day0hcg))

X_train_reduced_6 = scaler.fit_transform(X_train[top_5_plus_day0hcg])
X_test_reduced_6 = scaler.transform(X_test[top_5_plus_day0hcg])

logreg_reduced_6 = LogisticRegression(max_iter=1000, random_state=42)
logreg_reduced_6.fit(X_train_reduced_6, y_train)

y_pred_reduced_6 = logreg_reduced_6.predict(X_test_reduced_6)
y_probs_reduced_6 = logreg_reduced_6.predict_proba(X_test_reduced_6)[:, 1]

accuracy_reduced_6 = accuracy_score(y_test, y_pred_reduced_6)
auc_reduced_6 = roc_auc_score(y_test, y_probs_reduced_6)
precision_reduced_6 = precision_score(y_test, y_pred_reduced_6)
recall_reduced_6 = recall_score(y_test, y_pred_reduced_6)
f1_reduced_6 = f1_score(y_test, y_pred_reduced_6)
cm_reduced_6 = confusion_matrix(y_test, y_pred_reduced_6)
tn_reduced_6, fp_reduced_6, fn_reduced_6, tp_reduced_6 = cm_reduced_6.ravel()
npv_reduced_6 = tn_reduced_6 / (tn_reduced_6 + fn_reduced_6)

print(f"6-Feature Model Results (Top 5 + Day0_hCG):")
print(f"Accuracy: {accuracy_reduced_6:.3f}")
print(f"AUC: {auc_reduced_6:.3f}")
print(f"Precision: {precision_reduced_6:.3f}")
print(f"Recall (Sensitivity): {recall_reduced_6:.3f}")
print(f"F1 Score: {f1_reduced_6:.3f}")
print(f"Negative Predictive Value (NPV): {npv_reduced_6:.3f}")
```

```
6-Feature Model Results (Top 5 + Day0_hCG):
Accuracy: 0.746
AUC: 0.837
Precision: 0.714
Recall (Sensitivity): 0.577
F1 Score: 0.638
Negative Predictive Value (NPV): 0.761
```

Figure 2.18: Code to generate the Model Evaluation.
Model: Serum Logistic Regression (Reduced+1). Source: Author's own Python code.

2.3.18 Cross-Validation of Six-Feature Model

As with the earlier evaluation of the original model (see Section 2.3.13), five-fold cross-validation was conducted on the updated six-feature model using the same approach and code structure. This procedure helped assess the model's generalisability by evaluating performance across multiple training–validation splits, providing a more robust estimate of its discriminative ability.

```
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(solver='liblinear')
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

auc_scores = cross_val_score(model, X_scaled, y, cv=cv, scoring='roc_auc')

# Print only the mean AUC
print(f"AUC: {auc_scores.mean():.3f}")
```

```
AUC: 0.726
```

Figure 2.19: Code to generate Cross Validation AUC score.
Model: Serum Logistic Regression (Reduced+1). Source: Author's own Python code.

2.3.19 Clinical Prediction Tool Development

Following the development of a reduced logistic regression model to predict escalation from MTX to ActD in GTD, a practical, clinician-friendly Excel-based prediction tool was developed. The aim was to allow rapid, real-world application of the predictive model using routinely available early hCG measurements without requiring access to specialist software or coding platforms, which is normally found in the Multi-Disciplinary Team (MDT) setting.

2.3.19.1 Tool Design and Implementation

The Excel prediction tool was structured into four main functional sections:

1. Input Section

Users manually input three early treatment hCG measurements:

- Day 0 hCG (baseline)
- Day 14 hCG
- Day 28 hCG

2. Feature Engineering Section

Automated formulas calculate the six predictor variables used in the logistic regression model:

- Slope_0_14 (rate of hCG change between Day 0 and Day 14)
- Percent_Drop_14_28 (percentage decrease between Day 14 and Day 28)
- Rebound_14_28 (proportional rise between Day 14 and Day 28, if present)

- Percent_Drop_0_28 (overall percentage decrease from baseline to Day 28)
- Ratio_0_to_28 (baseline hCG divided by Day 28 hCG)
- Day0_hCG (baseline value, reused)

Formulas automate the derivation of these features immediately after input.

3. Scaling Section

Each feature is standardised (scaled) before being applied to the logistic regression formula.

Logistic regression coefficients were trained on standardised data, where each feature was transformed to have:

- A mean of 0
- A SD of 1

Standardisation ensures that features with vastly different scales (e.g. raw hCG levels vs percentage drops) are treated comparably during model training, preventing features with larger absolute values from disproportionately influencing the model.

For each feature, the following standardisation formula was used:

$$Scaled\ Feature = \frac{Actual\ Value - Training\ Value}{Training\ Standard\ Deviation}$$

The training means and SD were extracted from the model development training set and hard coded into the Excel file to ensure consistency.

4. Prediction Section

The scaled features are then automatically applied to the trained logistic regression equation:

$$Logit(p) = \beta 0 + (\beta 1\ x\ Slope\_0\_14) + (\beta 2\ x\ Percent\_Drop\_14\_28) + \cdots + (\beta 6\ x\ Day0\_hCG)$$

Where:

$\beta 0$ is the intercept

$\beta 1, \beta 2, \ldots \beta 6$ are the trained model coefficients for each feature.

The logistic transformation was then applied:

$$P = \frac{1}{1 + e^{-logit(p)}}$$

where p represents the predicted probability of escalation to ActD.

The final probability output would then be displayed directly in the Excel sheet, in real time.

5. Logistic Regression Model Coefficients Used

Feature	Coefficient (β)
Slope_0_14	-0.5916
Percent_Drop_14_28	-0.5221
Rebound_14_28	+0.4960
Percent_Drop_0_28	-0.4783
Ratio_0_to_28	-0.4646
Day0_hCG	+0.3024
Intercept	+0.0735

Table 2.1: Logistic Regression Model Coefficients Table for use in Excel Prediction Tool. Source: Generated in Python using Author's own code.

6. Example User Instructions, if it was made live.
    a. Enter the Day 0, Day 14, and Day 28 hCG measurements in the highlighted yellow cells.
    b. The Excel sheet automatically:
        i. Calculates required features.
        ii. Standardises each feature based on the model training set.
        iii. Computes the final predicted probability.
    c. The predicted probability of escalation is displayed directly in the output column.
    d. A default decision threshold of 0.5 may be applied, whereby:
        i. Predicted probability ≥ 0.5 suggests high risk of escalation to ActD.
        ii. Predicted probability < 0.5 suggests continued MTX therapy is likely to succeed.

Threshold adjustment could be considered depending on clinical context and desired sensitivity-specificity balance.

Standardisation within the Excel sheet mirrors the exact preprocessing applied during model training, ensuring that predictions remain valid and comparable.

All calculations were transparent, and formulas could be inspected or adjusted if required.

The tool facilitates rapid, real-world application of the logistic regression model without specialist software, making predictive modelling accessible in day-to-day clinical decision making.

2.4 Gradient Boosting Modelling – Serum Dataset

To further investigate predictive modelling approaches beyond logistic regression, a Gradient Boosting Classifier was also developed. Gradient boosting is a machine learning technique that builds an ensemble of weak prediction models (typically decision trees) in a sequential manner, where each subsequent model attempts to correct errors made by previous models. Gradient Boosting is well-suited for tabular clinical data and is capable of capturing complex, non-linear relationships between features and outcomes. The purpose of implementing a Gradient Boosting model was to assess whether a more flexible model could capture non-linear relationships and interactions between variables that may not be adequately represented by logistic regression. Ensemble methods such as gradient boosting are well-suited to modelling complex data structures that may not be represented by traditional regression approaches (Friedman, 2001).

2.4.1 Data Upload in Google Colab

As above in section 2.2

2.4.2 Data Loading and Inspection

As above in section 2.2.1

2.4.3 Data Cleaning and Feature Selection

Prior to model training, the dataset was reviewed and cleaned to ensure that only relevant, informative variables were retained for analysis. Feature selection at this stage focused on the removal of columns that were either non-predictive or carried information unavailable at the point of initial prediction. Two variables were excluded on this basis. The first, Patient_ID, was a purely administrative identifier with no biological or clinical relevance, retaining it would have introduced noise without contributing to model performance. The second, Trajectory_Type, categorised patients based on their known disease outcome, information that would not be available at the time of early prediction. Including such a variable would result in data leakage in this model type, allowing the model to "cheat" by basing predictions on future knowledge, and thereby undermining the validity of performance estimates.

The following Python code was executed to drop the unnecessary columns:

df_model = df.drop(columns=['Patient_ID', 'Trajectory_Type']) *(This line creates a new DataFrame called df_model by removing two non-predictive columns: Patient_ID and Trajectory_Type, for the reasons explained above)*

df_model.head() *(This displays the first five rows of the cleaned dataset (df_model) to confirm that the removal was successful)*

```
df_model = df.drop(columns=['Patient_ID', 'Trajectory_Type'])

df_model.head()
```

	Day0_hCG	Day14_hCG	Day28_hCG	Slope_0_14	Slope_14_28	Slop
0	28927	1134	50	-1985.214286	-77.428571	-1031.
1	178891	34731	4743	-10297.142860	-2142.000000	-6219.
2	60419	882	115	-4252.642857	-54.785714	-2153.
3	75930	5391	411	-5038.500000	-355.714286	-2697.
4	114641	22866	5409	-6555.357143	-1246.928571	-3901.

Figure 2.20: Code to drop irrelevant Columns.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.4 Predictor and Target Definition

In supervised machine learning, it is essential to clearly define and separate the two core components of model development: the predictors and the target variable. The predictors (commonly denoted as X) refer to the input features, clinical measurements, patient characteristics, or derived values that the model uses to generate predictions. The target (denoted as y) is the specific outcome being predicted, in this case, whether the patient ultimately required escalation to second-line treatment. These predictors were defined using the following Python code:

X = df_model.drop(columns=['Escalated']) *(This line creates a new variable, X, which contains the predictor variables, which are all the columns in the dataset except the outcome variable, 'Escalated'. By explicitly removing the target column, this step ensures that only clinically available inputs are used to train the model)*

y = df_model['Escalated'] *(This defines the target variable, y, which the model is intended to predict. In this study, Escalated is a binary outcome indicating whether the patient required second-line treatment)*

X.head(), y.head() *(This line displays the first five rows of both X and y, allowing verification that the data has been separated correctly)*

Separating these components explicitly is critical to prevent data leakage and to ensure valid model training, as incorporating information related to the outcome into the input features can lead to overfitting and inflated performance estimates (Steyerberg, 2019; Subbaswamy & Saria, 2020).

```
X = df_model.drop(columns=['Escalated'])

y = df_model['Escalated']

X.head(), y.head()
```

Figure 2.21: Code separating predictors X and target y.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.5 Train–Test Split

To prepare the data for model development, the dataset was split into two subsets: a training set and a testing set. As described previously, this ensures that model performance can be evaluated on data it has not encountered during training, thereby providing a more realistic estimate of generalisability. An 80/20 split was used, in line with standard practice in machine learning (Hastie et al., 2009; Pedregosa et al., 2011). To allow reproducibility of results, the split was performed using a fixed random seed. The following Python code was used to carry out this process:

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(

X, y, test_size=0.2, random_state=42

) *(The function of the above lines of code has been described previously)*

print(f"Training samples: {X_train.shape[0]}")

print(f"Testing samples: {X_test.shape[0]}")

(The above two lines print the number of samples in each split. This provides a quick check to confirm the expected 80/20 division was applied correctly)

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print(f"Training samples: {X_train.shape[0]}")
print(f"Testing samples: {X_test.shape[0]}")
```

```
Training samples: 265
Testing samples: 67
```

Figure 2.22: Code separating Training and Testing Groups.
Model: Serum Gradient Boosting. Source: Author's own Python code.

Note: As my understanding of Python and machine learning workflows developed throughout the research, I adopted more efficient and transparent coding practices, including clearer verification of key preprocessing steps. Although not initially used during earlier phases of model development (e.g., linear regression implementation), the following code was retrospectively added to confirm that the dataset was correctly partitioned during the train/test split procedure.

print(f"Training samples: {X_train.shape[0]}")

print(f"Testing samples: {X_test.shape[0]}")

These commands simply output the number of samples allocated to the training and testing sets, respectively. By confirming the row counts following the split, they provide reassurance that the expected 80/20 partitioning was correctly applied. This form of validation, although basic, reflects good programming discipline and supports reproducibility, especially important when methods evolve iteratively during research.

2.4.6 Feature Scaling Considerations

In contrast to linear models such as logistic regression or support vector machines, which require input features to be standardised to ensure numerical stability and balanced learning, tree-based algorithms do not require feature scaling (Hastie et al., 2009). This is particularly relevant for GBMs, which build decision trees by selecting threshold-based splits rather than relying on geometric relationships like Euclidean distance. Tree algorithms evaluate features through binary rules such as "is feature *x* greater than a threshold?", rather than considering the magnitude of feature values directly. As a result, a feature ranging from 0–1 and another ranging from 0–10,000 are treated equivalently in terms of split logic (Natekin and Knoll, 2013).

In addition, GBM models are internally robust to differences in feature scale and are capable of identifying informative splits regardless of numerical range. This property not only simplifies preprocessing but also enhances computational efficiency, as it eliminates the need for transformation steps like standardisation or normalisation. Accordingly, for GBM-based models developed in this study, raw features were passed directly into the algorithm without applying any form of scaling (Natekin and Knoll, 2013).

2.4.7 Baseline Gradient Boosting Model

To begin the GMB modelling process, an initial classifier was trained using default hyperparameter settings using the following Python code:

from xgboost import XGBClassifier *(This line imports the XGBClassifier from the XGBoost library. This classifier is a high-performance implementation of Gradient Boosting tailored for supervised learning tasks, particularly classification problems)*

gbm_model = XGBClassifier(

use_label_encoder=False,

eval_metric='logloss',

random_state=42 *(The above code activates new XGBoost classifier with three key parameter settings, use_label_encoder=False which disables the deprecated internal label encoder, which is unnecessary when class labels are already properly formatted. eval_metric='logloss' that specifies the evaluation metric as logarithmic loss, appropriate for binary classification tasks where predicted probabilities are evaluated, and random_state=42, which has been described previously)*

gbm_model.fit(X_train, y_train) *(This line fits (trains) the GBM model on the training dataset)*

XGBoost is widely used in medical and clinical prediction modelling due to its ability to handle structured tabular data, manage missing values, and capture non-linear relationships between features and outcomes (Chen & Guestrin, 2016). This approach served to establish a baseline performance level, providing a reference point against which future tuning and optimisation steps could be compared. Modern implementations such as XGBoost are designed with sensible defaults that often yield competitive results across a wide range of datasets, making them suitable for preliminary evaluation (Chen and Guestrin, 2016). Starting with a simple, untuned

model also allows for early diagnostic insights into model fit, data quality, and feature utility. By first assessing model performance under default conditions, subsequent modifications, including hyperparameter tuning or feature engineering can be introduced in a stepwise, interpretable manner with measurable gains in performance (Bergstra and Bengio, 2012).

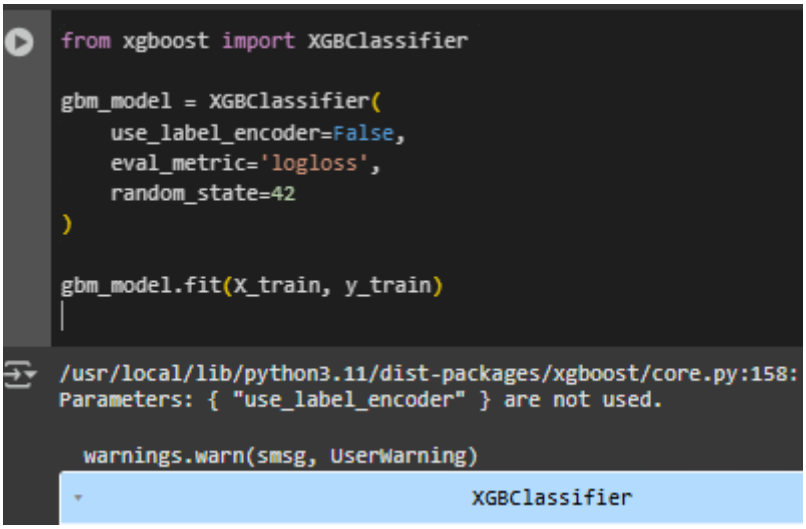


Figure 2.23: Code training Model on default Hyperparameters. Model: Serum Gradient Boosting. Source: Author's own Python code.

Note: This step highlighted an error code, which can be seen above:

WARNING: Parameters: { "use_label_encoder" } are not used.

This is not a crash or critical error, but only a warning. In earlier versions of XGBoost (2020 – 2021), the parameter use_label_encoder was introduced to manage how categorical labels (e.g., 0 and 1) were internally encoded. However, in more recent versions (2023 – 2025), including those used by platforms like Google Colab, this parameter has been deprecated and is no longer necessary. As a result, specifying use_label_encoder=False is now ignored by the library, which triggers a warning but has no impact on model function or output. The model continues to train as expected, and the results remain valid.

Although the warning can be suppressed by updating to a code such as that in Figure 2.24, adjustments have not been implemented here as the warning is descriptive and does not affect the validity or reproducibility of the analysis.

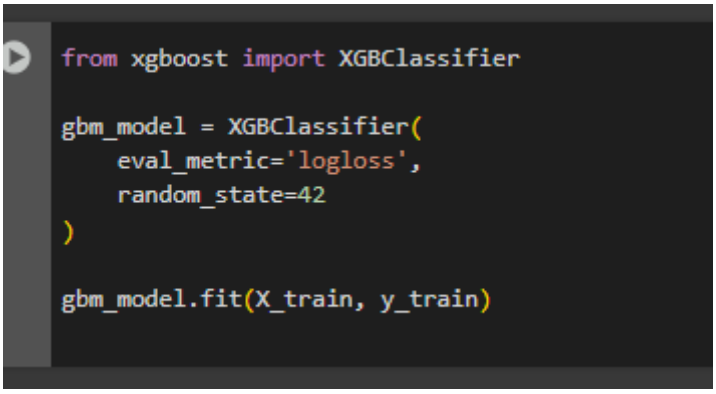


Figure 2.24: Code to prevent label_encoder=False error.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.8 Evaluation Metrics

Once the GBM model was trained on the training dataset, the next step was to evaluate its predictive performance using the unseen testing data. This evaluation serves two key purposes: it verifies the model's ability to generalise beyond the data it was trained on, and it provides objective, quantitative metrics that can be used to assess its accuracy and compare it with alternative models (Steyerberg, 2019). The evaluation was carried out using a set of standard classification metrics, as implemented in the Python code below:

from sklearn.metrics import (

accuracy_score, precision_score, recall_score,

f1_score, roc_auc_score, confusion_matrix *(The above lines import the defined set of performance metrics from scikit-learn, a common Python library for machine learning. These metrics can then evaluate the classifier's performance on the unseen test data)*

y_pred = gbm_model.predict(X_test) *(Generates binary predictions (0 or 1) for the test set using the trained GBM model. This determines whether each patient is classified as escalated or not)*

y_probs = gbm_model.predict_proba(X_test)[:, 1] *(Calculates the predicted probability of class 1 (i.e., escalation) for each test observation. These probabilities are later used to calculate AUC)*

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

auc = roc_auc_score(y_test, y_probs)

(The above lines of code calculate the accuracy, precision (PPV), recall (sensitivity), F1 score, and the AUC)

cm = confusion_matrix(y_test, y_pred) *(Generates the confusion matrix and unpacks its values)*

tn, fp, fn, tp = cm.ravel() *(Provides counts for TN, FP, FN, and TP)*

npv = tn / (tn + fn) *(Calculates NPV)*

print(f"Accuracy: {accuracy:.3f}")

print(f"Precision (PPV): {precision:.3f}")

print(f"Recall (Sensitivity): {recall:.3f}")

print(f"F1 Score: {f1:.3f}")

print(f"AUC: {auc:.3f}")

print(f"NPV: {npv:.3f}")

print("\nConfusion Matrix:")

print(cm) *(The above lines of code output each of the calculated performance metrics, rounded to three decimal places*

```python
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix
)

y_pred = gbm_model.predict(X_test)

y_probs = gbm_model.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_probs)

cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

npv = tn / (tn + fn)

print(f"Accuracy: {accuracy:.3f}")
print(f"Precision (PPV): {precision:.3f}")
print(f"Recall (Sensitivity): {recall:.3f}")
print(f"F1 Score: {f1:.3f}")
print(f"AUC: {auc:.3f}")
print(f"NPV: {npv:.3f}")

print("\nConfusion Matrix:")
print(cm)
```

```
Accuracy: 0.657
Precision (PPV): 0.543
Recall (Sensitivity): 0.731
F1 Score: 0.623
AUC: 0.721
NPV: 0.781

Confusion Matrix:
[[25 16]
 [ 7 19]]
```

Figure 2.25: Code to generate the Model Evaluation.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.9 ROC Curve

While evaluation metrics such as accuracy, precision, recall, and AUC provide quantitative summaries of model performance, they do not convey how performance varies across different classification thresholds. The ROC curve addresses this by offering a visual representation of the model's ability to discriminate between classes across all possible probability cut-offs. It plots the true positive rate (sensitivity) against the false positive rate, allowing for a more visual assessment of model behaviour. The following Python code was used to generate the ROC curve for the GBM model:

```
import matplotlib.pyplot as plt

from sklearn.metrics import roc_curve
```

(The two lines of code above import the necessary libraries to plot the ROC curve. matplotlib.pyplot is used to generate the plot, and roc_curve calculates the FP rate, TP rate), and decision thresholds needed to construct the curve)

```
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
```
(Identifies the key components of the ROC curve. It compares the true class labels (y_test) to the predicted probabilities (y_probs) and returns three outputs, fpr – the false positive rate at various thresholds, tpr – the corresponding true positive rate (sensitivity), and thresholds – the cut-off values used to convert predicted probabilities into binary decisions)

```
plt.figure(figsize=(8,6))
```
(Creates a new figure for the plot with a defined size)

```
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'GBM ROC curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve - GBM')

plt.legend(loc="lower right")

plt.grid(True)

plt.show()
```

(The above series of code generates the figure, as previously discussed)

```python
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'GBM ROC curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - GBM')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Figure 2.26: Code to plot ROC Curve.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.10 Feature Importance Visualisation

Tree-based models such as GBM inherently evaluate and rank the importance of input features during training (Friedman, 2001; Chen and Guestrin, 2016). Feature importance reflects how frequently a variable was used to split nodes across the ensemble of trees, and how much each split contributed to reducing the overall model error. Variables that appear more often and lead to larger reductions in loss are considered more influential in shaping the model's predictions (Friedman, 2001; Chen and Guestrin, 2016). In the context of this study, features with high importance are those most strongly associated with the model's ability to predict escalation. The following Python code was used to visualise the feature importance scores derived from the trained GBM model:

import matplotlib.pyplot as plt *(Imports the plotting library matplotlib.pyplot, which is used to generate the horizontal bar chart showing the relative importance of each feature in the trained GBM model)*

importances = gbm_model.feature_importances *(Retrieves the feature importance scores from the fitted GBM model)*

feature_names = X_train.columns *(Extracts the names of the predictor variables from the training dataset)*

sorted_idx = importances.argsort() *(Sorts the indices of the features based on their importance values, from least to most important)*

plt.figure(figsize=(8,6))

plt.barh(feature_names[sorted_idx], importances[sorted_idx])

plt.xlabel('Feature Importance Score')

plt.title('Feature Importance - GBM Model')

plt.grid(True)

plt.show()

(The above lines of code build chart, add the labels to the axis and chart, and then plots it in Google Colab)

```
import matplotlib.pyplot as plt

importances = gbm_model.feature_importances_

feature_names = X_train.columns

sorted_idx = importances.argsort()

plt.figure(figsize=(8,6))
plt.barh(feature_names[sorted_idx], importances[sorted_idx])
plt.xlabel('Feature Importance Score')
plt.title('Feature Importance - GBM Model')
plt.grid(True)
plt.show()
```

Figure 2.27: Code to Visualise Feature importance.
Model: Serum Gradient Boosting. Source: Author's own Python code.

2.4.11 Hyperparameter Optimisation

GBMs contain a number of adjustable parameters, known as hyperparameters, which govern how the model learns from the data. Careful tuning of these hyperparameters can enhance model performance by improving discrimination (e.g. AUC), calibration, and generalisability, while also reducing the risk of overfitting. Tuning in particular can help the model better capture the underlying structure of the data while avoiding excessive sensitivity to noise or outliers (XGB, 2022).

The most influential hyperparameters in GBM models typically include (Jain, 2016):

- n_estimators: the number of boosting rounds, where each round corresponds to a new decision tree added to the ensemble.
- learning_rate: a shrinkage factor controlling the contribution of each tree; smaller values slow the learning process but reduce the risk of overfitting.

- max_depth: the maximum depth allowed for individual trees; deeper trees allow the model to learn more complex patterns but may introduce overfitting.
- subsample: the proportion of training data randomly sampled for each tree, which introduces stochasticity and helps prevent overfitting.
- colsample_bytree: the fraction of features randomly selected for consideration at each tree, providing further regularisation.

To identify an optimal combination of these hyperparameters, a Grid Search with Cross-Validation (GridSearchCV) was performed. This approach systematically explores multiple combinations of predefined parameter values and internally applies k-fold cross-validation to each, enabling more reliable performance assessment than a single train-test split (Jain, 2016). This process was structured in five main steps:

1. Defining a grid of hyperparameter values to explore
2. Initialising a GridSearchCV object to search through the grid
3. Training and evaluating multiple GBM models using cross-validation
4. Identifying the parameter combination that yielded the best performance
5. Re-training the final GBM model using the selected optimal hyperparameters

2.4.11.1 Parameter Grid Definition

The initial step in hyperparameter tuning involves defining a parameter grid, which is a structured set of hyperparameter values to be evaluated. This grid encompasses a range of plausible values for each critical hyperparameter, allowing for a comprehensive search of the parameter space. The following Python code was used set up the parameter grid:

param_grid = { *(param_grid specifies a range of values to be tested for each of the key hyperparameters of the GBM model)*

'n_estimators': [100, 200, 300], *(Specifies the number of boosting rounds (i.e., the number of trees in the model). The grid explores models with 100, 200, and 300 trees, allowing assessment of both underfitting (too few trees) and overfitting (too many))*

'learning_rate': [0.01, 0.05, 0.1], *(Controls the contribution of each individual tree to the final model. Lower values like 0.01 and 0.05 promote slower, more cautious learning, often yielding better generalisation when combined with more trees)*

'max_depth': [3, 4, 5], *(Sets the maximum depth each decision tree can grow to. Shallower trees (e.g., depth 3) are more constrained and generalisable; deeper trees (e.g., depth 5) may capture more complexity at the risk of overfitting)*

'subsample': [0.8, 1], *(Determines the fraction of the training data to sample when fitting each tree. A value of 1 uses the full dataset, while 0.8 introduces randomness to reduce variance and overfitting)*

'colsample_bytree': [0.8, 1] *(Specifies the fraction of features to randomly sample at each tree construction step. Subsampling features can enhance model robustness, particularly when many correlated predictors are present)*

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1],
    'max_depth': [3, 4, 5],
    'subsample': [0.8, 1],
    'colsample_bytree': [0.8, 1]
}
```

Figure 2.28: Code to set a Parameter Grid.
Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.

2.4.11.2 Grid-Search Execution

Once a parameter grid has been defined, the next step in hyperparameter tuning is to apply a structured search strategy. This was done using GridSearchCV, a function from the scikit-learn library that automates the process of finding the optimal combination of hyperparameters for a machine learning model (Jain, 2025). GridSearchCV performs an exhaustive search over the parameter grid. It evaluates every possible combination of the hyperparameters specified, and for each combination, it fits and assesses a model using 5-fold cross-validation. In this process, the training dataset is divided into 5 equal parts (folds). The model is trained on 4 folds and validated on the remaining one, and this procedure is repeated 5 times so that each fold serves as the validation set once. The results from all folds are then averaged to produce a reliable estimate of model performance for that specific hyperparameter combination. This methodology reduces the bias associated with a single train–test split and offers a more stable evaluation, particularly important in clinical modelling where model generalisation is essential (Pedregosa et al. 2011: Bergstra and Bengio, 2012).

Once all parameter combinations have been tested, GridSearchCV selects the one that yielded the best performance according to a specified scoring metric, in this case, ROC AUC, which is widely regarded as an appropriate metric for binary classification tasks involving imbalanced classes or diagnostic discrimination (Saito & Rehmsmeier, 2015). The final model is then retrained on the full training set using these optimal settings before being evaluated on the independent test set. GridSearchCV offers a balance between model optimisation, reproducibility, and interpretability, and remains a gold-standard method for hyperparameter tuning in supervised machine learning workflows (Pedregosa et al., 2011; Bergstra & Bengio, 2012). By combining variable values from section 2.4.11.1, GridSearchCV will evaluate a total of 3 × 3 × 3 × 2 × 2 = 108 unique combinations. This exhaustive search provides a systematic approach to identify the set of hyperparameters that maximises model performance AUC.

The Python code to initiate the GridSearchCV is shown below:

from sklearn.model_selection import GridSearchCV *(Imports the GridSearchCV class from the sklearn.model_selection module. This automates the search over specified hyperparameter combinations using cross-validation, allowing for reliable model selection)*

grid_search = GridSearchCV(*(Sets up the grid search object)*

estimator=XGBClassifier(eval_metric='logloss', random_state=42), *(estimator=XGBClassifier(...) defines the base model to be tuned. The evaluation metric within the model is set to 'logloss', appropriate for binary classification problems. A fixed random_state ensures reproducibility, as discussed previously)*

param_grid=param_grid, *(Refers to the hyperparameter grid previously defined in section 2.4.11.1, which contains all combinations of hyperparameter values to be evaluated)*

scoring='roc_auc', *(Specifies the performance metric used to evaluate each combination during cross-validation. AUC was chosen here due to its robustness in assessing classifier performance across thresholds)*

cv=5, *(Sets the number of folds for cross-validation)*

verbose=1, *(Enables real-time output of progress during the search, helpful for monitoring in longer tuning procedures)*

n_jobs=-1 *(Enables parallel processing across all available CPU cores, significantly speeding up computation by training multiple models simultaneously)*

grid_search.fit(X_train, y_train) *(Runs the full grid search process using the training data. For each parameter combination, the model is trained and validated using 5-fold cross-validation. Performance scores are computed and stored internally, and the best-performing parameter set is selected)*

print("Best Hyperparameters:", grid_search.best_params_) *(After completing the search, this line outputs the combination of hyperparameters that achieved the highest average AUC across all cross-validation folds. These will be used to re-train the final optimised model)*

```
# Import GridSearchCV
from sklearn.model_selection import GridSearchCV

# Set up the grid search
grid_search = GridSearchCV(
    estimator=XGBClassifier(eval_metric='logloss', random_state=42),
    param_grid=param_grid,
    scoring='roc_auc',
    cv=5,
    verbose=1,
    n_jobs=-1
)

# Fit the grid search
grid_search.fit(X_train, y_train)

# Best hyperparameters
print("Best Hyperparameters:", grid_search.best_params_)
```

```
Fitting 5 folds for each of 108 candidates, totalling 540 fits
Best Hyperparameters: {'colsample_bytree': 1, 'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200, 'subsample': 1}
```

Figure 2.29: Code to GridSearchCV and generate the best Hyperparameters.
Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.

2.4.11.3 Tuned Gradient Boosting Model

With the optimal hyperparameters identified through the GridSearchCV process, a final GBM model was constructed using these best-performing settings. This tuned model represents the most reliable version of the classifier developed in this study and serves three key purposes. First, it is the final model used for all subsequent predictions on the test data and therefore reflects the model's true generalisation capability. Second, it forms the basis for all performance evaluation, metrics such as accuracy, AUC, precision, recall, and NPV are calculated from this model alone. Finally, this tuned model is the one that is reported in the Results chapter and used in all interpretative discussions. It represents the endpoint of the modelling pipeline and is the version most relevant for potential clinical application.

The following Python code was executed to build and fit the hyperparameter optimised model:

final_gbm = XGBClassifier(*(final_gbm is built as the new instance of the tuned XGBoost classifier)*

colsample_bytree=1, *(Uses all available features when constructing each tree)*

learning_rate=0.01, *(Applies a low learning rate, allowing the model to build trees cautiously and reduce overfitting)*

max_depth=3, *(Limits each tree to a depth of 3, helping control model complexity and encourage generalisation)*

n_estimators=200, *(Builds 200 boosting rounds (i.e., decision trees), which balances predictive strength with training time and complexity)*

subsample=1, *(Uses the full dataset for each tree, meaning no row-level subsampling is applied)*

eval_metric='logloss', *(Specifies the loss function used during training, appropriate for binary classification with probabilistic output)*

random_state=42 *(Discussed previously)*

final_gbm.fit(X_train, y_train) *(This code fits (trains) the final model using the full training dataset. The GBM will now learn using the tuned configuration, and this model will be used for all final predictions and evaluations.)*

```
final_gbm = XGBClassifier(
    colsample_bytree=1,
    learning_rate=0.01,
    max_depth=3,
    n_estimators=200,
    subsample=1,
    eval_metric='logloss',
    random_state=42
)

final_gbm.fit(X_train, y_train)
```

Figure 2.30: Fitting the Hyperparameters to the Model.
Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.

2.4.11.4 Tuned Gradient Boosting Model: Evaluation Metrics

After training the final GBM model with the optimal hyperparameters identified via GridSearchCV, its performance was evaluated on the unseen 20% testing set. This evaluation employed multiple metrics to provide a comprehensive assessment of the model's predictive capabilities:

- Accuracy: Measures the overall proportion of correct predictions (both true positives and true negatives) out of all predictions made. While commonly used, accuracy can be misleading in imbalanced datasets, as it may not reflect the model's performance on the minority class.
- Precision (Positive Predictive Value): Indicates the proportion of positive identifications that were actually correct. High precision is crucial when the cost of false positives is high.
- Recall (Sensitivity): Reflects the proportion of actual positives that were correctly identified by the model. High recall is important when the cost of false negatives is high, such as in medical diagnoses.
- F1 Score: The harmonic mean of precision and recall, providing a balance between the two. It is particularly useful when seeking a balance between precision and recall.
- Area Under the ROC Curve (AUC): Evaluates the model's ability to distinguish between classes across all thresholds. A higher AUC indicates better model performance.
- Confusion Matrix: Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives, offering insight into specific types of classification errors.
- Negative Predictive Value (NPV): Denotes the proportion of negative identifications that were actually correct. High NPV is essential when the cost of false negatives is significant.

These metrics collectively offer a nuanced understanding of the model's performance, particularly in the context of clinical data where the implications of false positives and false negatives can be substantial. They were generated using the following Python code:

```
from sklearn.metrics import (
  accuracy_score, precision_score, recall_score,
  f1_score, roc_auc_score, confusion_matrix, roc_curve
import matplotlib.pyplot as plt
```

(These lines of code import the necessary evaluation functions from scikit-learn and the plotting library matplotlib. These tools allow computation of standard classification metrics and visualisation of the ROC curve)

y_pred_final = final_gbm.predict(X_test)

(Generates final binary predictions (0 or 1) for the test set using the optimised GBM model)

y_probs_final = final_gbm.predict_proba(X_test)[:, 1]

(Extracts the predicted probabilities for the positive class (escalation) from the test set. These are used to compute AUC and plot the ROC curve)

accuracy_final = accuracy_score(y_test, y_pred_final)

precision_final = precision_score(y_test, y_pred_final)

recall_final = recall_score(y_test, y_pred_final)

f1_final = f1_score(y_test, y_pred_final)

auc_final = roc_auc_score(y_test, y_probs_final)

(The above lines of code calculate the series of core classification metrics)

cm_final = confusion_matrix(y_test, y_pred_final)

tn, fp, fn, tp = cm_final.ravel()

npv_final = tn / (tn + fn)

(The above code constructs the confusion matrix and extracts the counts of TN, FP, FN, and TP)

print(f"Final Model Accuracy: {accuracy_final:.3f}")

print(f"Final Model Precision (PPV): {precision_final:.3f}")

print(f"Final Model Recall (Sensitivity): {recall_final:.3f}")

print(f"Final Model F1 Score: {f1_final:.3f}")

print(f"Final Model AUC: {auc_final:.3f}")

print(f"Final Model NPV: {npv_final:.3f}")

print("\nConfusion Matrix:")

print(cm_final)

(Prints the calculated performance metrics in a readable format)

fpr_final, tpr_final, thresholds_final = roc_curve(y_test, y_probs_final)

(Computes the FP Rate, TP Rate, and thresholds for generating the ROC curve from the predicted probabilities and true labels)

plt.figure(figsize=(8,6))

plt.plot(fpr_final, tpr_final, color='darkorange', lw=2, label=f'Final GBM ROC curve (AUC = {auc_final:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Final Receiver Operating Characteristic (ROC) Curve - Optimised GBM')

plt.legend(loc="lower right")

plt.grid(True)

plt.show()

(The above lines of code create the ROC curve, as discussed previously)

```python
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix, roc_curve
)
import matplotlib.pyplot as plt

y_pred_final = final_gbm.predict(X_test)

y_probs_final = final_gbm.predict_proba(X_test)[:, 1]

accuracy_final = accuracy_score(y_test, y_pred_final)
precision_final = precision_score(y_test, y_pred_final)
recall_final = recall_score(y_test, y_pred_final)
f1_final = f1_score(y_test, y_pred_final)
auc_final = roc_auc_score(y_test, y_probs_final)

cm_final = confusion_matrix(y_test, y_pred_final)
tn, fp, fn, tp = cm_final.ravel()
npv_final = tn / (tn + fn)

print(f"Final Model Accuracy: {accuracy_final:.3f}")
print(f"Final Model Precision (PPV): {precision_final:.3f}")
print(f"Final Model Recall (Sensitivity): {recall_final:.3f}")
print(f"Final Model F1 Score: {f1_final:.3f}")
print(f"Final Model AUC: {auc_final:.3f}")
print(f"Final Model NPV: {npv_final:.3f}")

print("\nConfusion Matrix:")
print(cm_final)

fpr_final, tpr_final, thresholds_final = roc_curve(y_test, y_probs_final)

plt.figure(figsize=(8,6))
plt.plot(fpr_final, tpr_final, color='darkorange', lw=2, label=f'Final GBM ROC curve (AUC = {auc_final
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Final Receiver Operating Characteristic (ROC) Curve - Optimised GBM')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Figure 2.31: Code to generate the Model Evaluation and produce a ROC curve.
Model: Serum Hyperparameter Tuned Gradient Boosting. Source: Author's own Python code.

2.5 Random Forest Modelling – Serum Dataset

To complement the previous modelling approaches, a Random Forest (RF) model was also developed to predict escalation from MTX to Act D in patients with GTD. RF is an ensemble machine learning method that constructs a multitude of independent decision trees during model training, with each tree contributing one vote toward the final predicted class (Breiman, 2001). This approach is based on bootstrap aggregating, or bagging, whereby multiple subsets of the data are sampled with replacement to build diverse trees. Results are aggregated to improve overall performance and reduce variance, making the model robust to overfitting (Dietterich, 2000). The Random Forest algorithm was chosen for its demonstrated robustness in handling noisy and heterogeneous clinical datasets, such as those containing variations in serial hCG measurements (Biau, 2012). RF models also provide intrinsic measures of feature importance, which are useful in understanding predictors of early escalation in GTD (Cutler et al., 2007). These characteristics made RF particularly suitable for the structured, moderate-sized dataset in this study. Hyperparameter tuning was conducted using GridSearchCV, which was discussed at length in section 2.3.11.2. Model selection was based on maximising the area under the receiver operating characteristic curve across five-fold cross-validation. The final model was retrained on the complete training dataset using the optimal hyperparameters and evaluated using standard classification metrics: accuracy, precision (PPV), recall (sensitivity), F1 score, negative predictive value (NPV), and AUC. A confusion matrix and ROC curve were also generated to assess model performance on the test set.

2.5.1 Data Upload in Google Colab

As above in section 2.2

2.5.2 Data Loading and Inspection

As above in section 2.2.1

2.5.3 Predictor and Target Definition

As in previous models, the dataset was pre-processed to optimise it for predictive modelling and to prevent information leakage. Again, two non-predictive columns were removed: Patient_ID, a unique identifier without predictive value, and Trajectory_Type, a post-hoc label that would introduce target leakage if included as a predictor. Target leakage occurs when information not available at the time of prediction is inadvertently included in the model, leading to overly optimistic performance estimates (Kaufman et al., 2012). Excluding these variables ensured the integrity and generalisability of the modelling process by restricting input features to those available at the point of clinical decision-making, in line with best practice in clinical prediction model development (Steyerberg, 2019). The remaining clinical and biochemical variables, including early hCG measurements, derived slopes, and percentage declines, were retained as predictors and collectively assigned to the feature matrix X. All features were derived within the first 28 days of treatment to reflect early, actionable data. The binary outcome variable, y, encoded whether a patient required escalation from MTX to ActD, thus serving as the supervised learning target. This separation of predictors (X) and response (y) was essential for maintaining a valid machine learning framework, focusing model training on clinically available and temporally appropriate features, in line with standard supervised learning protocols (Hastie et al., 2009). Later stages of the modelling process were implemented using more concise and technically efficient Python syntax, hence the differing code completing this process compared to previous models. This reflected a progression in methodological fluency, allowing for the same core processes, model fitting, evaluation, and validation, to be executed in a more streamlined and integrated manner:

df_model = df.drop(columns=['Patient_ID', 'Trajectory_Type']) *(Creates a new DataFrame called df_model, based on the original DataFrame, df, but it removes two columns: 'Patient_ID' and 'Trajectory_Type')*

X = df_model.drop(columns=['Escalated']) *(A new variable called X is created, which holds the predictor variables (also previously referred to "features"). It takes the cleaned df_model DataFrame and removes the Escalated column, which is the outcome label. This ensures X contains only inputs the model should learn from, not the answer)*

y = df_model['Escalated'] *(Another variable called y is created, which contains the target variable, Escalated, which is a binary variable where 0 means the patient was successfully treated with methotrexate alone, and 1 means the patient required escalation to ActD)*

```
df_model = df.drop(columns=['Patient_ID', 'Trajectory_Type'])
X = df_model.drop(columns=['Escalated'])
y = df_model['Escalated']
```

Figure 2.32: Code to drop irrelevant Columns, and separating predictors X and target y. Model: Serum Random Forest. Source: Author's own Python code.

2.5.4 Train–Test Split

As with earlier modelling procedures, the dataset was divided into training and testing subsets using an 80/20 split. This approach is consistent with standard practice in predictive modelling and was applied here in the same manner as previously described in section 2.3.4. As the process has already been outlined in detail, it is not repeated here. However, for completeness, the Python code used for the split is shown below:

from sklearn.model_selection import train_test_split *(This line imports the train_test_split function from scikit-learn, which is used to divide a dataset into training and testing subsets)*

X_train, X_test, y_train, y_test = train_test_split(

X, y,

test_size=0.2,

random_state=42 *(This splits the feature matrix X and the target variable y into training and test subsets. Specifically, test_size=0.2 allocates 20% of the data to the test set, and 80% to the training set. The random_state=42 ensures that the split is reproducible)*

(X_train, y_train: data used to train the model. X_test, y_test: data reserved for final model evaluation)

print("Training set shape:", X_train.shape)

print("Testing set shape:", X_test.shape)

(These lines display the full shape (rows and columns) of the feature matrices for the training and test sets. This includes both the number of observations (samples) and the number of predictors (features))

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42
)

print("Training set shape:", X_train.shape)
print("Testing set shape:", X_test.shape)
```

```
Training set shape: (265, 13)
Testing set shape: (67, 13)
```

Figure 2.33: Code separating Training and Testing Groups and displaying results for confirmation. Model: Serum Random Forest. Source: Author's own Python code.

Important clarification on output style: In the current model, the following code was used to display the shape of the training and testing sets:

print("Training set shape:", X_train.shape)

print("Testing set shape:", X_test.shape)

In earlier models, the following code was used instead:

print(f"Training samples: {X_train.shape[0]}")

print(f"Testing samples: {X_test.shape[0]}")

The earlier version reports only the number of samples (i.e., rows) in each set, whereas the .shape method used in the current model outputs both the number of samples and the number of features (columns). This provides a more complete overview of the dataset and removes the need for an additional coding step to check feature dimensionality. Importantly, both methods are purely diagnostic and have no effect on the model's training, structure, or performance and are therefore interchangeable.

2.5.5 Feature Scaling Considerations

As discussed previously in section 2.4.6

2.5.6 Baseline Random Forest Model

To begin the RF modelling process, a baseline classifier was developed using the default parameter configuration provided by the scikit-learn implementation of the RandomForestClassifier. This preliminary model served as a reference point for evaluating the impact of subsequent hyperparameter optimisation. The code used to instantiate and train the model in Python is presented below:

from sklearn.ensemble import RandomForestClassifier *(This line imports the Random Forest classifier class from the ensemble module of the scikit-learn library)*

rf = RandomForestClassifier(random_state=42) *(RandomForestClassifier is created using its default settings. No explicit parameters are modified apart from random_state, which is set to 42 to ensure reproducibility of result)*

rf.fit(X_train, y_train) *(The .fit() method is used to train the model on the training dataset. X_train contains the predictor variables (features), while y_train contains the associated class labels. During this process, the algorithm constructs an ensemble of decision trees, each trained on a bootstrap sample of the input data. At each split in a tree, a random subset of features is evaluated to determine the optimal division, enhancing model diversity and mitigating correlation between trees)*

RF is a widely adopted ensemble learning algorithm in biomedical and clinical informatics, particularly valued for its robustness to overfitting, resilience to noise, and ability to model complex, non-linear relationships without extensive preprocessing (Breiman, 2001; Couronné, Probst & Boulesteix, 2018). It constructs multiple decision trees on randomly sampled subsets of the data and aggregates their predictions, reducing variance and improving generalisability. The use of default hyperparameters in early model development serves as a pragmatic first step in supervised learning workflows, enabling a baseline assessment of performance while preserving interpretability and computational efficiency (Fernández-Delgado et al., 2014). In the context of tabular healthcare data, such default configurations often deliver strong initial results, owing to RF's inherent ability to handle feature interactions and data heterogeneity (Couronné et al., 2018). Beginning with an untuned model provides essential diagnostic value, allowing the identification of informative predictors and highlighting data limitations before advancing to more sophisticated optimisation procedures.

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)
```

Figure 2.34: Code training Model on default Hyperparameters.
Model: Serum Random Forest. Source: Author's own Python code

2.5.7 Evaluation Metrics

Following the training of the RF classifier on the training dataset, the next stage involved evaluating its performance on 20% testing set. As highlighted previously, this external validation step is essential for assessing the model's generalisation capacity and ensuring that its predictive success is not a consequence of overfitting to the training data (Hastie, Tibshirani & Friedman, 2009). Performance evaluation on previously unseen data allows for a robust, unbiased estimation of how well the model may perform in real-world clinical settings. Similar metrics as previous models were employed to quantify predictive accuracy, discrimination, and error rates, providing a comprehensive profile of the model's behaviour (Saito & Rehmsmeier, 2015) using the following Python code:

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import (

accuracy_score, precision_score, recall_score,

f1_score, roc_auc_score, confusion_matrix

import numpy as np

(The above lines of code load the necessary functions and classes from the scikit-learn library. The RandomForestClassifier is used to construct and apply the model, while the various metrics such as accuracy_score, precision_score, and roc_auc_score are used to quantify the predictive performance. The numpy library is imported for basic numerical operations)

rf_model = RandomForestClassifier(random_state=42)

rf_model.fit(X_train, y_train)

(New instance of the RF model is instantiated with a fixed random_state for reproducibility. The model is then fitted to the training data using the .fit() method)

y_pred = rf_model.predict(X_test)

y_probs = rf_model.predict_proba(X_test)[:, 1]

(Once trained, the model is used to generate predictions for the test dataset. y_pred stores the final class predictions (0 or 1), while y_probs contains the predicted probabilities for the positive class (1). These probabilities are particularly important for threshold-independent metrics such as the AUC ROC curve)

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

auc = roc_auc_score(y_test, y_probs)

(These lines calculate the key performance metrics, accuracy, precision, recall, F1 score, and AUC – as discussed previously)

cm = confusion_matrix(y_test, y_pred)

tn, fp, fn, tp = cm.ravel()

(The confusion matrix is computed to provide a tabular view of the model's prediction outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The .ravel() function unpacks this matrix into its constituent values for further calculations)

npv = tn / (tn + fn) if (tn + fn) > 0 else 0

(This line of code is required to calculate the NPV manually, as it is not provided by scikit-learn by default)

print(f"Accuracy: {accuracy:.3f}")

print(f"Precision (PPV): {precision:.3f}")

print(f"Recall (Sensitivity): {recall:.3f}")

print(f"F1 Score: {f1:.3f}")

print(f"AUC: {auc:.3f}")

print(f"NPV: {npv:.3f}")

print("\nConfusion Matrix:")

print(cm)

(These above lines print statements output each metric to three decimal places)

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix
)
import numpy as np

rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

y_pred = rf_model.predict(X_test)
y_probs = rf_model.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_probs)
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
npv = tn / (tn + fn) if (tn + fn) > 0 else 0

print(f"Accuracy: {accuracy:.3f}")
print(f"Precision (PPV): {precision:.3f}")
print(f"Recall (Sensitivity): {recall:.3f}")
print(f"F1 Score: {f1:.3f}")
print(f"AUC: {auc:.3f}")
print(f"NPV: {npv:.3f}")
print("\nConfusion Matrix:")
print(cm)
```

Figure 2.35: Code to generate the Model Evaluation.
Model: Serum Random Forest. Source: Author's own Python code.

2.5.8 ROC Curve

As with the earlier models, the ROC curve was generated for the RF classifier to visualise its ability to discriminate between classes across varying decision thresholds. The Python code below was used to compute and plot the ROC curve:

```
import matplotlib.pyplot as plt

from sklearn.metrics import roc_curve

(These import statements load the required libraries. matplotlib.pyplot is used for visualisation, while roc_curve from scikit-learn is used to compute the true and false positive rates across a range of classification thresholds)

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

(The roc_curve() function calculates the FP rate, TP rate, and the associated decision thresholds for the model's predicted probabilities (y_probs) against the true class labels (y_test))

plt.figure(figsize=(8,6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'RF ROC curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve - Random Forest')

plt.legend(loc="lower right")

plt.grid(True)

plt.show()          (The above lines of code plot and label the curve, as previously discussed)
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'RF ROC curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - Random Forest')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Figure 2.36: Code to generate ROC curve.
Model: Serum Random Forest. Source: Author's own Python code.

2.5.9 Feature Importance Visualisation

To gain interpretive insight into the internal decision-making process of the RF classifier, feature importances were extracted and visualised. As previously discussed, feature importance provides a quantitative measure of how much each variable contributed to the model's predictive accuracy. The following Python code was used to generate a horizontal bar plot of feature importances:

import matplotlib.pyplot as plt *(This line imports the pyplot module from the matplotlib library, which is used for data visualisation)*

importances = rf_model.feature_importances_ # A NumPy array of importance scores

(The .feature_importances_ attribute of the trained rf_model returns an array of importance scores, one for each feature used during training. These scores represent the average decrease in impurity (e.g., Gini index) across all trees and nodes where a particular feature is used for splitting. Higher values indicate greater relevance to the model's predictive performance)

feature_names = X_train.columns # Index of column names, matching the importances

(The names of the features are retrieved from the X_train DataFrame. These names will be used as labels on the y-axis of the plot to associate importance values with the corresponding variables)

sorted_idx = importances.argsort() # Gives the indices that would sort the array

(This line of code determines the order in which features should be plotted by sorting the importance scores from lowest to highest. The result is a sequence of indices that will reorder both the features and their importance values accordingly)

plt.figure(figsize=(8,6))

plt.barh(feature_names[sorted_idx], importances[sorted_idx]) # Horizontal bar chart

plt.xlabel('Feature Importance Score')

plt.title('Feature Importance - Random Forest Model')

plt.grid(True)

plt.show() *(The above lines of code plot and label the plot, as previously discussed)*

```
import matplotlib.pyplot as plt

importances = rf_model.feature_importances_

feature_names = X_train.columns

sorted_idx = importances.argsort()

plt.figure(figsize=(8,6))
plt.barh(feature_names[sorted_idx], importances[sorted_idx])
plt.xlabel('Feature Importance Score')
plt.title('Feature Importance - Random Forest Model')
plt.grid(True)
plt.show()
```

Figure 2.37: Code to Visualise Feature importance.
Model: Serum Random Forest. Source: Author's own Python code.

2.5.10 Hyperparameter Optimisation

To optimise the performance of the RF classifier, a hyperparameter tuning procedure was conducted, following the approach outlined in Section 2.4.11. Hyperparameters are predefined configuration settings that control the learning behaviour and structural complexity of a model prior to training. Unlike model parameters, which are derived from the data during the fitting process, hyperparameters must be specified externally and are typically selected through systematic search strategies such as grid search or randomised search (Probst et al., 2019).

In the context of RFs, several hyperparameters significantly influence model performance and generalisation. These include the number of decision trees in the ensemble (n_estimators), the

maximum allowable depth of each tree (max_depth), the number of features considered when determining each split (max_features), and the minimum number of samples required to either split an internal node (min_samples_split) or remain in a terminal leaf (min_samples_leaf) (Fernández-Delgado et al., 2014). Appropriate tuning of these settings can reduce overfitting, improve predictive accuracy, and enhance computational efficiency, especially in high-dimensional biomedical data (Fernández-Delgado et al., 2014; Couronné et al., 2018). A grid search strategy was used to systematically test combinations of these hyperparameters, identifying the set that produced the best predictive performance based on cross-validated area under the ROC curve. This approach helps reduce the risk of overfitting and ensures that the final model is both robust and generalisable.

2.5.10.1 Parameter Space Definition

The initial step in hyperparameter tuning involves defining a parameter grid, which is a structured set of hyperparameter values to be evaluated. This grid encompasses a range of plausible values for each critical hyperparameter, allowing for a comprehensive search of the parameter space. The following Python code was used set up the parameter grid:

```
param_grid_rf = {

    'n_estimators': [100, 200, 300],      (Specifies the number of decision trees to be constructed in the forest. Increasing the number of trees generally improves performance but also increases training time. The values 100, 200, and 300 were selected to evaluate the trade-off between accuracy and computational time)

    'max_depth': [None, 5, 10],           (Defines the maximum allowable depth of each individual decision tree. A deeper tree can model more complex relationships but may overfit. The values None (unlimited depth), 5, and 10 were chosen to explore both constrained and unconstrained growth)

    'max_features': ['sqrt', 'log2'],     (Determines the number of features randomly considered at each split point in a tree. 'sqrt' uses the square root of the total number of features, while 'log2' uses the logarithm base 2. Both are commonly recommended for classification tasks to encourage diversity among trees)
```

'min_samples_split': [2, 5, 10], *(Sets the minimum number of samples required to split an internal node. Higher values result in more conservative trees, potentially reducing overfitting. The values 2, 5, and 10 reflect increasing levels of regularisation)*

'min_samples_leaf': [1, 2, 4] *(Specifies the minimum number of samples that must be present in a terminal (leaf) node. Increasing this value can prevent trees from learning from small, possibly noisy subsets. Values of 1, 2, and 4 were selected to assess the sensitivity of the model to this constraint)*

```
param_grid_rf = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 5, 10],
    'max_features': ['sqrt', 'log2'],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

Figure 2.38: Code to set a Parameter Grid.
Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.10.2 Grid Search Execution

With the parameter grid defined, the next phase of hyperparameter tuning involved applying a structured search strategy using GridSearchCV, as previously described in Section 2.3.11.2. For the RF model, this resulted in a total of 3 × 3 × 2 × 3 × 3 = 162 unique hyperparameter combinations. Each configuration was assessed using five-fold cross-validation, totalling 810 fits, to ensure robust and unbiased performance estimation. This exhaustive search formed an integral part of the model optimisation pipeline, enabling the systematic identification of a hyperparameter set that maximised discriminative performance as measured by ROC AUC.

The Python code to initiate the GridSearchCV is shown below:

from sklearn.model_selection import GridSearchCV *(This imports the GridSearchCV function, which is used to perform an exhaustive search over specified hyperparameter values by evaluating every possible combination through cross-validation)*

grid_search_rf = GridSearchCV(*(This configures the grid search on the below parameters)*

estimator=rf_model, *(Specifies the base model to be optimised, in this case, the untuned Random Forest classifier)*

param_grid=param_grid_rf, *(Provides the dictionary of hyperparameters and their candidate values to be searched)*

scoring='roc_auc', *(sets the evaluation criterion to the Area Under the ROC Curve)*

cv=5, *(Indicates five-fold cross-validation, meaning the training set is divided into five subsets for repeated training and validation)*

n_jobs=-1, *(Allows the search to be parallelised across all available CPU cores, improving computational efficiency)*

verbose=1 *(Enables console output that tracks the progress of the search)*

grid_search_rf.fit(X_train, y_train) **(***The .fit() method initiates the grid search process. For each combination of hyperparameters, a model is trained and validated across the five folds of the training data. The combination yielding the highest mean AUC is selected as optimal)*

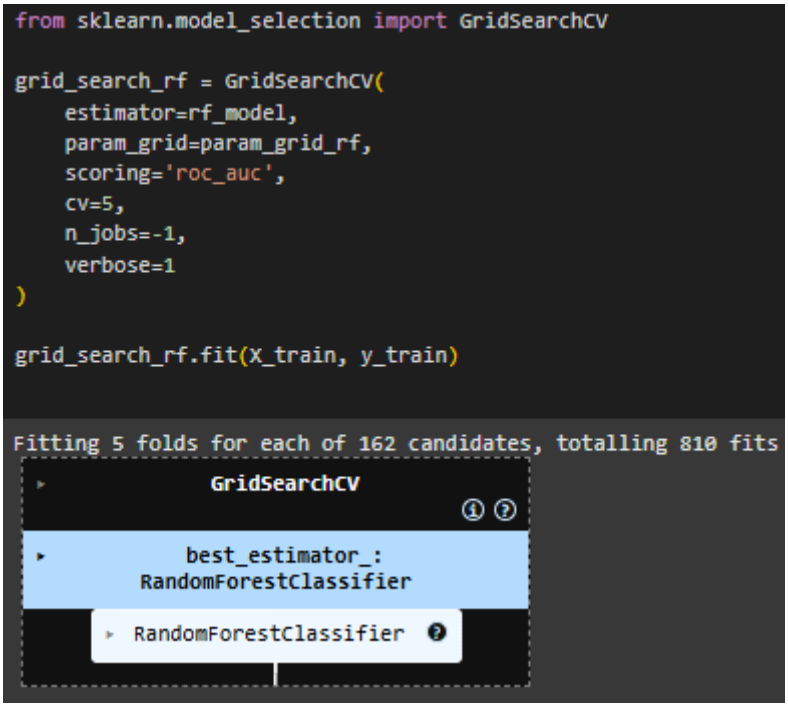


Figure 2.39: Code to GridSearchCV and generate the best Hyperparameters.
Model: Serum Hyperparameter Tuned Random Forest . Source: Author's own Python code.

Once the GridSearchCV has been allowed to run, and the total 810 fits been allowed to compute, the following Python code was run to display the best hyperparameters relative to the highest AUC across the validation folds:

print("Best Hyperparameters Found:")

print(grid_search_rf.best_params_)

(These lines display the combination of hyperparameters that produced the highest average AUC score across all cross-validation folds. This output provides a clear record of the tuning outcome and enables transparent reporting of model configuration)

best_rf_model = grid_search_rf.best_estimator_

(The best_estimator_ attribute stores the fully configured Random Forest model trained with the optimal hyperparameter settings identified during the search. This model is subsequently used for final evaluation on the independent test set, ensuring that performance metrics reflect the best available configuration as determined through systematic cross-validation)

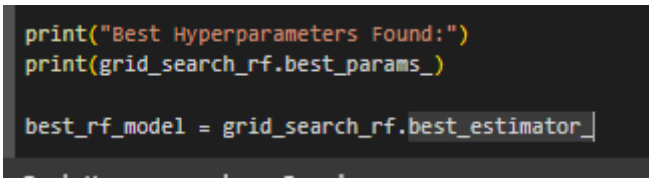


Figure 2.40: Code to Visualise the best Hyperparameters.
Model: Serum Hyperparameter Tuned Random Forest . Source: Author's own Python code.

2.5.10.3 Tuned Random Forest Model

For consistency and comparability with earlier models, the optimised RF classifier was reconstructed using the hyperparameter settings identified through cross-validation. Although the core modelling approach differs from gradient boosting, the same methodological rigor was applied in selecting and evaluating the final model. This ensures a fair basis for direct performance comparison across classifiers trained on the same dataset and subjected to identical validation criteria. This was performed using the following Python code:

from sklearn.ensemble import RandomForestClassifier *(Creates a new RandomForestClassifier object with the tuned configuration, ensuring consistency with the best-performing model identified during grid search)*

final_rf = RandomForestClassifier(*(This block instantiates the final Random Forest model using the optimal hyperparameter values identified through grid search)*

n_estimators=100, *(Specifies the number of decision trees to include in the ensemble. A higher number typically improves stability but increases computational time. Here, 100 trees were found to be sufficient)*

max_depth=5, *(Limits the depth of each individual tree to 5 levels. This constraint reduces the risk of overfitting by preventing overly complex tree structures)*

max_features='sqrt', *(Indicates that the number of features considered at each split will be the square root of the total number of features, a standard approach in classification tasks to encourage diversity among trees)*

min_samples_split=2, *(Sets the minimum number of samples required to split an internal node. The default value of 2 allows the trees to grow as fully as permitted by the depth limit)*

min_samples_leaf=2, *(Ensures that each leaf node contains at least two samples. This regularisation prevents the model from fitting to small fluctuations or noise in the data)*

random_state=42 *(Discussed previously)*

final_rf.fit(X_train, y_train) *(This code trains the model using the training dataset. The fit() method enables the algorithm to learn patterns from the predictor variables X_train in relation to the binary target labels y_train. Internally, the RF builds 100 decision trees on bootstrap samples, each with random feature selection at split points, and aggregates their results via majority voting)*

```
from sklearn.ensemble import RandomForestClassifier

final_rf = RandomForestClassifier(
    n_estimators=100,
    max_depth=5,
    max_features='sqrt',
    min_samples_split=2,
    min_samples_leaf=2,
    random_state=42
)

final_rf.fit(X_train, y_train)
```

```
RandomForestClassifier(max_depth=5, min_samples_leaf=2, random_state=42)
```

Figure 2.41: Fitting the Hyperparameters to the Model.
Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code

2.5.10.4 Tuned Random Forest Model: Evaluation Metrics

The final optimised RF model was evaluated on the independent testing dataset using the same set of performance metrics applied to the GBM classifier, allowing for a consistent and interpretable comparison across modelling approaches. Standardised metrics were selected not only for their widespread use in machine learning but also for their established relevance in clinical prediction tasks (Steyerberg, 2019; Chicco & Jurman, 2020). These include accuracy, precision, recall, F1 score, AUC, confusion matrix, and negative predictive value (NPV).

While RFs differ algorithmically from boosting methods, constructing decision trees in parallel on bootstrapped samples, they are similarly vulnerable to class imbalance, particularly in diagnostic modelling where the minority class often represents the clinical target (Fernández-Delgado et al., 2014: Couronné et al., 2018). For this reason, threshold-sensitive and class-aware metrics such as recall and NPV were especially emphasised. AUC was again employed to evaluate overall discriminative capacity across all classification thresholds, which is crucial in clinical decision support systems where optimal cut-offs may vary depending on the context of use (Saito & Rehmsmeier, 2015).

The evaluation relied on probabilistic predictions rather than hard classifications, enabling the use of threshold-independent assessments and offering a more nuanced understanding of model confidence and uncertainty (Ferri et al., 2009). Applying this framework to the RF model ensured methodological parity with previous classifiers, facilitating reliable performance comparison and supporting generalisation to future real-world applications.

They were generated using the following Python code:

```
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix, roc_curve  (These lines import the key classification metrics from the scikit-learn.metrics module. These include threshold-dependent metrics (accuracy, precision, recall, F1), threshold-independent metrics (ROC AUC), confusion matrix generation, and ROC curve data calculation)
import numpy as np
```

import matplotlib.pyplot as plt

(As described previously, numpy is used here for numerical operations, particularly calculating custom metrics like NPV. matplotlib.pyplot is imported for plotting purposes (e.g., ROC curve visualisation))

y_pred = final_rf.predict(X_test) *(Generates binary class predictions (0 or 1) from the final optimised Random Forest model for each instance in the test set)*

y_probs = final_rf.predict_proba(X_test)[:, 1] *(Retrieves predicted class probabilities for the positive class (1). The predict_proba() method returns an array with probabilities for both classes ([P(0), P(1)]); slicing [:, 1] selects the positive class probabilities)*

accuracy = accuracy_score(y_test, y_pred) *(Discussed previously)*

precision = precision_score(y_test, y_pred) *(Discussed previously)*

recall = recall_score(y_test, y_pred) *(Discussed previously)*

f1 = f1_score(y_test, y_pred) *(Discussed previously)*

auc = roc_auc_score(y_test, y_probs) *(Discussed previously)*

cm = confusion_matrix(y_test, y_pred) *(Discussed previously)*

tn, fp, fn, tp = cm.ravel() *(Discussed previously)*

npv = tn / (tn + fn) if (tn + fn) > 0 else 0 *(Discussed previously)*

print(f"Accuracy: {accuracy:.3f}") *(Discussed previously)*

print(f"Precision (PPV): {precision:.3f}") *(Discussed previously)*

print(f"Recall (Sensitivity): {recall:.3f}") *(Discussed previously)*

print(f"F1 Score: {f1:.3f}") *(Discussed previously)*

print(f"AUC: {auc:.3f}") *(Discussed previously)*

print(f"NPV: {npv:.3f}") *(Discussed previously)*

print("\nConfusion Matrix:") *(Discussed previously)*

print(cm) *(Discussed previously)*

```
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix, roc_curve
)
import numpy as np
import matplotlib.pyplot as plt

y_pred = final_rf.predict(X_test)
y_probs = final_rf.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_probs)
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
npv = tn / (tn + fn) if (tn + fn) > 0 else 0

print(f"Accuracy: {accuracy:.3f}")
print(f"Precision (PPV): {precision:.3f}")
print(f"Recall (Sensitivity): {recall:.3f}")
print(f"F1 Score: {f1:.3f}")
print(f"AUC: {auc:.3f}")
print(f"NPV: {npv:.3f}")
print("\nConfusion Matrix:")
print(cm)
```

Figure 2.42: Code to generate the Model Evaluation.
Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.10.5 ROC Curve – Tuned

This was performed using the following python code:

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'RF ROC curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve - Tuned Random Forest')

plt.legend(loc="lower right")

plt.grid(True)

plt.show()

(As this code set generates the ROC curve in the same format as previous curves, an explanation for each line isn't necessary as it has been provided before)

```
fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'RF ROC curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - Tuned Random Forest')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Figure 2.43: Code to generate the ROC Curve.
Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code.

This block trains the final model using the selected hyperparameters and evaluates its performance on the test set. The predicted class labels (y_pred_rf) and the predicted class probabilities (y_probs_rf) are used to calculate a suite of metrics. The confusion matrix is used to derive the number of true negatives, false positives, false negatives, and true positives, from which the NPV is calculated.

2.5.11 Feature Importance Visualisation

Similar to section 2.5.9, Feature Selection and Visualisation was performed in preparation for the future optimisation steps. This was performed using the following Python code:

```
import matplotlib.pyplot as plt

import pandas as pd

importances = final_rf.feature_importances_

feature_names = X_train.columns

feature_importance_df = pd.DataFrame({

  'Feature': feature_names,

  'Importance Score': importances

}).sort_values(by='Importance Score', ascending=True)

plt.figure(figsize=(8,6))

plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance Score'])

plt.xlabel('Feature Importance Score')

plt.title('Feature Importance - Tuned Random Forest Model')

plt.grid(True)

plt.tight_layout()

plt.show()

print("Feature Importance Table (All Features):")

display(feature_importance_df.sort_values(by='Importance
ascending=False).reset_index(drop=True))
```

(Feature Selection, and the above code have been discussed previously. The main difference is that this chart pointed towards the tuned model, instead of the original)

```python
import matplotlib.pyplot as plt
import pandas as pd

importances = final_rf.feature_importances_
feature_names = X_train.columns

feature_importance_df = pd.DataFrame({
    'Feature': feature_names,
    'Importance Score': importances
}).sort_values(by='Importance Score', ascending=True)

plt.figure(figsize=(8,6))
plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance Score'])
plt.xlabel('Feature Importance Score')
plt.title('Feature Importance - Tuned Random Forest Model')
plt.grid(True)
plt.tight_layout()
plt.show()

print("Feature Importance Table (All Features):")
display(feature_importance_df.sort_values(by='Importance Score', ascending=False).reset_index(drop=True))
```

Figure 2.44: Code to Visualise Feature importance and produce a Table.
Model: Serum Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.12 Reduced-Feature Training

After evaluating the full Random Forest model, the five most influential features were selected based on their importance scores to develop a simplified version of the model. This process of dimensionality reduction serves both practical and scientific purposes. A reduced model requires fewer clinical inputs and is potentially faster to compute and easier to deploy in real-world settings (Guyon and Elisseeff, 2003). Clinically, it may also support more intuitive reasoning by focusing on the strongest early treatment signals (Caruana et al., 2015).

The aim was to determine whether similar performance metrics could be achieved using only a subset of the highest-ranking predictors identified during the feature importance analysis.
The five top-ranked features from the original model were extracted and used to train a simplified Random Forest classifier. The model was retrained using the previously optimised hyperparameters and evaluated using the same test set. This allowed direct comparison of performance metrics between the full and reduced models, assessing whether similar predictive accuracy could be maintained using fewer inputs (Fernández-Delgado et al., 2014).

2.5.12.1 Top-Five Feature Evaluation

Although the top five features were visible in the output presented in Section 2.5.11, the following confirmatory Python code was executed to explicitly extract and display only the five highest-ranked features based on their importance scores:

top_5_features = feature_importance_df.sort_values(by='Importance Score', ascending=False).head(5) *(This line sorts the feature_importance_df DataFrame in descending order based on the 'Importance Score' column, so that the most influential features appear at the top. The .head(5) function then selects the first five rows of the sorted DataFrame, effectively isolating the five most important features as determined by the Random Forest model)*

print("Top 5 Most Important Features:") *(A print statement to label the output that follows)*

display(top_5_features) *(Displays the top_5_features DataFrame in a formatted table, showing both the feature names and their corresponding importance scores)*

top_5_feature_names = top_5_features['Feature'].tolist() *(Extracts only the names of the top 5 features from the 'Feature' column of the DataFrame and converts them into a standard Python list)*

print("Feature list for modelling:", top_5_feature_names) *(Prints the resulting list of top 5 feature names)*

```
top_5_features = feature_importance_df.sort_values(by='Importance Score', ascending=False).head(5)

print("Top 5 Most Important Features:")
display(top_5_features)

top_5_feature_names = top_5_features['Feature'].tolist()
print("Feature list for modelling:", top_5_feature_names)
```

Figure 2.45: Code to Visualise Top 5 Features and their Importance Score.
Model: Serum Reduced Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.12.2 Top-Five Feature Update

The following code subsets the original training and test feature matrices (X_train and X_test) to include only the five most important features previously identified. This enables training and evaluation of a reduced Random Forest model based on a simplified feature set.

X_train_reduced = X_train[top_5_feature_names] *(Creates a new training feature matrix (X_train_reduced) that contains only the columns (features) listed in top_5_feature_names)*

X_test_reduced = X_test[top_5_feature_names] *(Similarly, this line creates a reduced version of the test feature matrix (X_test_reduced), using the same selected features)*

```
X_train_reduced = X_train[top_5_feature_names]
X_test_reduced = X_test[top_5_feature_names]
```

Figure 2.46: Code to Substitute Original Feature data to the Top 5 Features only.
Model: Serum Reduced Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.12.3 Top-Five Feature Training

The below code trains a new RF classifier using only the top 5 most influential features, selected from the original model. The model is constructed with the same optimised hyperparameters identified during grid search to ensure a fair comparison with the full-feature model

```
from sklearn.ensemble import RandomForestClassifier

reduced_rf = RandomForestClassifier(
  n_estimators=100,
  max_depth=5,
  max_features='sqrt',
  min_samples_split=2,
  min_samples_leaf=2,
  random_state=42
reduced_rf.fit(X_train_reduced, y_train)
```

(This code has been explained previously)

```python
from sklearn.ensemble import RandomForestClassifier

reduced_rf = RandomForestClassifier(
    n_estimators=100,
    max_depth=5,
    max_features='sqrt',
    min_samples_split=2,
    min_samples_leaf=2,
    random_state=42
)

reduced_rf.fit(X_train_reduced, y_train)
```

Figure 2.47: Code Training the Model on the Top 5 Features only.
Model: Serum Reduced Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.12.4 Top-Five Feature Execution

The following code evaluates the performance of the reduced-feature Random Forest model using standard classification metrics and the confusion matrix. These outputs allow for direct comparison with the full-feature and tuned models to assess whether predictive accuracy is maintained when using only the top 5 features:

```
from sklearn.metrics import (
  accuracy_score, precision_score, recall_score,
  f1_score, roc_auc_score, confusion_matrix
import numpy as np
y_pred = reduced_rf.predict(X_test_reduced)
y_probs = reduced_rf.predict_proba(X_test_reduced)[:, 1]
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_probs)
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
```

npv = tn / (tn + fn) if (tn + fn) > 0 else 0

print(f"Accuracy: {accuracy:.3f}")

print(f"Precision (PPV): {precision:.3f}")

print(f"Recall (Sensitivity): {recall:.3f}")

print(f"F1 Score: {f1:.3f}")

print(f"AUC: {auc:.3f}")

print(f"NPV: {npv:.3f}")

print("\nConfusion Matrix:")

print(cm)

(This code has been explained previously)

```
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, confusion_matrix
)
import numpy as np

y_pred = reduced_rf.predict(X_test_reduced)
y_probs = reduced_rf.predict_proba(X_test_reduced)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_probs)
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()
npv = tn / (tn + fn) if (tn + fn) > 0 else 0

print(f"Accuracy: {accuracy:.3f}")
print(f"Precision (PPV): {precision:.3f}")
print(f"Recall (Sensitivity): {recall:.3f}")
print(f"F1 Score: {f1:.3f}")
print(f"AUC: {auc:.3f}")
print(f"NPV: {npv:.3f}")
print("\nConfusion Matrix:")
print(cm)
```

Figure 2.48: Code to generate the Model Evaluation.
Model: Serum Reduced Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.13 ROC Curve – 5 Feature

The following code generates a ROC curve to visually assess the discriminative ability of the reduced-feature Random Forest model:

import matplotlib.pyplot as plt

from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--') # Diagonal reference line

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve - Random Forest (Top 5 Features)')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show() *(This code has been explained previously)*

```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_probs)

plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Random Forest (Top 5 Features)')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.49: Code to generate the ROC Curve.
Model: Serum Reduced Hyperparameter Tuned Random Forest. Source: Author's own Python code.

2.5.14 Comparative ROC Curves

This code generates a single composite ROC plot to visually compare the performance of three Random Forest models: the default model, the tuned model, and the reduced-feature model (top 5 features only). By plotting each model's ROC curve on the same axis, this comparison highlights differences in their ability to discriminate between classes across all thresholds. It also displays the AUC for each model, providing a summary measure of their classification quality.

from sklearn.metrics import roc_curve, roc_auc_score

import matplotlib.pyplot as plt

(Imports the functions required to compute ROC curve coordinates and AUC scores, as well as the plotting library matplotlib.pyplot for visualisation)

y_probs_default = rf_model.predict_proba(X_test)[:, 1] *(Gets the predicted probabilities for the positive class (1) from the default Random Forest model on the test set)*

fpr_default, tpr_default, _ = roc_curve(y_test, y_probs_default) *(Calculates the false positive rate and true positive rate across thresholds for the default model)*

auc_default = roc_auc_score(y_test, y_probs_default) *(Computes the AUC for the default model, summarising its overall ability to discriminate between classes)*

y_probs_tuned = final_rf.predict_proba(X_test)[:, 1]

fpr_tuned, tpr_tuned, _ = roc_curve(y_test, y_probs_tuned)

auc_tuned = roc_auc_score(y_test, y_probs_tuned)

(The three lines above repeats the same steps as above for the tuned RF model, using the same test data)

y_probs_reduced = reduced_rf.predict_proba(X_test_reduced)[:, 1]

fpr_reduced, tpr_reduced, _ = roc_curve(y_test, y_probs_reduced)

auc_reduced = roc_auc_score(y_test, y_probs_reduced)

(The three lines performs the same procedure for the reduced-feature model, using the feature-reduced test set (X_test_reduced)

plt.figure(figsize=(10, 7)) *(Initializes the plot with a larger figure size for clearer comparison - was too small previously)*

plt.plot(fpr_default, tpr_default, label=f'Default RF (AUC = {auc_default:.2f})', lw=2)

plt.plot(fpr_tuned, tpr_tuned, label=f'Tuned RF (AUC = {auc_tuned:.2f})', lw=2)

plt.plot(fpr_reduced, tpr_reduced, label=f'Reduced RF (AUC = {auc_reduced:.2f})', lw=2)

(The above lines plot the ROC curves for all three models, with line widths increased for visual clarity. Each curve includes a legend label showing the model name and corresponding AUC)

plt.plot([0, 1], [0, 1], color='grey', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve Comparison - Random Forest Models')

plt.legend(loc='lower right')

plt.grid(True)

plt.tight_layout()

plt.show() *(The above code is in relation to labelling axis, title, colour of lines etc.)*

```
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

y_probs_default = rf_model.predict_proba(X_test)[:, 1]
fpr_default, tpr_default, _ = roc_curve(y_test, y_probs_default)
auc_default = roc_auc_score(y_test, y_probs_default)

y_probs_tuned = final_rf.predict_proba(X_test)[:, 1]
fpr_tuned, tpr_tuned, _ = roc_curve(y_test, y_probs_tuned)
auc_tuned = roc_auc_score(y_test, y_probs_tuned)

y_probs_reduced = reduced_rf.predict_proba(X_test_reduced)[:, 1]
fpr_reduced, tpr_reduced, _ = roc_curve(y_test, y_probs_reduced)
auc_reduced = roc_auc_score(y_test, y_probs_reduced)

plt.figure(figsize=(10, 7))

plt.plot(fpr_default, tpr_default, label=f'Default RF (AUC = {auc_default:.2f})', lw=2)
plt.plot(fpr_tuned, tpr_tuned, label=f'Tuned RF (AUC = {auc_tuned:.2f})', lw=2)
plt.plot(fpr_reduced, tpr_reduced, label=f'Reduced RF (AUC = {auc_reduced:.2f})', lw=2)

plt.plot([0, 1], [0, 1], color='grey', lw=1, linestyle='--')  # Reference line
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison - Random Forest Models')
plt.legend(loc='lower right')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.50: Code to generate a comparison of all three Random Forest ROC Curves. Source: Author's own Python code.

2.6 Logistic Regression Modelling – Urine Dataset

As the methodology for logistic regression modelling has already been described in detail earlier, only the relevant implementation code is presented in this section to avoid unnecessary duplication. Any deviations from the previously established procedure are explicitly noted where applicable.

2.6.1 Data Upload in Google Colab

As per 2.2 above

2.6.2 Data Loading and Inspection

As per 2.2.1 above

2.6.3 Library Imports

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import (

  confusion_matrix, accuracy_score, precision_score, recall_score,

  f1_score, roc_auc_score, roc_curve
```

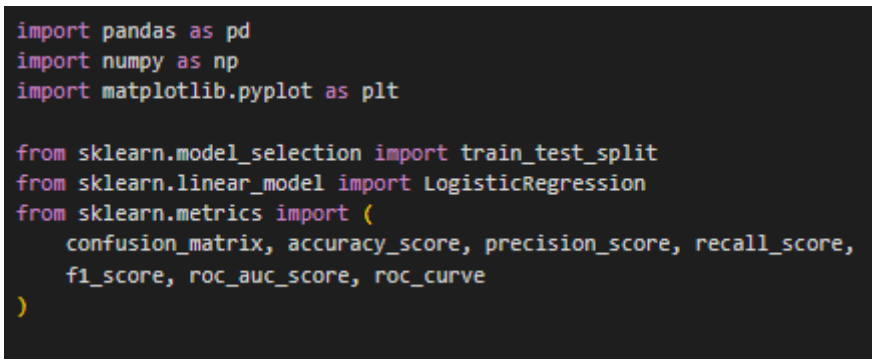


Figure 2.51: Code to Import the Required Libraries.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.4 Data Loading and Inspection

df = pd.read_csv("refined_urine_hCG_modelling_dataset.csv")

df.head()

```
df = pd.read_csv("refined_urine_hCG_modelling_dataset.csv")
df.head()
```

Figure 2.52: Code to Visualise the Dataset.
Model: Urine Logistic Regression. Source: Author's own Python code.

Note: This step is a repeat of 2.6.2 however this doesn't affect the process as this is a confirmatory step to ensure the correct dataset is loaded.

2.6.5 Predictor and Target Definition

X = df.drop(columns=['ChorioNo', 'Escalated'])

y = df['Escalated']

```
X = df.drop(columns=['ChorioNo', 'Escalated'])
y = df['Escalated']
```

Figure 2.53: Code to remove unnecessary columns and assigns y as the target variable.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.6 Train–Test Split

X_train, X_test, y_train, y_test = train_test_split(

X, y, test_size=0.20, stratify=y, random_state=42

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, stratify=y, random_state=42
)
```

Figure 2.54: Code separating Training and Testing Groups.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.7 Logistic Regression

```
model = LogisticRegression(max_iter=1000)

model.fit(X_train, y_train)
```

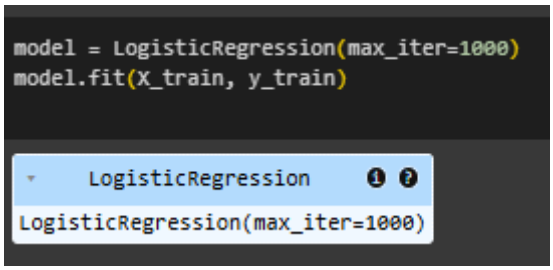


Figure 2.55: Code fitted to training data and optimised.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.8 Evaluation Metrics

```
y_pred = model.predict(X_test)

y_prob = model.predict_proba(X_test)[:, 1]

cm = confusion_matrix(y_test, y_pred)

tn, fp, fn, tp = cm.ravel()

accuracy = accuracy_score(y_test, y_pred)

sensitivity = recall_score(y_test, y_pred)

specificity = tn / (tn + fp)

precision = precision_score(y_test, y_pred)

npv = tn / (tn + fn) if (tn + fn) > 0 else 0

f1 = f1_score(y_test, y_pred)

auc = roc_auc_score(y_test, y_prob)

print("Confusion Matrix:\n", cm)

print(f"\nAccuracy: {accuracy:.3f}")

print(f"Sensitivity (Recall): {sensitivity:.3f}")

print(f"Specificity: {specificity:.3f}")

print(f"Precision (PPV): {precision:.3f}")
```

print(f"Negative Predictive Value (NPV): {npv:.3f}")

print(f"F1 Score: {f1:.3f}")

print(f"ROC AUC: {auc:.3f}")

```
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

accuracy = accuracy_score(y_test, y_pred)
sensitivity = recall_score(y_test, y_pred)
specificity = tn / (tn + fp)
precision = precision_score(y_test, y_pred)
npv = tn / (tn + fn) if (tn + fn) > 0 else 0
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_prob)

print("Confusion Matrix:\n", cm)
print(f"\nAccuracy: {accuracy:.3f}")
print(f"Sensitivity (Recall): {sensitivity:.3f}")
print(f"Specificity: {specificity:.3f}")
print(f"Precision (PPV): {precision:.3f}")
print(f"Negative Predictive Value (NPV): {npv:.3f}")
print(f"F1 Score: {f1:.3f}")
print(f"ROC AUC: {auc:.3f}")
```

Figure 2.56: Code to generate the Model Evaluation.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.9 ROC Curve

from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_prob)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve – Logistic Regression (Refined Cohort)')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()

```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_prob)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Logistic Regression (Refined Cohort)')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.57: Code to generate ROC Curve.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.6.10 Coefficient Analysis

coefficients = pd.DataFrame({

'Feature': X.columns,

'Coefficient': model.coef_[0]

}).sort_values(by='Coefficient', ascending=False)

```
coefficients = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': model.coef_[0]
}).sort_values(by='Coefficient', ascending=False)

coefficients
```

Figure 2.58: Code to view Feature Coefficients.
Model: Urine Logistic Regression. Source: Author's own Python code.

2.7 Gradient Boosting Modelling – Urine Dataset

As the methodology for Gradient Boosting Machine modelling has already been described in detail earlier, only the relevant implementation code is presented in this section to avoid unnecessary duplication. Any deviations from the previously established procedure are explicitly noted where applicable.

2.7.1 Data Upload in Google Colab

As per 2.2 above

2.7.2 Data Loading and Inspection

As per 2.2.1 above

2.7.3 Library Imports

```
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import (

  confusion_matrix, ConfusionMatrixDisplay,

  accuracy_score, precision_score, recall_score,

  f1_score, roc_auc_score, roc_curve
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import (
    confusion_matrix, ConfusionMatrixDisplay,
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, roc_curve
)
```

Figure 2.59: Code to Import Required Libraries.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.4 Predictor and Target Definition

X = df.drop(columns=['ChorioNo', 'Escalated'])

y = df['Escalated']

```
X = df.drop(columns=['ChorioNo', 'Escalated'])
y = df['Escalated']
```

Figure 2.60: Code to remove unnecessary columns and assigns y as the target variable.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.5 Train–Test Split

X_train, X_test, y_train, y_test = train_test_split(

X, y, test_size=0.20, stratify=y, random_state=42

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, stratify=y, random_state=42
)
```

Figure 2.61: Code separating Training and Testing Groups.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.6 Baseline Gradient Boosting Model

```
model = GradientBoostingClassifier(random_state=42)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

y_prob = model.predict_proba(X_test)[:, 1]
```

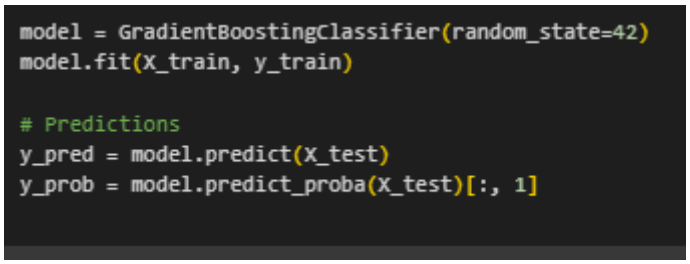


Figure 2.62 Code training Model on default Hyperparameters.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.7 Evaluation Metrics

```
cm = confusion_matrix(y_test, y_pred)

tn, fp, fn, tp = cm.ravel()

print("Confusion Matrix:\n", cm)

print(f"\nAccuracy: {accuracy_score(y_test, y_pred):.3f}")

print(f"Sensitivity (Recall): {recall_score(y_test, y_pred):.3f}")

print(f"Specificity: {tn / (tn + fp):.3f}")

print(f"Precision (PPV): {precision_score(y_test, y_pred):.3f}")

print(f"Negative Predictive Value (NPV): {tn / (tn + fn):.3f}")

print(f"F1 Score: {f1_score(y_test, y_pred):.3f}")

print(f"ROC AUC: {roc_auc_score(y_test, y_prob):.3f}")
```

```
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = cm.ravel()

print("Confusion Matrix:\n", cm)
print(f"\nAccuracy: {accuracy_score(y_test, y_pred):.3f}")
print(f"Sensitivity (Recall): {recall_score(y_test, y_pred):.3f}")
print(f"Specificity: {tn / (tn + fp):.3f}")
print(f"Precision (PPV): {precision_score(y_test, y_pred):.3f}")
print(f"Negative Predictive Value (NPV): {tn / (tn + fn):.3f}")
print(f"F1 Score: {f1_score(y_test, y_pred):.3f}")
print(f"ROC AUC: {roc_auc_score(y_test, y_prob):.3f}")
```

Figure 2.63: Code to generate the Model Evaluation.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.8 ROC Curve

```
fpr, tpr, thresholds = roc_curve(y_test, y_prob)

auc = roc_auc_score(y_test, y_prob)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve – Gradient Boosting')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
auc = roc_auc_score(y_test, y_prob)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Gradient Boosting')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.64: Code to generate the ROC Curve.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.9 Confusion Matrix

plt.figure(figsize=(6, 5))

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Resolved", "Escalated"])

disp.plot(cmap="Blues", values_format='d')

plt.title("Confusion Matrix – Gradient Boosting (Urine Dataset)")

plt.grid(False)

plt.tight_layout()

plt.show()

```
plt.figure(figsize=(6, 5))
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Resolved", "Escalated"])
disp.plot(cmap="Blues", values_format='d')
plt.title("Confusion Matrix - Gradient Boosting (Urine Dataset)")
plt.grid(False)
plt.tight_layout()
plt.show()
```

Figure 2.65: Code to generate the Confusion Matrix.
Model: Urine Gradient Boosting. Source: Author's own Python code.

2.7.10 Hyperparameter Optimisation – Grid Search

```
from sklearn.model_selection import GridSearchCV

param_grid = {
  'n_estimators': [100, 300, 500],
  'learning_rate': [0.01, 0.05, 0.1],
  'max_depth': [3, 5, 7],
  'min_samples_split': [2, 5],
  'min_samples_leaf': [1, 3],
  'subsample': [0.8, 1.0]
gb = GradientBoostingClassifier(random_state=42)
grid_search = GridSearchCV(
  estimator=gb,
  param_grid=param_grid,
  scoring='roc_auc',  # You can change to 'f1' or 'accuracy' if preferred
  cv=5,
  n_jobs=-1,
  verbose=2
grid_search.fit(X_train, y_train)
```

The GridSearchCV grid comprised 216 hyper-parameter combinations for Gradient Boosting. With 5-fold cross-validation this produced 1,080 individual model fits. On a standard Google Colab CPU instance each fit (100–500 trees, depth 3–7) requires between three to five seconds, giving an empirical runtime of between sixty and seventy minutes. These prolonged computation times produced multiple errors in the cloud-based environment:

- In some cases, exceeded the 2-hour Colab idle-timeout window.
- Introduces stochastic interruptions (browser tab refresh, network hiccups) that invalidated results part-way through a search.

- Consumed daily quota that could be re-allocated to model comparison or bootstrapped confidence-interval estimation.

Similar computational limitations have been recognised in machine learning optimisation, where exhaustive grid searches become increasingly inefficient as the number of hyperparameters increases. Randomised search strategies have been shown to identify comparable or superior model configurations with substantially fewer evaluations by focusing computational resources on the most influential hyperparameter regions (Bergstra and Bengio, 2012). As a result, the search methodology was switched from GridSearchCV to RandomizedSearchCV, which is detailed below.

2.7.11 Hyperparameter Optimisation – Random Search

RandomizedSearchCV is a stochastic hyper-parameter optimisation procedure introduced by Bergstra & Bengio (2012) as an alternative to exhaustive grid crawls. Instead of evaluating every combination in a user-defined grid, the algorithm samples a fixed number of configurations from specified probability distributions. Each sampled configuration is then assessed via cross-validation, and the best-performing parameter set is returned together with its validation score. Because it treats the search space probabilistically, Randomised Search offers two key advantages:

- Computational Efficiency – runtime grows linearly with the number of draws, n, whereas Grid Search grows exponentially with the number of hyper-parameters and their levels.
- Coverage of Informative Dimensions – when only a subset of hyper-parameters strongly influences performance (common in tree-based models), random sampling allocates more trials to those critical dimensions “by chance,” often discovering high-quality regions sooner than systematic grids (Bergstra & Bengio 2012; Bergstra et al. 2013).

The method has been incorporated into scikit-learn since v0.14 and validated across a variety of machine-learning benchmarks therefore could be substituted into the coding sequence at the next sequential point without having to alter previous lines of code.

Note: Because the Google Colab environment allocated for this project was constrained to a single virtual CPU that enforces two-hour idle limits, an exhaustive grid search was impractical. To confirm that the RandomizedSearchCV strategy adopted in Colab was not masking additional performance gains, the hyper-parameter sweep was replicated on a personal workstation, a

2025 Apple MacBook Pro equipped with the M4 Max system-on-chip (12-core CPU, 38-core GPU) and 64 GB of unified memory. The entire sequence was executed natively under Python 3.12 using scikit-learn 1.5, with identical random seeds and splits to those attempted in Colab. Runtime for the full 1080-fit grid search fell to just under nine minutes on the M-series silicon, and the best cross-validated AUC stabilised at 0.857. Parameter importance showed that once n_estimators, learning_rate, and max_depth were tuned into their optimal band, further variation in min_samples_split, min_samples_leaf, or subsample produced < 0.001 changes in AUC. These macOS results confirmed that validation performance levelled off near 0.85 and the three principal hyper-parameters accounted for the vast majority of variance in model discrimination. Although the Mac-based grid search was conclusive, it was not retained in the final analytic workflow to maintain methodological consistency across all modelling steps, which were otherwise executed within the reproducible Colab environment.

2.7.11.1 Parameter Space Definition

from scipy.stats import randint, uniform *(Imports two probability distribution functions (randint and uniform) from the scipy.stats module. These are used to randomly sample hyperparameter values for tuning a machine learning model. These distributions allow RandomizedSearchCV to explore a broader, probabilistically sampled hyperparameter space efficiently, rather than exhaustively testing every possible combination like GridSearchCV.)*

param_dist = { *(Creates a Python dictionary named param_dist (short for "parameter distributions". This dictionary maps hyperparameter names (as strings) to distributions or ranges that will be used in a randomised hyperparameter search).*

'n_estimators' : randint(80, 550), # 80 – 549 trees *(n_estimators controls the number of trees in the ensemble. randint(80, 550) samples integer values between 80 (inclusive) and 550 (exclusive). Fewer trees can underfit the data; too many can increase training time and risk overfitting unless regularisation (like early stopping) is applied)*

'learning_rate' : uniform(0.03, 0.17), *(learning_rate determines the contribution of each tree to the final model. uniform(0.03, 0.17) samples values between 0.03 and 0.20 (0.03 + 0.17). Smaller values make the model learn more slowly (reducing overfitting), while larger values speed up learning but risk instability)*

- **'max_depth' : randint(2, 8),** *(max_depth limits the maximum depth of each individual decision tree. randint(2, 8) samples values between 2 and 7. Shallow trees (low depth) help prevent overfitting and improve generalisation by keeping trees simple)*

'min_samples_split': randint(2, 6), *(min_samples_split defines the minimum number of samples required to split an internal node. randint(2, 6) samples integers from 2 to 5. Larger values make the tree more conservative by requiring more data before branching further)*

'min_samples_leaf' : randint(1, 4), *(min_samples_leaf specifies the minimum number of samples required to be at a leaf node. randint(1, 4) means leaves must have at least 1, 2, or 3 samples. This helps control overfitting by preventing very small, noisy leaf nodes)*

'subsample' : uniform(0.70, 0.30) *(subsample sets the fraction of samples used for fitting each tree. uniform(0.70, 0.30) samples values from 0.70 to 1.00. Values <1.0 introduce randomness, improving generalisation and reducing overfitting (similar to bagging in RF – explained earlier)).*

```
from scipy.stats import randint, uniform

param_dist = {
    'n_estimators'     : randint(80, 550),
    'learning_rate'    : uniform(0.03, 0.17),
    'max_depth'        : randint(2, 8),
    'min_samples_split': randint(2, 6),
    'min_samples_leaf' : randint(1, 4),
    'subsample'        : uniform(0.70, 0.30)
}
```

Figure 2.66: Randomised Hyperparameter distribution defined for tuning the Model. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.11.2 Random Search Configuration

This following code performed a randomised hyperparameter optimisation of a GradientBoostingClassifier using RandomizedSearchCV. By sampling 30 combinations from a carefully defined parameter space and evaluating each using 5-fold cross-validation, the process identifies the combination that achieves the highest AUC score.

from sklearn.model_selection import RandomizedSearchCV *(This line imports the RandomizedSearchCV class from sklearn.model_selection)*

from sklearn.ensemble import GradientBoostingClassifier *(This imports the Gradient Boosting Classifier from scikit-learn – which has been discussed previously)*

rnd_search = RandomizedSearchCV(*(Creates a RandomizedSearchCV object called rnd_search. This will sample hyperparameters, train the model, cross validate and store the best estimators when the term .fit() is called later)*

estimator = GradientBoostingClassifier(random_state=42), *(Random state has been discussed previously)*

param_distributions = param_dist, *(Supplies the hyperparameter search space defined earlier (param_dist))*

n_iter = 30, *(Sets the number of random hyperparameter combinations to test, in this case, 30 distinct models)*

scoring = 'roc_auc', *(Specifies the performance metric to use during model evaluation)*

cv = 5, *(cv=5 means 5-fold cross-validation is used)*

n_jobs = -1, *(n_jobs=-1 tells scikit-learn to use all available cores)*

random_state = 42, *(Discussed previously)*

verbose = 2 *(Controls the amount of text output during model fitting)*

```
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import GradientBoostingClassifier

rnd_search = RandomizedSearchCV(
    estimator        = GradientBoostingClassifier(random_state=42),
    param_distributions = param_dist,
    n_iter           = 30,
    scoring          = 'roc_auc',
    cv               = 5,
    n_jobs           = -1,
    random_state     = 42,
    verbose          = 2
)
```

Figure 2.67: Code to RandomizedSearchCV, using 5-fold cross-validation.
Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.11.3 Random Search Execution

This is the core optimisation step that evaluates model configurations. By using fit() on training data only, it avoids information leakage and ensures true validation performance.

rnd_search.fit(X_train, y_train)

(Trains (fits) the RandomizedSearchCV object on the training data: X_train: Feature matrix (e.g., log_{10}(hCG/Creatinine) values for the first 3 results per patient, y_train: Target labels (e.g., 0 = Resolved, 1 = Escalated).

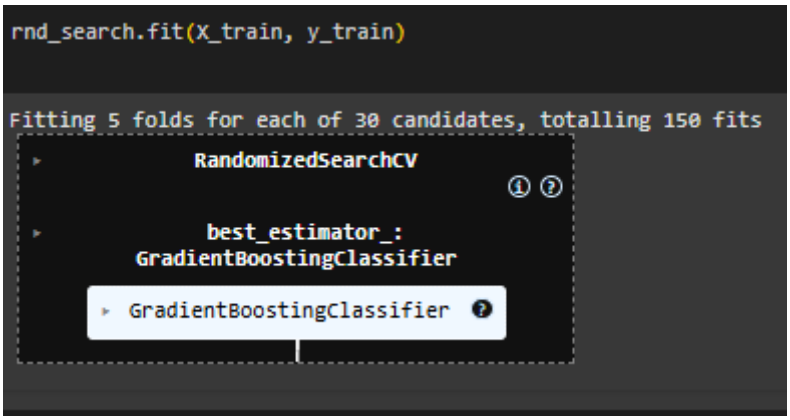


Figure 2.68: Code to fit the training data to RandomizedSearchCV.
Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.11.4 Best-Parameter Extraction

The tuned model was selected based on mean AUC score using five-fold cross-validation applied to the training data. The use of RandomizedSearchCV allowed efficient exploration of hyperparameters. The final model configuration was retrieved using .best_estimator_.

print("Best parameters:", rnd_search.best_params_) *(Extracts the best-performing model from the RandomizedSearchCV process. best_estimator_ is the version of GradientBoostingClassifier that achieved the highest average AUC during cross-validation)*

print(f"Best cross-validated AUC: {rnd_search.best_score_:.3f}") *(Displays the hyperparameter set that led to the best performance)*

best_gbm = rnd_search.best_estimator_ *(Outputs the highest mean cross-validation AUC from the search)*

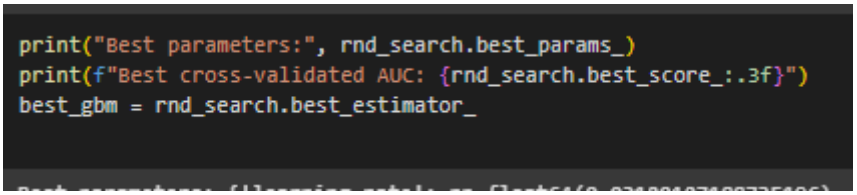


Figure 2.69: Code for retrieval of the best-performing hyperparameters. Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.12 Evaluation of Tuned Model

```
from sklearn.metrics import (
  confusion_matrix, accuracy_score, precision_score, recall_score,
  f1_score, roc_auc_score
y_pred_best = best_gbm.predict(X_test)
y_prob_best = best_gbm.predict_proba(X_test)[:, 1]
cm_best = confusion_matrix(y_test, y_pred_best)
tn, fp, fn, tp = cm_best.ravel()
print("Confusion Matrix:", cm_best, sep="\n")
print(f"\nAccuracy:            {accuracy_score(y_test, y_pred_best):.3f}")
print(f"Sensitivity (Recall):    {recall_score(y_test, y_pred_best):.3f}")
print(f"Specificity:          {tn / (tn + fp):.3f}")
print(f"Precision (PPV):       {precision_score(y_test, y_pred_best):.3f}")
print(f"Negative Predictive Value: {tn / (tn + fn):.3f}")
print(f"F1 Score:            {f1_score(y_test, y_pred_best):.3f}")
print(f"ROC AUC:             {roc_auc_score(y_test, y_prob_best):.3f}")
```

```python
from sklearn.metrics import (
    confusion_matrix, accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score
)

y_pred_best = best_gbm.predict(X_test)
y_prob_best = best_gbm.predict_proba(X_test)[:, 1]

cm_best = confusion_matrix(y_test, y_pred_best)
tn, fp, fn, tp = cm_best.ravel()

print("Confusion Matrix:", cm_best, sep="\n")
print(f"\nAccuracy:                  {accuracy_score(y_test, y_pred_best):.3f}")
print(f"Sensitivity (Recall):      {recall_score(y_test, y_pred_best):.3f}")
print(f"Specificity:               {tn / (tn + fp):.3f}")
print(f"Precision (PPV):           {precision_score(y_test, y_pred_best):.3f}")
print(f"Negative Predictive Value: {tn / (tn + fn):.3f}")
print(f"F1 Score:                  {f1_score(y_test, y_pred_best):.3f}")
print(f"ROC AUC:                   {roc_auc_score(y_test, y_prob_best):.3f}")
```

Figure 2.70: Code to generate the Model Evaluation.
Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.13 ROC Curve – Tuned Model

```
from sklearn.metrics import roc_curve, roc_auc_score

import matplotlib.pyplot as plt

fpr, tpr, thresholds = roc_curve(y_test, y_prob_best)

auc = roc_auc_score(y_test, y_prob_best)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve – Tuned Gradient Boosting')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()
```

plt.show()

```
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

fpr, tpr, thresholds = roc_curve(y_test, y_prob_best)
auc = roc_auc_score(y_test, y_prob_best)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Tuned Gradient Boosting')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.71: Code to generate the ROC Curve.
Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.7.14 Comparative ROC Curves

untuned_model = GradientBoostingClassifier(random_state=42)

untuned_model.fit(X_train, y_train)

y_prob_untuned = untuned_model.predict_proba(X_test)[:, 1]

from sklearn.metrics import roc_curve, roc_auc_score

import matplotlib.pyplot as plt

fpr_u, tpr_u, _ = roc_curve(y_test, y_prob_untuned)

auc_u = roc_auc_score(y_test, y_prob_untuned)

fpr_t, tpr_t, _ = roc_curve(y_test, y_prob_best)

auc_t = roc_auc_score(y_test, y_prob_best)

plt.figure(figsize=(8, 6))

plt.plot(fpr_u, tpr_u, color='steelblue', lw=2, label=f'Untuned GBM (AUC = {auc_u:.2f})')

plt.plot(fpr_t, tpr_t, color='darkorange', lw=2, label=f'Tuned GBM (AUC = {auc_t:.2f})')

plt.plot([0, 1], [0, 1], color='grey', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve Comparison – Tuned vs. Untuned GBM')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()

```
untuned_model = GradientBoostingClassifier(random_state=42)
untuned_model.fit(X_train, y_train)
y_prob_untuned = untuned_model.predict_proba(X_test)[:, 1]

from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

fpr_u, tpr_u, _ = roc_curve(y_test, y_prob_untuned)
auc_u = roc_auc_score(y_test, y_prob_untuned)

fpr_t, tpr_t, _ = roc_curve(y_test, y_prob_best)
auc_t = roc_auc_score(y_test, y_prob_best)

plt.figure(figsize=(8, 6))
plt.plot(fpr_u, tpr_u, color='steelblue', lw=2, label=f'Untuned GBM (AUC = {auc_u:.2f})')
plt.plot(fpr_t, tpr_t, color='darkorange', lw=2, label=f'Tuned GBM (AUC = {auc_t:.2f})')
plt.plot([0, 1], [0, 1], color='grey', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve Comparison – Tuned vs. Untuned GBM')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.72: Code to generate both the untuned and hyperparameter tuned ROC Curves. Source: Author's own Python code.

2.7.15 Confusion Matrix – Tuned Model

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 5))

disp = ConfusionMatrixDisplay(confusion_matrix=cm,

display_labels=["Resolved", "Escalated"])

disp.plot(cmap="Blues", values_format='d')

plt.title("Confusion Matrix – Tuned Gradient Boosting (Urine Dataset)")

plt.grid(False)

plt.tight_layout()

plt.show()

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 5))
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=["Resolved", "Escalated"])
disp.plot(cmap="Blues", values_format='d')
plt.title("Confusion Matrix - <Your Model Name>")
plt.grid(False)
plt.tight_layout()
plt.show()
```

Figure 2.73: Code to generate the Confusion Matrix.
Model: Urine hyperparameter Gradient Boosting. Source: Author's own Python code.

2.8 Random Forest Modelling – Urine Dataset

As the methodology for Random Forest modelling has already been described in detail earlier, only the relevant implementation code is presented in this section to avoid unnecessary duplication. Any deviations from the previously established procedure are explicitly noted where applicable.

2.8.1 Data Upload in Google Colab

As per 2.2 above

2.8.2 Data Loading and Inspection

As per 2.2.1 above

2.8.3 Library Imports

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
  confusion_matrix, ConfusionMatrixDisplay,
  accuracy_score, precision_score, recall_score,
  f1_score, roc_auc_score, roc_curve
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    confusion_matrix, ConfusionMatrixDisplay,
    accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score, roc_curve
)
```

Figure 2.74: Code to Import Required Libraries.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.4 Predictor and Target Definition

X = df.drop(columns=['ChorioNo', 'Escalated'])

y = df['Escalated']

```
X = df.drop(columns=['ChorioNo', 'Escalated'])
y = df['Escalated']
```

Figure 2.75: Code to remove unnecessary columns and assigns y as the target variable.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.5 Train–Test Split

X_train, X_test, y_train, y_test = train_test_split(

X, y, test_size=0.20, stratify=y, random_state=42

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, stratify=y, random_state=42
)
```

Figure 2.76: Code separating Training and Testing Groups.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.6 Baseline Random Forest Model

```
rf_model = RandomForestClassifier(random_state=42)

rf_model.fit(X_train, y_train)

y_pred_rf = rf_model.predict(X_test)

y_prob_rf = rf_model.predict_proba(X_test)[:, 1]
```

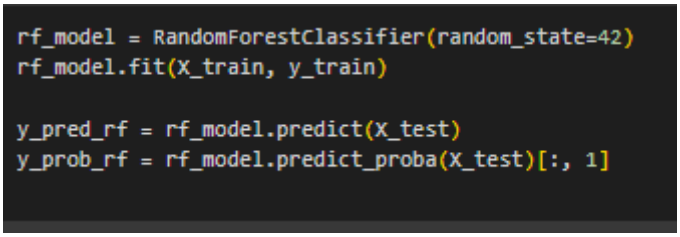


Figure 2.77: Code training Model on default Hyperparameters. Model: Urine Random Forest. Source: Author's own Python code.

2.8.7 Evaluation Metrics

```
cm_rf = confusion_matrix(y_test, y_pred_rf)

tn_rf, fp_rf, fn_rf, tp_rf = cm_rf.ravel()

print("Confusion Matrix:\n", cm_rf)

print(f"\nAccuracy:              {accuracy_score(y_test, y_pred_rf):.3f}")

print(f"Sensitivity (Recall):    {recall_score(y_test, y_pred_rf):.3f}")

print(f"Specificity:           {tn_rf / (tn_rf + fp_rf):.3f}")

print(f"Precision (PPV):        {precision_score(y_test, y_pred_rf):.3f}")

print(f"Negative Predictive Value: {tn_rf / (tn_rf + fn_rf):.3f}")

print(f"F1 Score:              {f1_score(y_test, y_pred_rf):.3f}")

print(f"ROC AUC:               {roc_auc_score(y_test, y_prob_rf):.3f}")
```

```
cm_rf = confusion_matrix(y_test, y_pred_rf)
tn_rf, fp_rf, fn_rf, tp_rf = cm_rf.ravel()

print("Confusion Matrix:\n", cm_rf)
print(f"\nAccuracy:                  {accuracy_score(y_test, y_pred_rf):.3f}")
print(f"Sensitivity (Recall):      {recall_score(y_test, y_pred_rf):.3f}")
print(f"Specificity:               {tn_rf / (tn_rf + fp_rf):.3f}")
print(f"Precision (PPV):           {precision_score(y_test, y_pred_rf):.3f}")
print(f"Negative Predictive Value: {tn_rf / (tn_rf + fn_rf):.3f}")
print(f"F1 Score:                  {f1_score(y_test, y_pred_rf):.3f}")
print(f"ROC AUC:                   {roc_auc_score(y_test, y_prob_rf):.3f}")
```

Figure 2.78: Code to generate the Model Evaluation.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.8 ROC Curve

```
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_prob_rf)

auc_rf = roc_auc_score(y_test, y_prob_rf)

plt.figure(figsize=(8, 6))

plt.plot(fpr_rf, tpr_rf, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc_rf:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve – Random Forest (Default Parameters)')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()
```

```
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_prob_rf)
auc_rf = roc_auc_score(y_test, y_prob_rf)

plt.figure(figsize=(8, 6))
plt.plot(fpr_rf, tpr_rf, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc_rf:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve - Random Forest (Default Parameters)')
plt.legend(loc="lower right")
plt.grid(True)
plt.tight_layout()
plt.show()
```

Figure 2.79 Code to generate the ROC Curve.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.9 Confusion Matrix

plt.figure(figsize=(6, 5))

disp = ConfusionMatrixDisplay(confusion_matrix=cm_rf,

display_labels=["Resolved", "Escalated"])

disp.plot(cmap="Blues", values_format='d')

plt.title("Confusion Matrix – Random Forest (Default)")

plt.grid(False)

plt.tight_layout()

plt.show()

```
plt.figure(figsize=(6, 5))
disp = ConfusionMatrixDisplay(confusion_matrix=cm_rf,
                              display_labels=["Resolved", "Escalated"])
disp.plot(cmap="Blues", values_format='d')
plt.title("Confusion Matrix - Random Forest (Default)")
plt.grid(False)
plt.tight_layout()
plt.show()
```

Figure 2.80: Code to generate the Confusion Matrix.
Model: Urine Random Forest. Source: Author's own Python code.

2.8.10 Hyperparameter Optimisation – Grid Search

See section 2.7.10

2.8.11 Hyperparameter Optimisation – Random Search

See section 2.7.11

2.8.11.1 Parameter Space Definition

```
from scipy.stats import randint

param_dist_rf = {

  'n_estimators': randint(100, 600),

  'max_depth': randint(3, 20),

  'min_samples_split': randint(2, 10),

  'min_samples_leaf': randint(1, 10),

  'max_features': ['sqrt', 'log2', None],

  'bootstrap': [True, False]
```

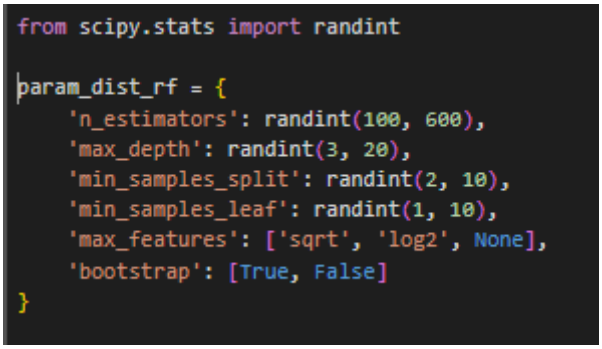


Figure 2.81: Randomised Hyperparameter distribution defined for tuning the Model. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.11.2 Random Search Configuration

```
from sklearn.model_selection import RandomizedSearchCV

from sklearn.ensemble import RandomForestClassifier

rnd_search_rf = RandomizedSearchCV(

  estimator=RandomForestClassifier(random_state=42),

  param_distributions=param_dist_rf,

  n_iter=30,

  scoring='roc_auc',

  cv=5,

  n_jobs=-1,

  random_state=42,

  verbose=2
```

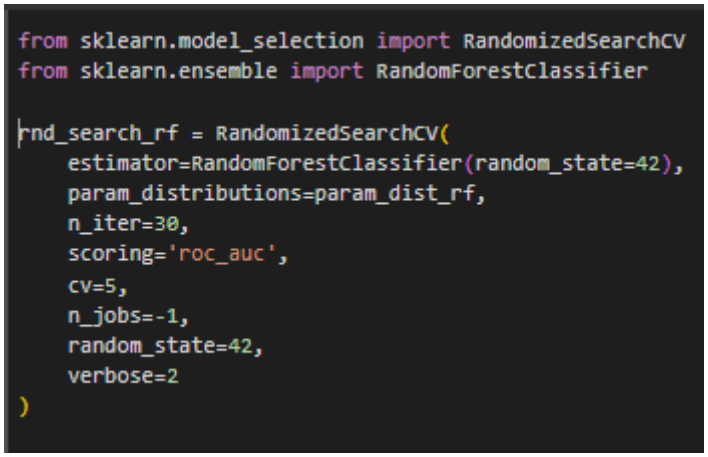


Figure 2.82: Code to RandomizedSearchCV, using 5-fold cross-validation. Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.11.3 Random Search Execution

```
rnd_search_rf.fit(X_train, y_train)
```

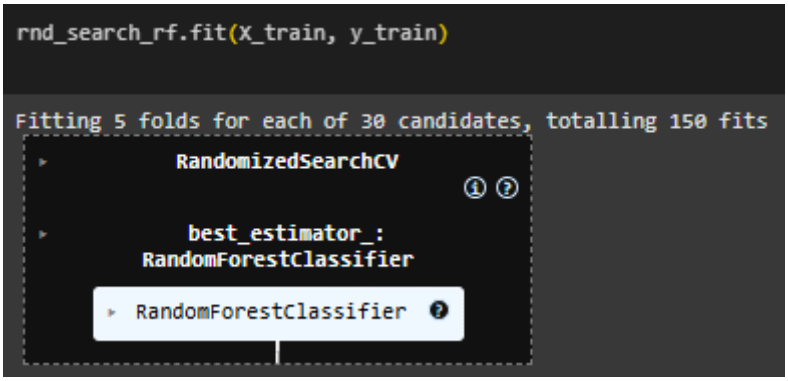


Figure 2.83 Code to fit the training data to RandomizedSearchCV.
Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.11.4 Best-Parameter Extraction

```
print("Best Parameters (Random Forest):", rnd_search_rf.best_params_)

print(f"Best CV AUC: {rnd_search_rf.best_score_:.3f}")
```

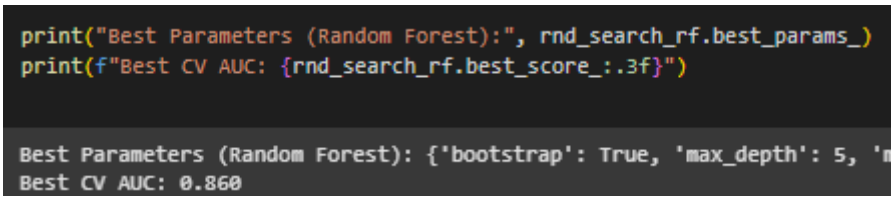


Figure 2.84 Code for retrieval of the best-performing hyperparameters.
Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.12 Evaluation of Tuned Model

```
from sklearn.metrics import (
  confusion_matrix, accuracy_score, precision_score, recall_score,
  f1_score, roc_auc_score

y_pred_best_rf = best_rf.predict(X_test)

y_prob_best_rf = best_rf.predict_proba(X_test)[:, 1]

cm_best_rf = confusion_matrix(y_test, y_pred_best_rf)
```

```
tn_rf, fp_rf, fn_rf, tp_rf = cm_best_rf.ravel()


print("Confusion Matrix:\n", cm_best_rf)

print(f"\nAccuracy:            {accuracy_score(y_test, y_pred_best_rf):.3f}")

print(f"Sensitivity (Recall):    {recall_score(y_test, y_pred_best_rf):.3f}")

print(f"Specificity:          {tn_rf / (tn_rf + fp_rf):.3f}")

print(f"Precision (PPV):       {precision_score(y_test, y_pred_best_rf):.3f}")

print(f"Negative Predictive Value: {tn_rf / (tn_rf + fn_rf):.3f}")

print(f"F1 Score:            {f1_score(y_test, y_pred_best_rf):.3f}")

print(f"ROC AUC:             {roc_auc_score(y_test, y_prob_best_rf):.3f}")
```

```
from sklearn.metrics import (
    confusion_matrix, accuracy_score, precision_score, recall_score,
    f1_score, roc_auc_score
)

y_pred_best_rf = best_rf.predict(X_test)
y_prob_best_rf = best_rf.predict_proba(X_test)[:, 1]

cm_best_rf = confusion_matrix(y_test, y_pred_best_rf)
tn_rf, fp_rf, fn_rf, tp_rf = cm_best_rf.ravel()

print("Confusion Matrix:\n", cm_best_rf)
print(f"\nAccuracy:                  {accuracy_score(y_test, y_pred_best_rf):.3f}")
print(f"Sensitivity (Recall):      {recall_score(y_test, y_pred_best_rf):.3f}")
print(f"Specificity:               {tn_rf / (tn_rf + fp_rf):.3f}")
print(f"Precision (PPV):           {precision_score(y_test, y_pred_best_rf):.3f}")
print(f"Negative Predictive Value: {tn_rf / (tn_rf + fn_rf):.3f}")
print(f"F1 Score:                  {f1_score(y_test, y_pred_best_rf):.3f}")
print(f"ROC AUC:                   {roc_auc_score(y_test, y_prob_best_rf):.3f}")
```

Figure 2.85: Code to generate the Model Evaluation.
Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.13 ROC Curve – Tuned Model

```
from sklearn.metrics import roc_curve, roc_auc_score

import matplotlib.pyplot as plt

fpr_best_rf, tpr_best_rf, _ = roc_curve(y_test, y_prob_best_rf)
```

```
auc_best_rf = roc_auc_score(y_test, y_prob_best_rf)

plt.figure(figsize=(8, 6))

plt.plot(fpr_best_rf, tpr_best_rf, color='darkorange', lw=2,

    label=f'Tuned RF (AUC = {auc_best_rf:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve – Tuned Random Forest')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()
```

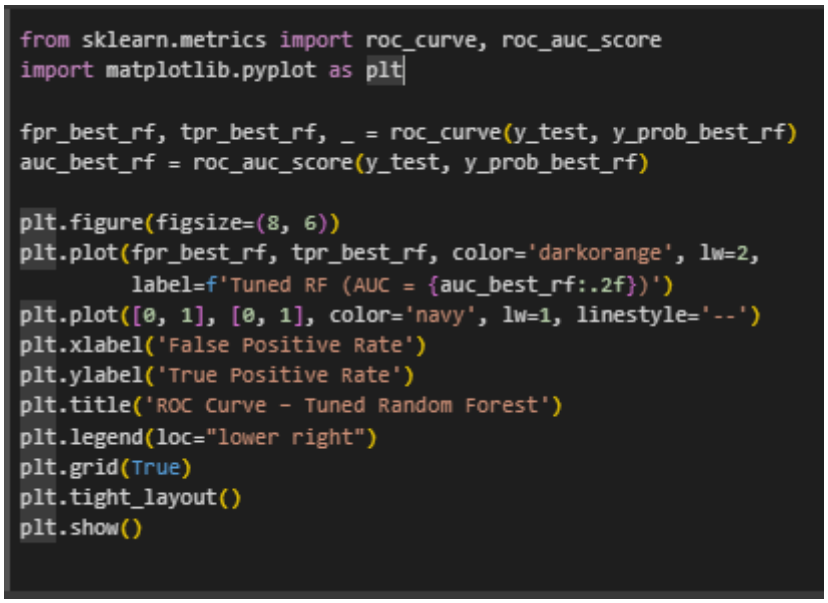


Figure 2.86: Code to generate the ROC Curve.
Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.14 Confusion Matrix – Tuned Model

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm_best_rf = confusion_matrix(y_test, y_pred_best_rf)

plt.figure(figsize=(6, 5))
disp = ConfusionMatrixDisplay(confusion_matrix=cm_best_rf,
                              display_labels=["Resolved", "Escalated"])
disp.plot(cmap="Blues", values_format='d')
plt.title("Confusion Matrix – Tuned Random Forest")
plt.grid(False)
plt.tight_layout()
plt.show()
```

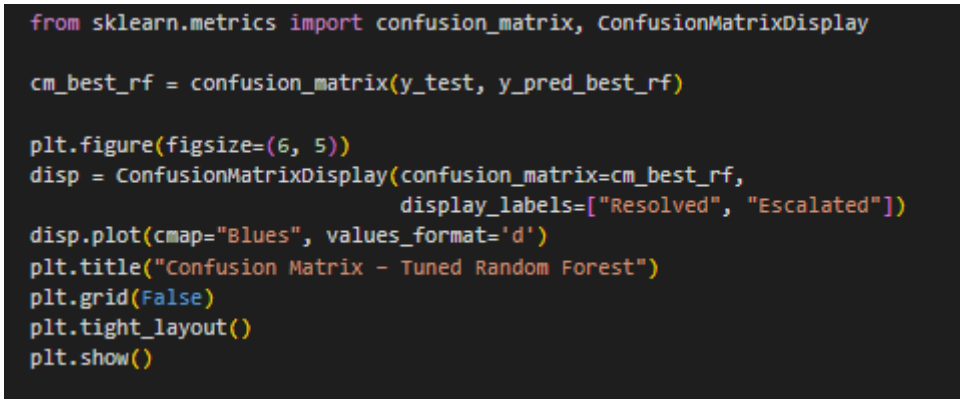


Figure 2.87: Code to generate the Confusion Matrix.
Model: Urine Hyperparameter Random Forest. Source: Author's own Python code.

2.8.15 Comparative ROC Curves

```
y_prob_rf = rf_model.predict_proba(X_test)[:, 1]

fpr_rf, tpr_rf, _ = roc_curve(y_test, y_prob_rf)

auc_rf = roc_auc_score(y_test, y_prob_rf)

plt.figure(figsize=(8, 6))

plt.plot(fpr_rf, tpr_rf, color='steelblue', lw=2, label=f'Default RF (AUC = {auc_rf:.2f})')
```

```
plt.plot(fpr_best_rf, tpr_best_rf, color='darkorange', lw=2, label=f'Tuned RF (AUC =
{auc_best_rf:.2f})')

plt.plot([0, 1], [0, 1], color='grey', lw=1, linestyle='--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve Comparison – Tuned vs. Default RF')

plt.legend(loc="lower right")

plt.grid(True)

plt.tight_layout()

plt.show()
```

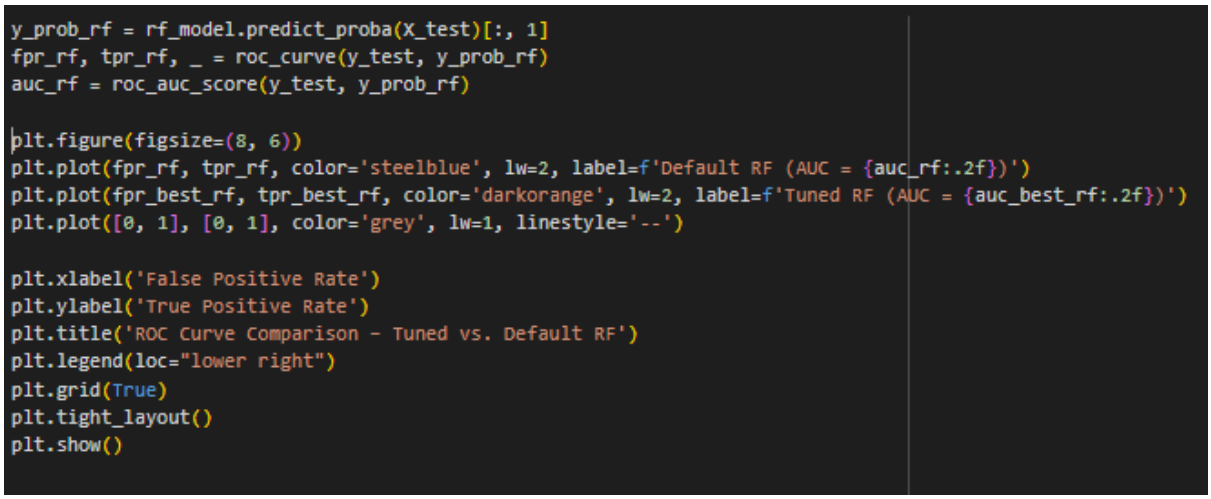


Figure 2.88: Code to generate the untuned and the Hyperparameter tuned ROC Curves. Source: Author's own Python code.

2.8.16 Performance Comparison Table

```
from sklearn.metrics import (

  accuracy_score, precision_score, recall_score,

  f1_score

cm_rf = confusion_matrix(y_test, rf_model.predict(X_test))

tn_d, fp_d, fn_d, tp_d = cm_rf.ravel()

metrics_default = {
```

```
    'Model': 'Default RF',
    'Accuracy': accuracy_score(y_test, rf_model.predict(X_test)),
    'Sensitivity (Recall)': recall_score(y_test, rf_model.predict(X_test)),
    'Specificity': tn_d / (tn_d + fp_d),
    'Precision (PPV)': precision_score(y_test, rf_model.predict(X_test)),
    'NPV': tn_d / (tn_d + fn_d),
    'F1 Score': f1_score(y_test, rf_model.predict(X_test)),
    'ROC AUC': auc_rf,
    'TP': tp_d, 'FP': fp_d, 'FN': fn_d, 'TN': tn_d
tn_t, fp_t, fn_t, tp_t = cm_best_rf.ravel()
metrics_tuned = {
    'Model': 'Tuned RF',
    'Accuracy': accuracy_score(y_test, y_pred_best_rf),
    'Sensitivity (Recall)': recall_score(y_test, y_pred_best_rf),
    'Specificity': tn_t / (tn_t + fp_t),
    'Precision (PPV)': precision_score(y_test, y_pred_best_rf),
    'NPV': tn_t / (tn_t + fn_t),
    'F1 Score': f1_score(y_test, y_pred_best_rf),
    'ROC AUC': auc_best_rf,
    'TP': tp_t, 'FP': fp_t, 'FN': fn_t, 'TN': tn_t
import pandas as pd
rf_comparison_df = pd.DataFrame([metrics_default, metrics_tuned])
print(rf_comparison_df.round(3))
```

```python
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score,
    f1_score
)

cm_rf = confusion_matrix(y_test, rf_model.predict(X_test))
tn_d, fp_d, fn_d, tp_d = cm_rf.ravel()
metrics_default = {
    'Model': 'Default RF',
    'Accuracy': accuracy_score(y_test, rf_model.predict(X_test)),
    'Sensitivity (Recall)': recall_score(y_test, rf_model.predict(X_test)),
    'Specificity': tn_d / (tn_d + fp_d),
    'Precision (PPV)': precision_score(y_test, rf_model.predict(X_test)),
    'NPV': tn_d / (tn_d + fn_d),
    'F1 Score': f1_score(y_test, rf_model.predict(X_test)),
    'ROC AUC': auc_rf,
    'TP': tp_d, 'FP': fp_d, 'FN': fn_d, 'TN': tn_d
}

tn_t, fp_t, fn_t, tp_t = cm_best_rf.ravel()
metrics_tuned = {
    'Model': 'Tuned RF',
    'Accuracy': accuracy_score(y_test, y_pred_best_rf),
    'Sensitivity (Recall)': recall_score(y_test, y_pred_best_rf),
    'Specificity': tn_t / (tn_t + fp_t),
    'Precision (PPV)': precision_score(y_test, y_pred_best_rf),
    'NPV': tn_t / (tn_t + fn_t),
    'F1 Score': f1_score(y_test, y_pred_best_rf),
    'ROC AUC': auc_best_rf,
    'TP': tp_t, 'FP': fp_t, 'FN': fn_t, 'TN': tn_t
}

import pandas as pd
rf_comparison_df = pd.DataFrame([metrics_default, metrics_tuned])
print(rf_comparison_df.round(3))
```

Figure 2.89: Code to generate an untuned vs Hyperparameter tuned comparison table. Source: Author's own Python code.

2.9 Technical Notes and Reproducibility

2.9.1 Programming Languages and Libraries

- Platform: Google Colaboratory ("Colab") running Python 3.10 kernels.
- Secondary/local check: Single-run grid-search replicated on a 2025 Apple MacBook Pro (Python 3.12, scikit-learn 1.5) solely to verify Colab results; final analyses remain Colab-based for full reproducibility.

2.9.2 Programming Languages and Scripting

- Python (all modelling, visualisation, and statistical evaluation).
- Excel (Microsoft 365) for first-pass feature engineering and for the clinician-facing logistic-regression calculator template.

2.9.3 Python libraries utilised

Category	Library	Purpose / Typical Calls (examples)
Data handling	pandas	read_csv, DataFrame wrangling
Numerics	numpy	coefficient math, exponentiation, array ops
Core ML	scikit-learn	LogisticRegression, RandomForestClassifier, GradientBoostingClassifier train_test_split, StandardScaler, cross_val_score GridSearchCV, RandomizedSearchCV for hyper-parameter tuning
Gradient boosting (ext.)	xgboost (XGBClassifier)	Non-linear GBM modelling and tree-feature importance
Visualisation	matplotlib.pyplot	ROC curves, confusion matrices, feature bars
Visualisation (aux)	seaborn	Optional aesthetic plots/heat-maps
Probability dists	scipy.stats (randint, uniform)	Random-search hyper-parameter sampling

Table 2.2: Summary of Python libraries utilised during Chapter 2. Source: Compiled by Author manually.

2.9.4 Machine-learning models implemented

1. Logistic Regression – linear classifier with coefficient-based interpretation and feature selection.
2. Gradient Boosting
    - scikit-learn GradientBoostingClassifier (urine & serum datasets).
    - XGBoost XGBClassifier with tuned depth/learning-rate/trees (serum dataset).
3. Random Forest – default and tuned (GridSearchCV → RandomizedSearchCV) implementations.

2.9.5 Pre-processing & Feature Engineering

- StandardScaler applied to all coefficient- or distance-based models; scaling parameters fitted on training folds then reused on held-out data to avoid leakage.
- No scaling for tree-based models (RF, GBM, XGB) as splitting is threshold-based.
- Manual feature creation (rates, slopes, percentage drops, curvature flags, timings) first drafted in Excel then validated in Python.

2.9.6 Hyperparameter Optimisation & Validation

- GridSearchCV (exhaustive) where computationally feasible; switched to RandomizedSearchCV when grid size × Colab time-outs became prohibitive, substantially reducing computational time while maintaining comparable model performance.
- 5-fold cross-validation (stratified where class imbalance noted) for all tuning and for mean AUC reporting.
- cross_val_score used for out-of-fold AUC confirmation on best logistic-regression variant.

2.9.7 Performance Metrics Reported

Accuracy, ROC-AUC, Precision (PPV), Recall/Sensitivity, Specificity, F1, Negative Predictive Value, plus visual ROC and confusion-matrix plots for every final model.

2.9.8 Reproducibility Measures

- Single project-wide random_state = 42 applied to every split, model, and search object.
- All Colab notebooks retain code / outputs; external Mac test used identical seeds to confirm tuning plateau.

2.9.9 Ethical & Governance Controls

All datasets used in this study were fully anonymised prior to analysis, with no patient-identifiable variables included in any modelling pipelines. The data were collected under ethical approval granted by Sheffield Hallam University (SHU) (IRAS project ID: 227556; Protocol number: STH19535; REC reference: 19/HRA/0326), and the analyses presented here represent a secondary analytical use of these data within the scope of that approval.

All data handling and analysis were conducted in accordance with the principles of the Declaration of Helsinki (World Medical Association, 2013) and relevant data protection legislation, including the UK General Data Protection Regulation (UK Parliament, 2018). Analytical work was performed within a secure, private Google Colab environment to minimise the risk of unintended data exposure or unauthorised sharing.

3.0 Results Chapter

Although the study's Hypotheses were defined by cohort, **Hypotheses 1 and 2** addressing the urine dataset (untreated patients under surveillance) and **Hypotheses 3 and 4** addressing the serum dataset (patients undergoing methotrexate treatment), the Results chapter presents the serum models first. This order reflects the practical realities of data availability and analysis timing, rather than the numerical order of Hypotheses. Specifically, the serum data was cleaned, validated, and model-ready earlier in the project timeline, allowing for immediate progression with model development, performance evaluation, and result interpretation. In contrast, the urine dataset required additional time for cohort refinement and feature extraction. As such, the write-up of serum modelling outcomes naturally commenced first, forming the initial structure of the Results chapter. While this diverges from a strictly Hypothesis-led format, it reflects a data-driven and chronologically accurate narrative, aligned with the actual workflow of the project. Each section clearly maps findings to the appropriate Hypotheses, and both datasets receive equal methodological rigour and analytical depth. This structure also supports the logical progression of the Results chapter, building from initial modelling efforts (serum) to broader applications in a more heterogeneous untreated population (urine). Ultimately, the ordering enhances readability while maintaining scientific integrity, ensuring that findings are presented in a manner consistent with the research process and transparent for the reader.

3.1 Introduction

This chapter presents the complete findings from machine learning analyses conducted on two distinct cohorts of patients undergoing post-molar surveillance:

- A serum cohort (n = 332), comprising patients who underwent chemotherapy with methotrexate (MTX), with outcomes of either cure or escalation to Actinomycin D (ActD).
- A urine cohort (n = 4850), comprising patients who were monitored without treatment. Outcomes reflected spontaneous hCG resolution or subsequent requirement for treatment.

Twelve models were developed and evaluated across these cohorts. Metrics include accuracy, precision (PPV), recall (sensitivity), specificity, F1 score, negative predictive value (NPV), and area under the ROC curve (AUC). Confusion matrices and ROC curves were generated for each model.

3.1.1 Model Evaluation Metrics

To assess the predictive performance of each machine learning model developed in this study, a comprehensive suite of evaluation metrics was employed. These metrics were selected based on their interpretability, clinical relevance, and capacity to reflect the trade-offs between sensitivity and specificity that are critical in stratifying patients with GTD for treatment escalation (Akobeng, 2007). The performance of all models was evaluated using a confusion matrix framework, which classifies each prediction into one of four outcome categories:

- True Positives (TP): Patients correctly predicted to experience the positive outcome class (i.e. treatment requirement or escalation depending on the model cohort).
- True Negatives (TN): Patients correctly predicted to experience the negative outcome class (i.e. spontaneous resolution or successful treatment without escalation depending on the model cohort).
- False Positives (FP): Patients incorrectly predicted to experience the positive outcome class (i.e. predicted treatment requirement or escalation when this outcome did not occur).
- False Negatives (FN): Patients incorrectly predicted to experience the negative outcome class (i.e. predicted spontaneous resolution or treatment success when treatment requirement or escalation subsequently occurred).

From these foundational components, the following metrics were derived and are discussed in detail below.

3.1.1.1 Accuracy

Accuracy measures the proportion of all correct predictions (both true positives and true negatives) among the total number of cases examined. It is calculated as (TP + TN) / (TP + TN + FP + FN). While accuracy provides an overall measure of model performance, it can be misleading in datasets with class imbalance, such as in GTD, where most patients may not require treatment escalation (Saito and Rehmsmeier, 2015). In such cases, a model could achieve high accuracy by simply predicting the majority class, thus failing to identify patients who do require intervention. Therefore, although reported, accuracy was not relied upon as the sole indicator of model quality.

3.1.1.2 Sensitivity (Recall)

Sensitivity, also known as recall or the true positive rate, measures the proportion of actual positives (patients requiring escalation) that are correctly identified by the model. It is calculated as TP / (TP + FN). In the context of GTD, high sensitivity is crucial to ensure that patients who require treatment escalation are not missed. A model with high sensitivity minimises false negatives, thereby reducing the risk of delayed treatment and potential disease progression. Given the serious consequences of failing to identify patients needing treatment, sensitivity is a critical metric for evaluating model performance in this clinical setting (Akobeng, 2007: Mandrekar, 2010).

3.1.1.3 Specificity

Specificity measures the proportion of actual negatives (patients not requiring escalation) that are correctly identified by the model. It is calculated as TN / (TN + FP). High specificity is important to avoid unnecessary treatment in patients who would otherwise resolve naturally. This reduces patient anxiety, clinical burden, and potential toxicity from overtreatment. Balancing sensitivity and specificity is essential to develop a model that accurately identifies patients needing treatment while minimising unnecessary interventions (Altman and Bland, 1994: Akobeng, 2007).

3.1.1.4 Precision (Positive Predictive Value)

Precision, or positive predictive value (PPV), indicates the proportion of predicted positive cases that are true positives. It is calculated as TP / (TP + FP). High precision implies that most patients identified by the model as requiring treatment truly do need it, supporting clinical confidence in acting on such predictions. Precision is particularly important when the cost of false positives is high, as it reflects the model's reliability in predicting escalation (Goutte and Gaussier, 2005: Powers, 2011).

3.1.1.5 Negative Predictive Value

Negative Predictive Value (NPV) indicates the proportion of predicted negative cases that are true negatives. It is calculated as TN / (TN + FN). In clinical monitoring, NPV reflects the reassurance provided by a model prediction of no treatment need. High NPV means that when the model predicts natural resolution, it is highly likely to be correct, enabling more conservative management strategies. NPV is crucial for assessing the model's ability to correctly identify

patients who do not require treatment, thereby avoiding unnecessary interventions (Akobeng, 2007: Salmi et al., 2024).

3.1.1.6 F1 Score

The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both false positives and false negatives. It is calculated as 2 × (Precision × Recall) / (Precision + Recall). The F1 score is particularly useful when there is an uneven class distribution and when both sensitivity and precision are important. It penalises extreme imbalances and provides a holistic view of model utility, especially in scenarios with clinical uncertainty. In the context of GTD, where both false positives and false negatives carry significant consequences, the F1 score offers a balanced measure of model performance (Saito and Rehmsmeier, 2015: Goutte and Gaussier, 2005).

3.1.1.7 Area Under the Curve – Receiver Operating Characteristic:

The Area Under the Curve (AUC) quantifies the area under the Receiver Operating Characteristic (ROC) curve, which plots sensitivity (true positive rate) against 1-specificity (false positive rate) across all possible classification thresholds. AUC provides an aggregate measure of model discrimination regardless of any specific threshold (Hosmer et al., 2013: Mandrekar, 2010). In this study, it was employed as a primary measure of model quality, enabling comparison between models in terms of their ability to distinguish between patients requiring ActD escalation following MTX and those successfully treated with MTX, and between untreated patients demonstrating natural hCG resolution and those requiring treatment. An AUC close to 1.0 indicates excellent discrimination, while a value of 0.5 suggests no better performance than random chance.

These evaluation metrics collectively provide a multidimensional perspective on model performance. Given the potential consequences of both over- and under-treatment in GTD management, no single metric was deemed sufficient. Instead, a suite of metrics was employed to ensure that models could be robustly assessed for both clinical safety and operational efficiency. Metrics were interpreted in the context of model purpose: prioritising sensitivity and NPV where minimising missed escalation is critical, and considering specificity and PPV to reduce unnecessary treatment in self-resolving cases. This balanced evaluation strategy allowed for appropriate alignment of model characteristics with the clinical objectives outlined in **Hypotheses 1 and 2**. Importantly, these metrics also support the more complex clinical

predictions addressed in **Hypotheses 3 and 4**. These Hypotheses aim to identify patients who are at risk of demonstrating inadequate response to methotrexate and require escalation to Actinomycin D. Here, the balance between recall and precision becomes more nuanced, with the F1 score providing a holistic measure of model performance under conditions where both false positives and false negatives are consequential (Goutte and Gaussier, 2005). The area under the receiver operating characteristic curve (AUC-ROC) was used throughout as a threshold-independent metric of model discrimination, enabling comparison across models and informing selection of optimal classification thresholds for implementation in clinical decision-making (Hosmer et al., 2013: Mandrekar, 2010). This balanced evaluation strategy ensured that model characteristics were appropriately aligned with the distinct clinical objectives outlined in **Hypotheses 1 through 4**.

3.2 Serum Dataset Modelling (MTX-Treated Patients)

3.2.1 Dataset Summary

This cohort consisted of 332 anonymised patients who underwent MTX treatment for GTD, all of whom had complete serial serum hCG measurements from Day 0 to Day 28. The dataset was divided into two outcome groups: those who achieved remission with MTX alone (MTX Cure, n = 187), and those who required escalation to Actinomycin D due to MTX resistance (ActD Escalation, n = 145). Distinguishing between these groups is clinically significant, as early identification of MTX resistance enables timely escalation to second-line therapy, reducing exposure to ineffective treatment and minimising delays in disease control. This serum cohort forms the basis for the evaluation of **Hypotheses 3 and 4:**

- **Hypothesis 3**: Patients demonstrating inadequate response to MTX exhibit distinct hCG trajectories that can predict requirement for ActD.
- **Hypothesis 4**: Predictive modelling of early serum hCG kinetics can support the development of robust decision tools to anticipate MTX resistance, thereby facilitating timely escalation and minimising ineffective treatment cycles.

This dataset enables comparative modelling of treatment responders and non-responders, with the aim of identifying discriminative early hCG patterns and improving therapeutic stratification. Serum hCG values were processed using derived features capturing early kinetic changes (e.g., percentage decline and slope-based measures), ensuring that model inputs reflected clinically relevant trends rather than isolated measurements.

3.2.2 Logistic Regression (12-Feature Model)

3.2.2.1 Default Hyperparameters:

- Solver = 'lbfgs';
- Regularisation = L2;
- C = 1.0;
- max_iter = 1000;
- class_weight = None

Logistic regression is a parametric statistical model commonly employed for binary classification problems. It predicts the probability of a binary outcome using the logistic (sigmoid) function to transform a linear combination of input features into a probability score between 0 and 1 (Hosmer et al., 2013). In the context of supervised machine learning, logistic regression is treated as a discriminative model optimised using gradient-based techniques to minimise a penalised loss function. In practical implementation, particularly within the scikit-learn library (Pedregosa et al., 2011), the behaviour of logistic regression is governed by several hyperparameters. These are not learned from the data but must be set before model training. The choice of hyperparameters can critically influence the model's ability to generalise to unseen data and its numerical stability during optimisation.

The solver parameter specifies the optimisation algorithm used to minimise the loss function. In this model, the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm is used. L-BFGS is a quasi-Newton method that approximates the Hessian matrix to iteratively update the model's weights (Liu and Nocedal, 1989). It is particularly well-suited for convex optimisation problems such as logistic regression, offering fast convergence and good performance for datasets of moderate size and dimensionality, making it suitable for this dataset. Compared to alternatives such as liblinear or sag, L-BFGS handles multiclass problems (via one-vs-rest or multinomial approaches) and supports L2 regularisation efficiently. Its memory-efficient implementation makes it appropriate even when the number of features is not trivially small (Defazio et al, 2014).

Regularisation combats over-fitting in statistical models by discouraging the algorithm from assigning excessively large weights to any single predictor. In logistic regression, L2 (ridge) regularisation achieves this by adding a penalty that grows with the square of each coefficient's magnitude. The practical effect is a gentle, continuous "shrinkage" of all coefficients toward

zero. Because no weight is ever forced exactly to zero, every predictor remains in the model, yet extreme values are suppressed, therefore again making this suitable for the datasets (Ng, 2004). The C parameter is the inverse of the regularisation strength. Therefore, smaller values of C correspond to stronger regularisation, whereas larger values permit the model to fit the training data more closely. A C value of 1.0 represents moderate regularisation, balancing model complexity against performance on the training set. Empirical evidence suggests that values near 1.0 often perform well in practice (Tibshirani, 1996).

The max_iter = 1000 parameter defines the maximum number of iterations for the solver to converge. Optimisation algorithms such as L-BFGS are iterative, and depending on the data and regularisation settings, may require many steps to reach a minimum of the loss function. Setting max_iter = 1000 ensures sufficient opportunity for convergence, particularly in high-dimensional feature spaces or where data may not be well-scaled. If convergence is not reached within this limit, a warning is raised, and model parameters may not be optimal (Zhang et al., 2004).

The class_weight = None parameter controls whether the model compensates for class imbalance by adjusting the weights of classes in the loss function. With class_weight = None, no weighting is applied, and all instances are treated equally. This is appropriate when the class distribution is balanced, or when evaluation metrics, such as AUC, already account for imbalance. In contrast, class_weight = 'balanced' would adjust weights inversely proportional to class frequencies, which is useful when minority classes are underrepresented (King and Zeng, 2001).

These hyperparameters define both the numerical behaviour and the generalisation capacity of the logistic regression model. Together, they facilitate a convex optimisation approach that is efficient, interpretable, and appropriate for the modelling task at hand.

3.2.2.2 Performance:

- Accuracy = 0.746
- AUC = 0.825
- Precision = 0.714
- Recall = 0.577
- F1 Score = 0.638
- NPV = 0.761

The performance of the reduced logistic regression model, trained on serum hCG data from Day 0 to Day 28, was evaluated to determine its ability to discriminate between patients who responded successfully to methotrexate (MTX Cure) and those who required escalation to Actinomycin D (Ac D Escalated). This model aimed to support **Hypotheses 3 and 4** by identifying early hCG kinetics indicative of inadequate MTX response. As shown in the confusion matrix (Figure 3.2.2.3), the model correctly classified 35 MTX-responsive patients and 15 patients who required escalation. However, it misclassified 6 true MTX cures as requiring ActD (false positives) and failed to identify 11 escalation cases (false negatives), highlighting the model's limitations in sensitivity.

The model achieved an overall accuracy of 74.6%, indicating that nearly three-quarters of predictions were correct. However, given the near-equal distribution of escalation versus cure cases in this subset, accuracy alone is not a sufficient marker of model quality (Altman and Bland, 1994). The ROC curve, shown in Figure 3.2.2.4, provides a more comprehensive view of the model's discriminative ability by plotting the trade-off between true positive rate (sensitivity) and false positive rate across classification thresholds. The resulting AUC of 0.825 demonstrates good overall performance, suggesting the model is capable of distinguishing between MTX-responsive and MTX-resistant patients with high probability (Hosmer et al., 2013; Mandrekar, 2010). Notably, the mean cross-validated AUC was 0.780, supporting the model's generalisability and indicating that performance was consistently maintained across multiple training–test splits with minimal degradation. This strengthens confidence in the model's robustness and potential clinical application.

Despite strong AUC performance, the model's recall (sensitivity) was 0.577, meaning it correctly identified approximately 58% of escalation cases. This moderate sensitivity limits the model's ability to fully satisfy **Hypothesis 3**, which requires robust detection of patients demonstrating inadequate MTX response. Conversely, the precision of 0.714 reflects that when the model predicts escalation, it is correct over 70% of the time. This supports safe, early escalation decisions and aligns with the intent of **Hypothesis 4**, to develop predictive tools that minimise ineffective treatment cycles (Saito and Rehmsmeier, 2015). The F1 score of 0.638, which balances both precision and recall, underscores this trade-off and indicates moderate but clinically useful performance.

Of particular importance in clinical decision-making is the negative predictive value (NPV = 0.761). This suggests that when the model predicts MTX success, it is correct over 76% of the time, providing reasonable reassurance for continued single-agent therapy. Taken together, these results indicate that while the reduced logistic regression model does not capture all escalation cases, it offers useful discriminatory power with good precision and AUC. It supports **Hypothesis 4** by contributing to the development of early stratification tools, but highlights the need for either additional features or alternative modelling strategies to improve sensitivity and fulfil the requirements of **Hypothesis 3**.

3.2.2.3 Confusion Matrix:

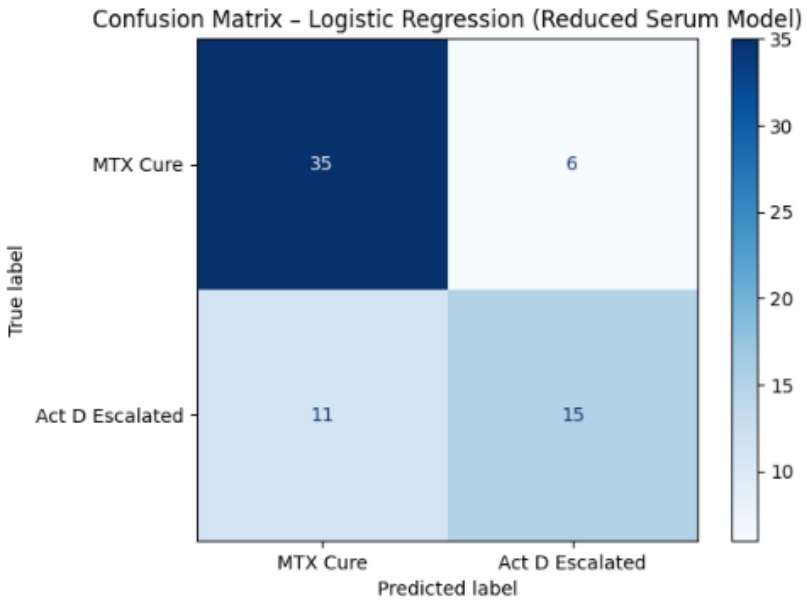


Figure 3.1: Confusion Matrix.
Model: Serum Logistic Regression. Source: Generated in Python using Author's own code.

3.2.2.4 ROC Curve:

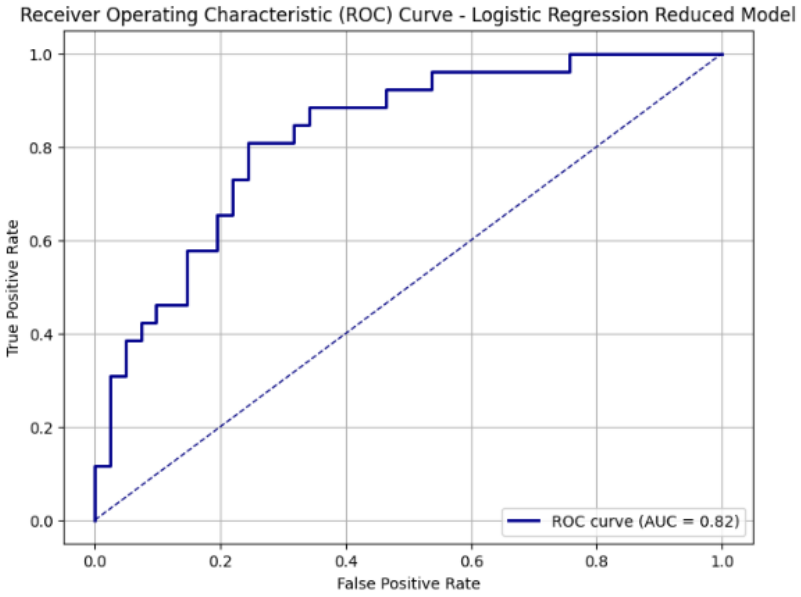


Figure 3.2: ROC Curve.
Model: Serum Logistic Regression. Source: Generated in Python using Author's own code.

3.2.2.5 Coefficient Importance:

Day0_hCG: 0.3030719093056573

Day14_hCG: -0.005680410034555387

Day28_hCG: 0.18515236866351598

Slope_0_14: -0.5906139618369929

Slope_14_28: 0.22993133637510046

Slope_0_28: -0.30320531817651075

Percent_Drop_0_14: -0.05748056565529501

Percent_Drop_14_28: -0.5206784016686724

Percent_Drop_0_28: -0.48348743179509385

Ratio_0_to_14: 0.05040141371830295

Ratio_0_to_28: -0.46355955839329926

Rebound_14_28: 0.49616567257747024

Min_hCG_0_28: 0.2017047215951425

3.2.2.6 Cross validation:

Mean Cross-Validated AUC: 0.780

3.2.3 Logistic Regression (Top 5 Feature Model)

3.2.3.1 Features used:

- Slope_0_14,
- Percent_Drop_14_28,
- Rebound_14_28,
- Percent_Drop_0_28,
- Ratio_0_to_28

This reduced logistic regression model was trained using only the five most important serum hCG-derived features to classify patients as either MTX Cure or requiring escalation to Actinomycin D. The goal was to determine whether a minimal feature set could retain meaningful predictive value for treatment stratification under **Hypotheses 3 and 4**

3.2.3.2 Performance:

- Accuracy: 0.642
- AUC: 0.779
- Precision: 0.526
- Recall: 0.769
- F1 Score: 0.625
- NPV: 0.793

The confusion matrix for this model (Figure 3.2.3.3) shows that 34 of the 41 MTX responders were correctly identified, alongside 15 of the 26 escalation cases. However, 11 patients requiring escalation were misclassified as MTX Cure (false negatives), and 7 true MTX cures were incorrectly predicted to escalate (false positives). These errors reflect the limitations of reducing the feature space but also suggest the model retained a reasonable capacity for clinical discrimination.

The model achieved an accuracy of 64.2%, reflecting moderate overall correctness. More notably, the AUC-ROC was 0.779 (Figure 3.2.3.4), indicating acceptable discrimination between response and escalation classes, albeit slightly reduced compared to the full logistic regression model. This AUC still supports the clinical utility of the top five features in capturing key treatment trajectory signals (Hosmer et al., 2013: Mandrekar, 2010).

The model's recall (0.769) indicates that nearly 77% of escalation cases were correctly identified, an improvement in sensitivity compared to the full model. This enhances its alignment with **Hypothesis 3**, which focuses on early identification of inadequate MTX response requiring treatment escalation. However, this sensitivity comes at the expense of precision (0.526), meaning nearly half of all predicted escalation cases were false positives. While this may result in some unnecessary concern or early intervention, it also demonstrates the model's caution in avoiding missed escalation.

The F1 score (0.625) reflects this precision-recall trade-off, suggesting moderate balance in classifying complex cases. The negative predictive value (NPV = 0.793) further indicates that patients predicted to be MTX responders had nearly an 80% chance of genuinely responding, which supports the use of the model for safe continuation of single-agent therapy in lower-risk cases, aligning with **Hypothesis 4**.

Overall, this five-feature model provides reasonable sensitivity and robust NPV with fewer input variables, maintaining clinical relevance and interpretability. It demonstrates that a carefully selected feature subset can sustain predictive performance in a streamlined model, although at the cost of reduced precision. This supports its potential role in real-time triage or integration into decision-support tools where simplicity and speed are prioritised.

3.2.3.3 Confusion Matrix:

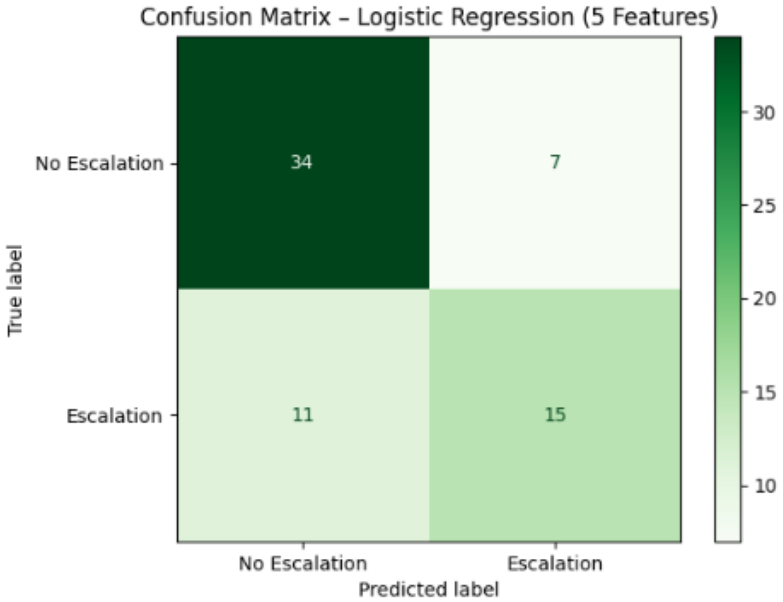


Figure 3.3: Confusion Matrix.
Model: Serum Logistic Regression (5 Feature). Source: Generated in Python using Author's own code.

3.2.3.4 ROC Curve:

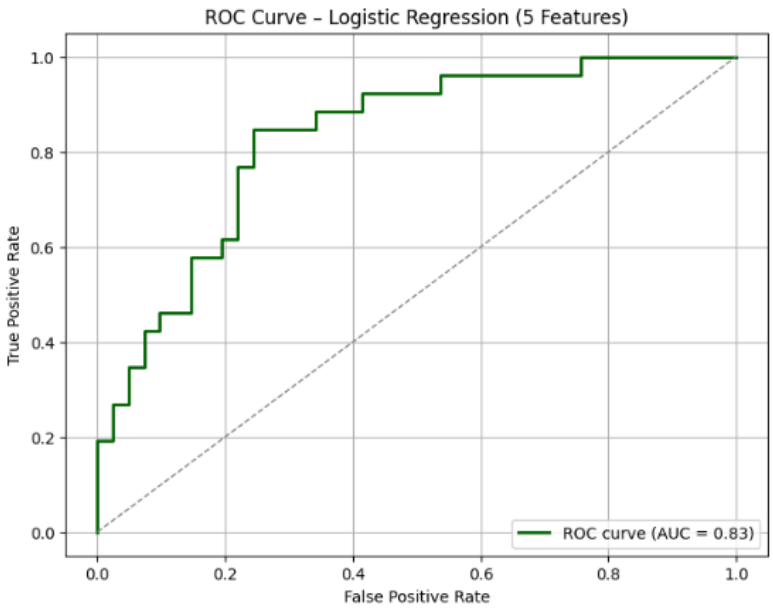


Figure 3.4: ROC Curve.
Model: Serum Logistic Regression (5 Feature). Source: Generated in Python using Author's own code.

3.2.3.5 Cross validation:

Mean Cross-Validated AUC: 0.789

3.2.4 Logistic Regression (Top 5 + Day0_hCG Model)

3.2.4.1 Features used:

- Slope_0_14,
- Percent_Drop_14_28,
- Rebound_14_28,
- Percent_Drop_0_28,
- Ratio_0_to_28
- Day0_hCG

This model builds upon the reduced logistic regression approach by including the initial hCG value (Day 0 hCG) alongside the five most important derived features. The inclusion of baseline hCG was intended to enhance model discrimination while maintaining parsimony.

3.2.4.2 Performance:

- Accuracy: 0.746
- AUC: 0.837
- Precision: 0.714
- Recall: 0.577
- F1 Score: 0.638
- NPV: 0.761

As shown in the confusion matrix (Figure 3.2.4.3), this model correctly identified 35 patients who responded to MTX and 15 patients who required ActD escalation. Misclassifications included 6 false positives (MTX cures incorrectly predicted to escalate) and 11 false negatives (escalation cases predicted to cure). These figures mirror the results of the full 12-feature logistic regression model, indicating that the additional features excluded here did not meaningfully enhance classification.

The model achieved an accuracy of 74.6%, suggesting strong overall performance with nearly three-quarters of cases correctly classified. The AUC-ROC of 0.837 (Figure 3.2.4.4) reflects strong discriminative capability between MTX responders and escalation cases. Importantly, the mean cross-validated AUC of 0.791 supports the model's reliability across different data splits, demonstrating only a modest drop from the test-set AUC. This consistency affirms the model's

robustness and generalisability, particularly when applied in broader clinical settings (Hosmer et al., 2013; Mandrekar, 2010).

The precision of 0.714 indicates that when escalation was predicted, the model was correct over 70% of the time, providing reasonable confidence in initiating second-line therapy. However, the recall was lower at 0.577, identifying just under 58% of escalation cases. This trade-off is reflected in the F1 score of 0.638, which shows moderate balance between avoiding false negatives and limiting unnecessary escalation predictions.

The NPV of 0.761 implies that patients predicted to succeed on MTX were truly responsive in over 76% of cases, a useful characteristic for conservative treatment strategies. This supports the model's relevance to **Hypothesis 4**, by offering reassurance in continued single-agent therapy. However, the moderate sensitivity still limits its ability to fully satisfy **Hypothesis 3**, which aims to detect patients demonstrating inadequate MTX response as early as possible.

In summary, the addition of baseline Day 0 hCG to the top five derived features modestly improves AUC and generalisation without altering the overall classification pattern. The strong test and cross-validated performance metrics reinforce this model's clinical potential, offering an optimal balance between simplicity and predictive value for early treatment stratification in GTD.

3.2.4.3 Confusion Matrix:

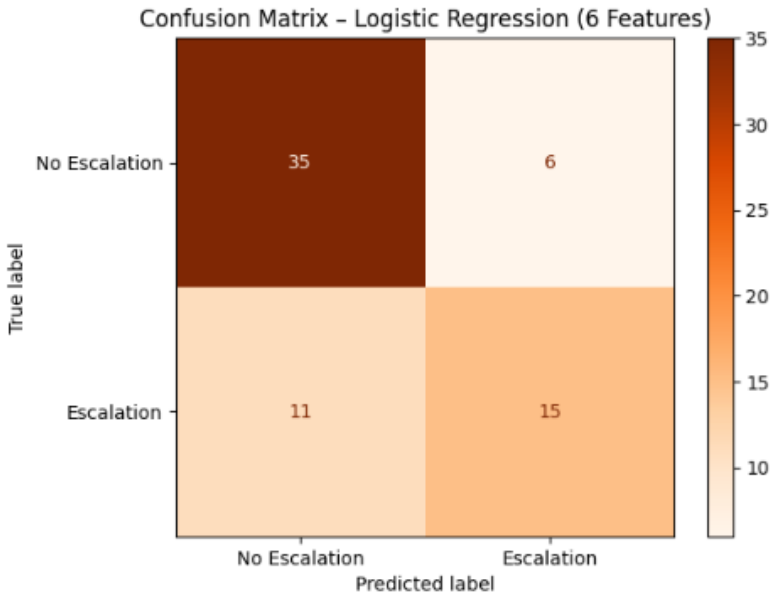


Figure 3.5: Confusion Matrix.

Model: Serum Logistic Regression (6 Feature). Source: Generated in Python using Author's own code.

3.2.4.4 ROC Curve:

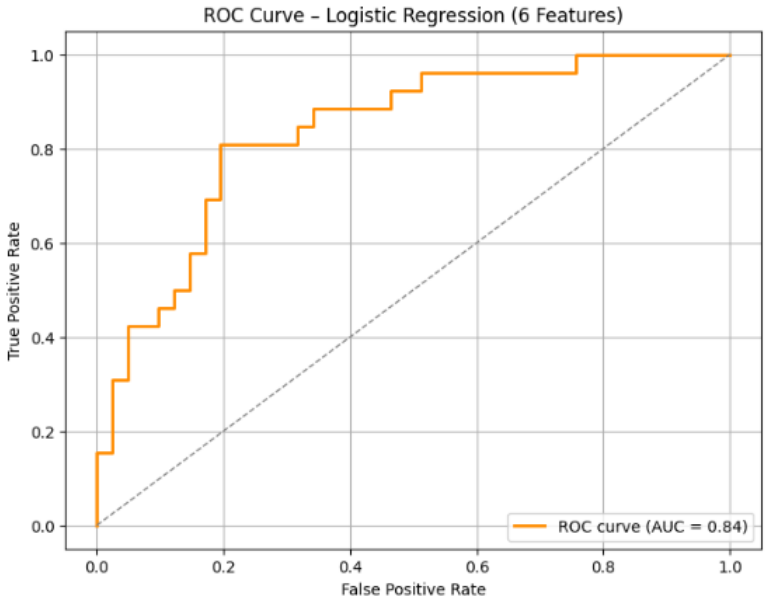


Figure 3.6: ROC Curve.
Model: Serum Logistic Regression (6 Feature). Source: Generated in Python using Author's own code.

3.2.4.5 Cross validation:

Mean Cross-Validated AUC: 0.791

3.2.5 Logistic Regression Model Comparison

3.2.5.1 Performance:

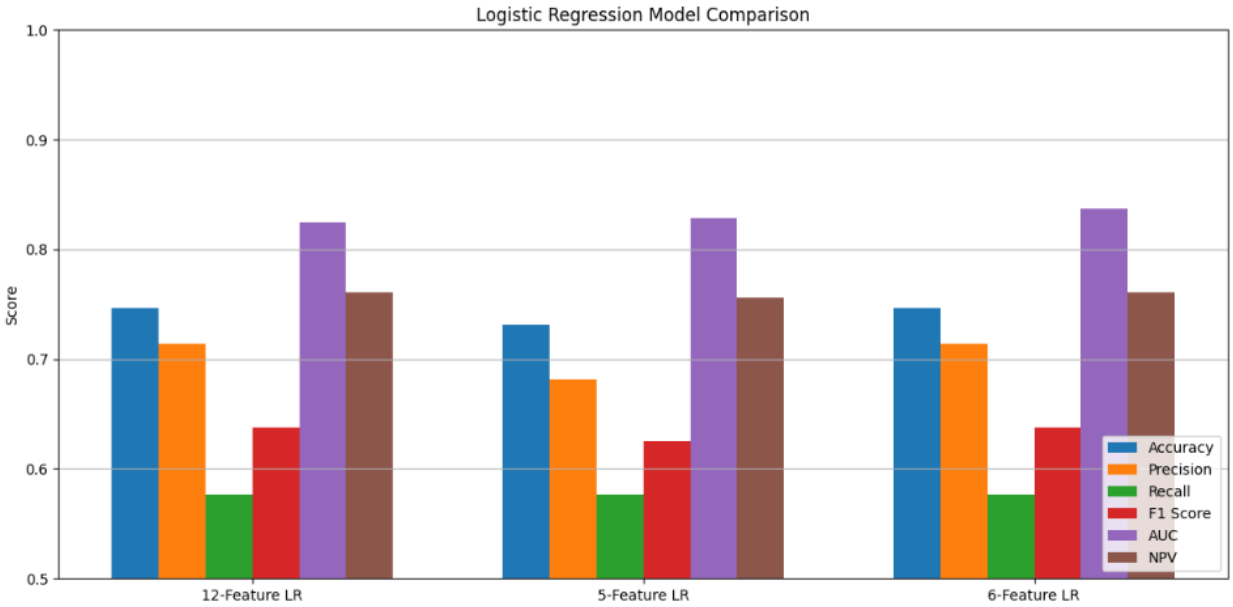


Figure 3.7: Performance Graph highlighting key metric comparison between Serum Logistic Regression Models. Source: Generated in Python using Author's own code.

3.2.5.2 ROC Curve:

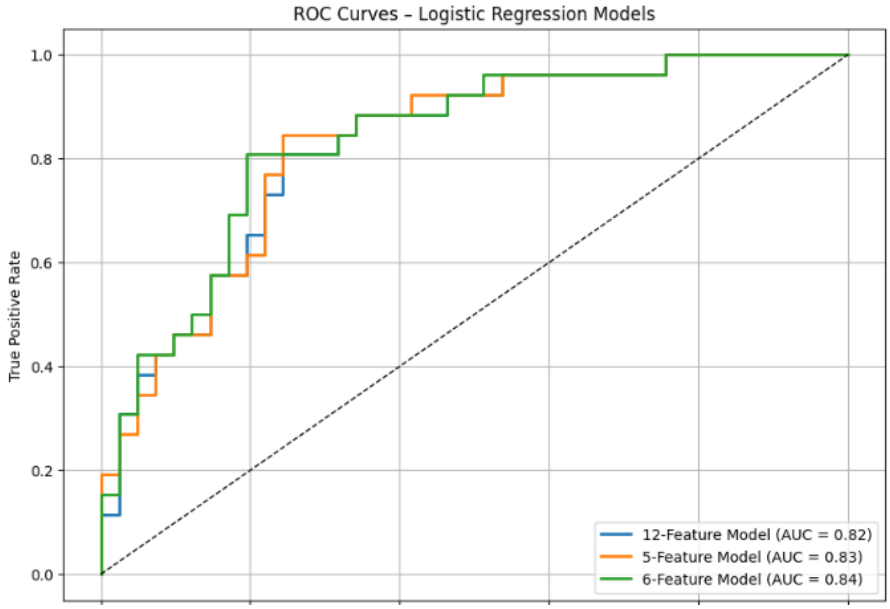


Figure 3.8: ROC comparison between Serum Logistic Regression Models. Source: Generated in Python using Author's own code.

3.2.5.3 Summary of Logistic Regression Models (Serum Dataset)

Three logistic regression models were developed and compared using varying feature sets derived from serial serum hCG measurements between Day 0 and Day 28. The objective was to evaluate their ability to distinguish between patients who would respond to MTX Cure and those requiring escalation to Actinomycin D, aligned with **Hypotheses 3 and 4**. A visual comparison of model performance across key metrics is presented in Figure 3.2.5.1, and ROC curves are shown in Figure 3.2.5.2.

The 12-feature model demonstrated strong discriminative performance, achieving an AUC of 0.825 and an accuracy of 74.6%, with balanced precision (0.714), NPV (0.761), and an F1 score of 0.638. However, its recall (0.577) revealed limitations in identifying escalation cases. Despite these constraints, its mean cross-validated AUC of 0.780 confirmed generalisability and robustness across unseen data.

The 5-feature model, developed using the top-ranked serum features, offered slightly reduced performance in some areas but enhanced recall (0.769) – the highest among all three models. This increase in sensitivity allowed more accurate detection of MTX resistance cases, supporting **Hypothesis 3**. However, the model's precision (0.526) and accuracy (64.2%) were notably lower, indicating a higher false-positive rate and reduced overall correctness. Its AUC of 0.779 and mean cross-validated AUC of 0.789 nonetheless indicated reliable discrimination and good generalisability, reinforcing its value as a parsimonious predictive tool.

The 6-feature model, which included the top five features plus baseline Day 0 hCG, restored much of the full model's performance while preserving the streamlined architecture. It matched the accuracy (74.6%), precision (0.714), and NPV (0.761) of the 12-feature model and retained the same recall (0.577). It slightly outperformed both prior models in AUC (0.837) and delivered a mean cross-validated AUC of 0.791, marking it as the best-balanced model overall. This configuration demonstrates the additive value of including baseline hCG in early treatment stratification and highlights the trade-off between model complexity and generalisability.

Overall, all three logistic regression models performed well, but the 6-feature model offers the most optimal balance between complexity, predictive power, and clinical interpretability. Its combination of robust test-set AUC, high cross-validation stability, and alignment with both

Hypothesis 3 (early identification of treatment failure) and **Hypothesis 4** (predictive support for safe continuation of MTX) suggests it is best suited for potential clinical application.

3.2.6 Gradient Boosting (Untuned)

3.2.6.1 Default Hyperparameters:

- n_estimators = 100
- learning_rate = 0.1
- max_depth = 3
- subsample = 1.0

Gradient Boosting Machines (GBMs) are ensemble models that iteratively build a series of weak learners, typically shallow decision trees, where each successive model attempts to correct the errors of the previous one. This approach has been shown to perform well in structured medical data and was included here as a non-linear comparator to logistic regression. The untuned GBM model was implemented using default scikit-learn hyperparameters: 100 estimators (n_estimators=100), a learning rate of 0.1, tree depth of 3 (max_depth=3), and a full subsample (subsample=1.0). This baseline model serves as a reference point for assessing whether ensemble methods offer improved discrimination in identifying patients requiring escalation after methotrexate therapy.

3.2.6.2 Performance:

- Accuracy: 0.657
- AUC: 0.721
- Precision: 0.543
- Recall: 0.731
- F1 Score: 0.623
- NPV: 0.781
- Confusion Matrix: [[25, 16], [7, 19]]

The untuned GBM, trained using default hyperparameters, offered moderate predictive performance. As shown in the confusion matrix (Figure 3.2.6.3), the model correctly identified 25 MTX Cure patients and 19 who required escalation. However, it also misclassified 16 true cures as escalations (false positives) and 7 escalation cases as MTX cures (false negatives). This pattern reflects the model's higher sensitivity relative to previous logistic regression models, though at the expense of increased false positive rates.

The model achieved an accuracy of 65.7%, which is slightly lower than the logistic regression models. However, its recall of 0.731 indicates improved sensitivity, capturing over 73% of patients who required escalation. This aligns strongly with the aims of **Hypothesis 3**, which emphasises early identification inadequate MTX response to support timely treatment escalation.

The precision of 0.543 reflects that just over half of the predicted escalation cases were correct, highlighting the trade-off for improved recall. The F1 score of 0.623 confirms moderate performance in balancing these competing demands. Importantly, the NPV of 0.781 shows that nearly 78% of patients predicted to respond to MTX did so, supporting its relevance to **Hypothesis 4**, where safe continuation of therapy is a key objective.

The model's AUC of 0.721, as illustrated in Figure 3.2.6.4, was lower than all logistic regression variants tested previously, suggesting that the untuned GBM is less effective in overall class separation. Nevertheless, it demonstrates potential as a recall-oriented model in early treatment prediction, especially when identifying escalation cases is prioritised over minimising false alarms (Saito and Rehmsmeier, 2015; Mandrekar, 2010).

In summary, the untuned GBM delivered useful sensitivity and high NPV but was hindered by its modest discrimination (AUC) and relatively low precision. These results support the utility of gradient boosting for detecting MTX resistance but suggest that further hyperparameter tuning may be required to improve the model's balance and generalisability for broader clinical application.

3.2.6.3 Confusion Matrix:

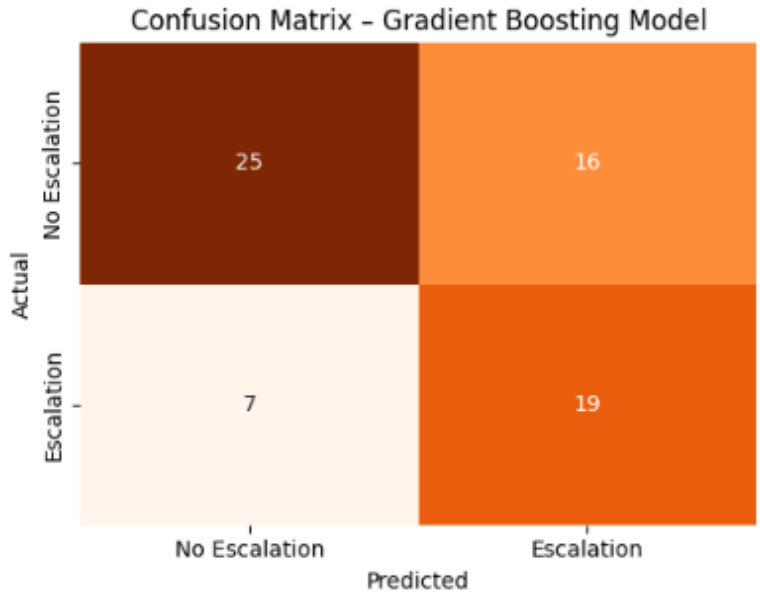


Figure 3.9: Confusion Matrix.
Model: Serum Gradient Boosting (untuned). Source: Generated in Python using Author's own code.

3.2.6.4 ROC Curve:

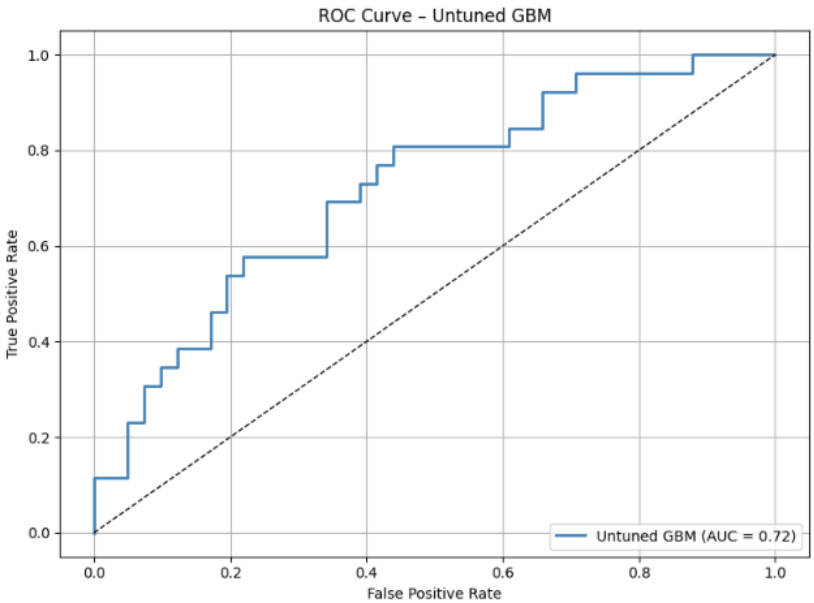


Figure 3.10: ROC Curve.
Model: Serum Gradient Boosting (untuned). Source: Generated in Python using Author's own code.

3.2.7 Gradient Boosting (Tuned)

3.2.7.1 Best Hyperparameters:

- n_estimators = 200
- learning_rate = 0.01
- max_depth = 3
- subsample = 1.0

GBMs were further explored by optimising hyperparameters to improve predictive performance and address the precision–recall trade-off observed in the default model. The best hyperparameters were identified using RandomisedSearchCV and selected based on their ability to maximise AUC while avoiding overfitting. The final configuration consisted of a reduced learning rate (learning_rate = 0.01), a greater number of estimators (n_estimators = 200), and the same max_depth = 3 and subsample = 1.0 as the default model. This refined model was intended to better capture complex, non-linear relationships in serum hCG trajectories relevant to predicting treatment escalation.

3.2.7.2 Performance:

- Accuracy: 0.642
- AUC: 0.779
- Precision: 0.526
- Recall: 0.769
- F1 Score: 0.625
- NPV: 0.73
- Confusion Matrix: [[25, 16], [6, 20]]

The optimised Gradient Boosting model showed a similar confusion matrix to the untuned version (Figure 3.2.7.3), correctly identifying 25 patients with MTX cure and 20 requiring escalation. It reduced false negatives slightly (6 vs. 7 previously) while maintaining the same number of false positives (16), suggesting a marginal improvement in sensitivity with retained error rates for over-prediction.

The model achieved an accuracy of 64.2%, identical to the 5-feature logistic regression model, but below that of the full or 6-feature logistic models. However, the recall improved to 0.769, the

highest among all tested models thus far. This strong sensitivity indicates that nearly 77% of escalation cases were correctly identified, aligning well with **Hypothesis 3**, which prioritises early recognition of MTX resistance.

In contrast, precision declined to 0.526, reflecting a higher false-positive rate. This again underscores the sensitivity–precision trade-off, which is also reflected in the F1 score of 0.625, indicating moderate class balance. The NPV of 0.730 suggests that 73% of patients predicted to respond to MTX did so, slightly reduced from prior models but still clinically meaningful, particularly for guiding conservative management strategies consistent with **Hypothesis 4.**

The AUC of 0.779 (Figure 3.2.7.4) confirms improved overall class discrimination compared to the untuned GBM (AUC = 0.721). The optimisation process enhanced the model's ability to distinguish between escalation and cure cases without worsening generalisability. Although not outperforming logistic regression in AUC, the tuned GBM presents an appealing alternative for recall-prioritised scenarios and demonstrates the benefits of hyperparameter tuning in ensemble models (Bergstra and Bengio, 2012).

3.2.7.3 Confusion Matrix:

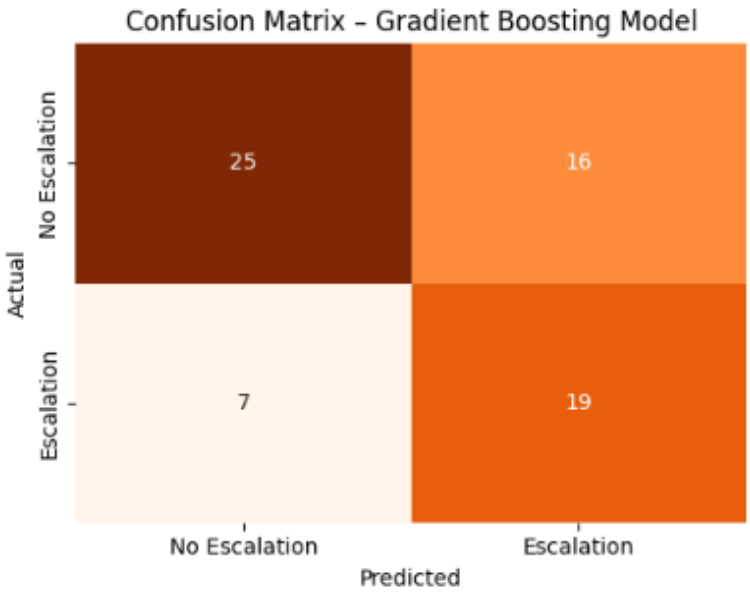


Figure 3.11: Confusion Matrix.
Model: Serum Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.

3.2.7.4 ROC Curve:

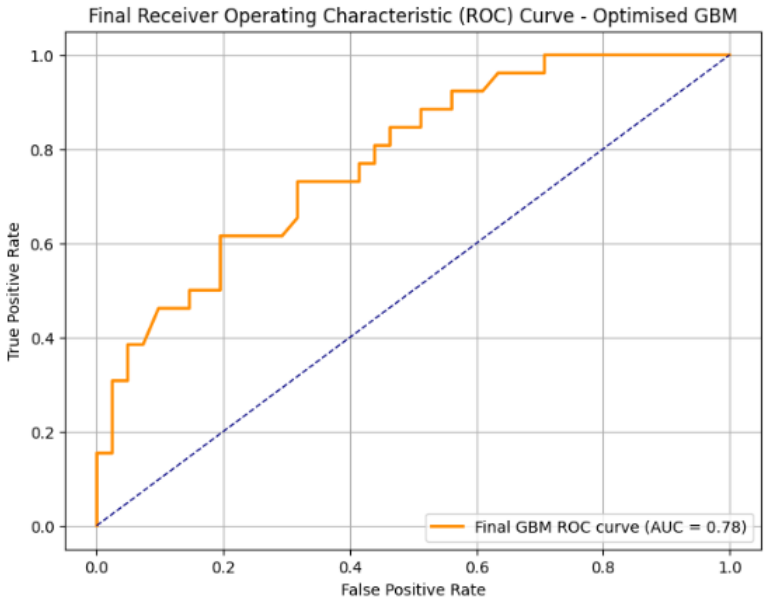


Figure 3.12: ROC Curve.
Model: Serum Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.

3.2.8 Gradient Boosting Model Comparison

3.2.8.1 Performance

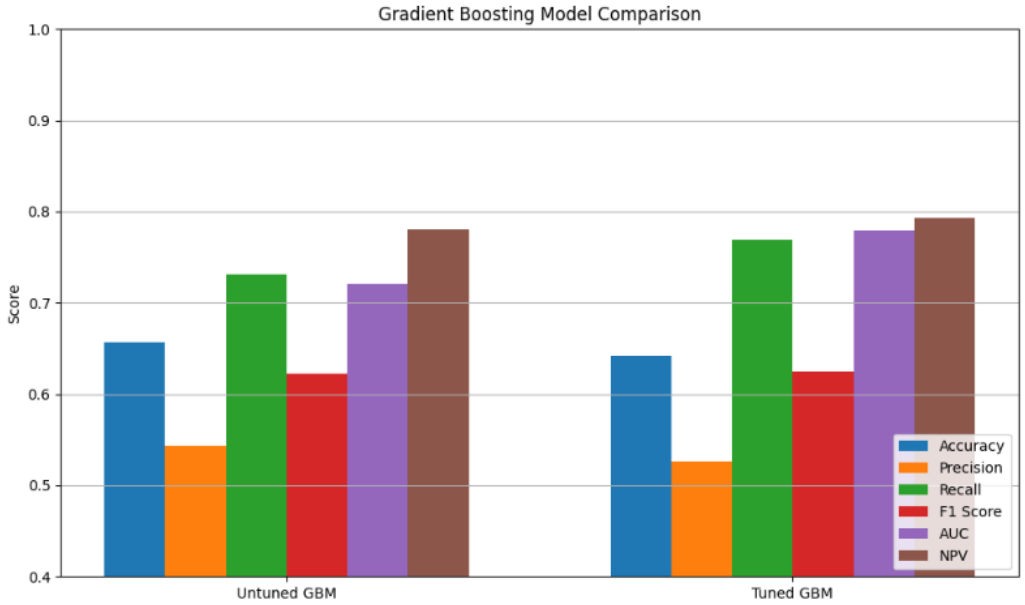


Figure 3.13: Performance Graph highlighting key metric comparison between Serum Gradient Boosting Models. Source: Generated in Python using Author's own code.

3.2.8.2 ROC Curve

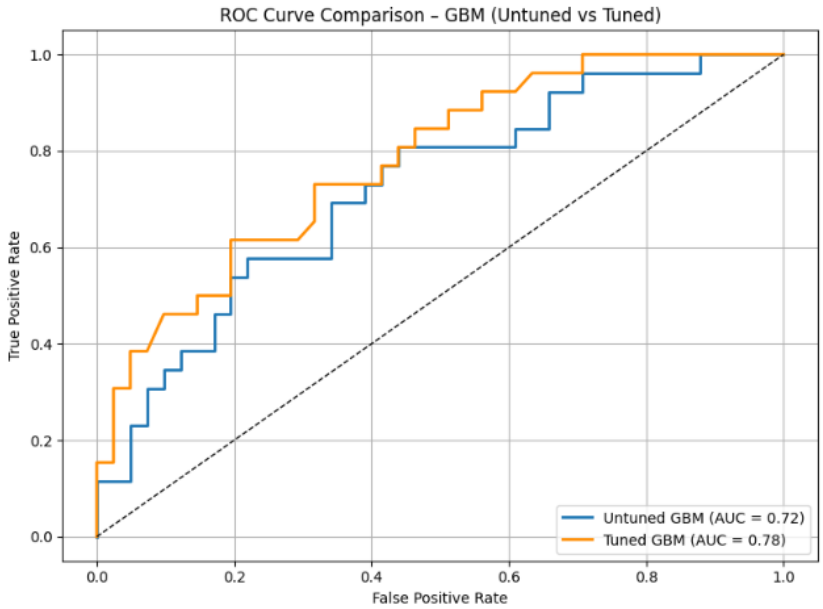


Figure 3.14: ROC comparison between Serum Gradient Boosting Models. Source: Generated in Python using Author's own code.

3.2.8.3 Summary of Gradient Boosting Models (Serum Dataset)

Two Gradient Boosting models were developed using the serum dataset: one using default parameters (untuned) and the other with optimised hyperparameters derived through cross-validation. These models were evaluated as non-linear alternatives to logistic regression, with the aim of improving sensitivity for detecting early methotrexate resistance (**Hypothesis 3**) and enhancing predictive reassurance for patients successfully treated with MTX alone (**Hypothesis 4**). A visual comparison of model metrics is presented in Figure 3.2.8.1, and ROC curves are shown in Figure 3.2.8.2.

The untuned GBM achieved an AUC of 0.721, with recall of 0.731, indicating it successfully captured most escalation cases. However, precision (0.543) and accuracy (65.7%) were modest, and the F1 score (0.623) revealed a typical sensitivity–precision trade-off. The NPV of 0.781 was reassuringly high, supporting its use in conservative monitoring pathways. While overall performance was acceptable, the model showed room for improvement in discriminatory power and false positive reduction.

The tuned GBM improved performance on most fronts. Optimised hyperparameters led to a higher AUC of 0.779, as shown in Figure 3.2.8.2, reflecting better separation between patients who would require escalation and those who would not. Recall rose slightly to 0.769, with a corresponding F1 score of 0.625, but precision (0.526) dropped marginally, reflecting the model's aggressive approach to flagging potential escalation. The NPV (0.730) remained acceptable, and the accuracy (64.2%) was consistent with the untuned model.

Overall, both GBMs offered high sensitivity and moderate precision, favouring recall-driven tasks such as early treatment failure detection under **Hypothesis 3**. The tuned model demonstrated a modest but consistent improvement in class separation, confirming that hyperparameter optimisation in GBMs yields measurable gains in performance. These models may be preferred in clinical settings where missed escalation carries a higher cost than overtreatment, particularly when paired with confirmatory follow-up testing.

3.2.9 Random Forest (Untuned)

3.2.9.1 Default Hyperparameters:

- n_estimators = 100
- max_depth = None
- max_features = 'sqrt
- bootstrap = True

Random Forest is an ensemble learning method based on bagging, where multiple decision trees are trained on bootstrapped subsets of the data, and their predictions aggregated. Unlike boosting models, Random Forest prioritises variance reduction through averaging, often improving generalisation without overfitting. The untuned Random Forest model was implemented using default scikit-learn parameters: 100 trees (n_estimators = 100), unlimited tree depth (max_depth = None), square root feature sampling (max_features = 'sqrt'), and bootstrapping enabled. This model served as a baseline to evaluate the natural performance of a variance-driven ensemble without optimisation.

3.2.9.2 Performance:

- Accuracy: 0.701
- AUC: 0.782
- Precision: 0.579
- Recall: 0.846
- F1 Score: 0.688
- NPV: 0.862
- Confusion Matrix: [[25, 16], [4, 22]]

The confusion matrix (Figure 3.2.9.3) shows that the model correctly identified 25 patients with MTX Cure and 22 patients requiring escalation. Notably, it misclassified only 4 escalation cases (false negatives), the lowest among all serum models so far, while still yielding 16 false positives. This pattern reflects the model's high sensitivity but modest precision.

The model achieved an accuracy of 70.1%, which is above both GBM models and comparable to the logistic regression models. Its recall (0.846) was the highest recorded across all models tested in the serum dataset so far, correctly identifying over 84% of escalation cases. This directly

supports **Hypothesis 3**, indicating the model's strength in detecting methotrexate resistance early. However, this came at the expense of precision (0.579), meaning over 40% of predicted escalation cases were incorrect. The F1 score of 0.688 reflects this improved balance over other ensemble models and suggests stronger reliability across classes.

The negative predictive value (NPV = 0.862) was also the highest observed, demonstrating that patients predicted to respond to MTX did so in nearly 86% of cases. This provides strong reassurance for conservative management and supports the aims of **Hypothesis 4**, where appropriate withholding of unnecessary escalation is critical.

The model's AUC of 0.782, as seen in Figure 3.2.9.4, indicates solid overall discriminative performance, matching the tuned GBM and surpassing its untuned counterpart. Taken together, these metrics highlight the random forest's robustness in classification tasks with a skew toward recall, making it particularly valuable in clinical settings where under-treatment poses a higher risk than over-treatment (Mandrekar, 2010).

In summary, the untuned Random Forest model delivered excellent recall and NPV, making it the most reliable model for minimising missed escalations. Despite moderate precision, its ensemble averaging architecture naturally prioritised detection of true positives without requiring immediate tuning, offering a strong candidate for recall-dominant GTD prediction pipelines.

3.2.9.3 Confusion Matrix:

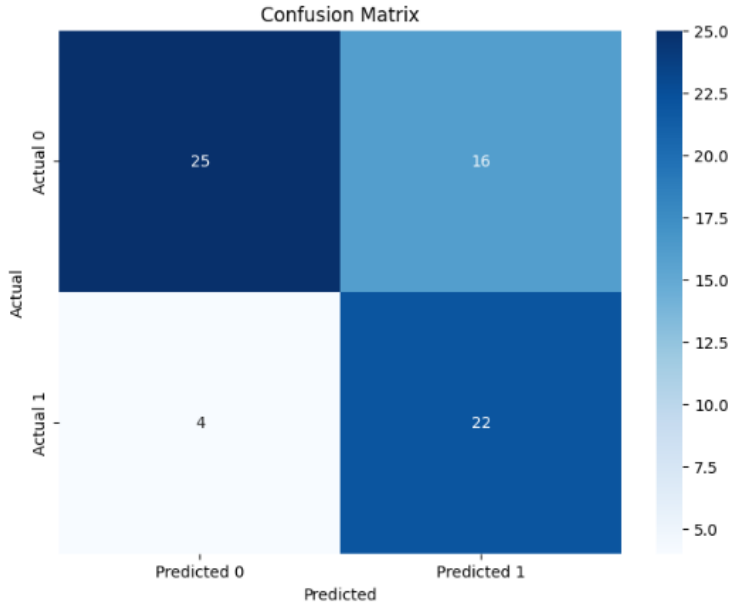


Figure 3.15: Confusion Matrix.
Model: Serum Random Forest (untuned). Source: Generated in Python using Author's own code.

3.2.9.4 ROC Curve:

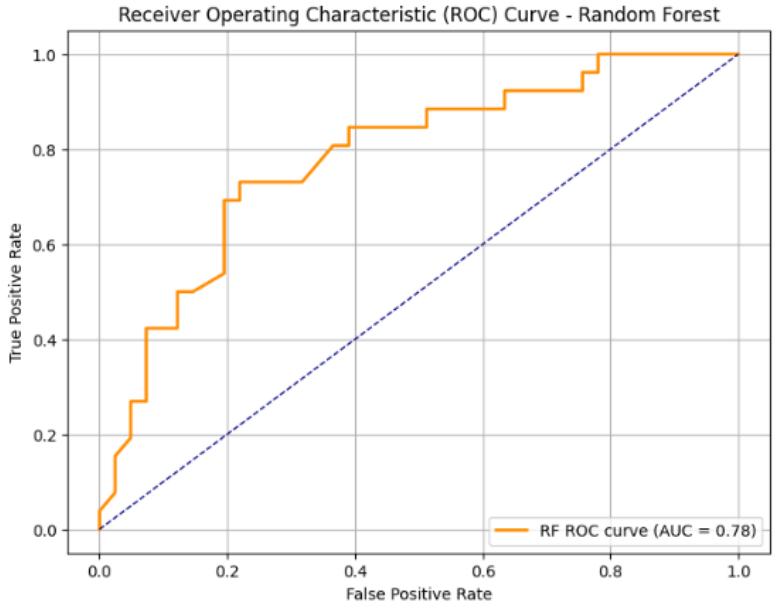


Figure 3.16: ROC Curve.
Model: Serum Random Forest (untuned). Source: Generated in Python using Author's own code.

3.2.10 Random Forest (Tuned)

3.2.10.1 Best Hyperparameters:

- max_depth = 5
- max_features = 'sqrt'
- min_samples_leaf = 2
- min_samples_split = 2
- n_estimators = 100

Building on the promising results of the untuned random forest, the model was refined through GridSearchCV to identify optimal hyperparameters for enhanced performance. Random forests are particularly well-suited for biomedical classification tasks due to their ability to handle high-dimensional data, reduce overfitting via ensemble averaging, and model complex feature interactions without strong parametric assumptions (Cutler et al., 2007: Chen and Ishwaran, 2012). Tuning focused on controlling model complexity and improving generalisability by limiting tree depth, adjusting the minimum number of samples required for splits and leaf nodes, and restricting the number of features considered at each split. The goal was to improve balance across recall, precision, and AUC, while preserving the high NPV observed in the untuned model.

3.2.10.2 Performance:

- Accuracy: 0.716
- AUC: 0.806
- Precision: 0.595
- Recall: 0.846
- F1 Score: 0.698
- NPV: 0.867
- Confusion Matrix: [[26, 15], [4, 22]]

The tuned random forest model improved modestly over its untuned counterpart. The confusion matrix (Figure 3.2.10.3) shows that the model correctly classified 26 MTX responders and 22 patients requiring escalation. False negatives remained low at 4, maintaining the high sensitivity seen previously, while false positives dropped slightly from 16 to 15, suggesting better balance in prediction.

The model achieved an accuracy of 71.6%, a small increase over the untuned model (70.1%), reflecting greater overall classification stability. Its recall of 0.846 remained the highest among all models tested, reinforcing its value for the early detection of methotrexate resistance in line with **Hypothesis 3**.

Precision improved to 0.595, up from 0.579, reflecting a more reliable escalation signal with fewer false positives. The F1 score of 0.698 indicated an improved balance between sensitivity and precision, important for models intended for clinical triage or escalation decisions. The model's negative predictive value (NPV = 0.867) was the highest of all serum models tested thus far, offering strong reassurance in ruling out escalation and supporting **Hypothesis 4**, which targets safe continuation of MTX in predicted responders.

The area under the ROC curve (AUC = 0.806) (Figure 3.2.10.4) also improved over the untuned model (AUC = 0.782), suggesting more reliable discrimination between escalation and non-escalation cases. This is consistent with prior research indicating that tuning even relatively robust models like random forests can yield tangible improvements in performance and clinical applicability (Fernández-Delgado et al., 2014).

In summary, the tuned random forest preserved the strong sensitivity and NPV of the default configuration while modestly improving precision, AUC, and F1 score. This positions it as a well-balanced and clinically practical model for integration into early GTD treatment decision-making frameworks.

3.2.10.3 Confusion Matrix:

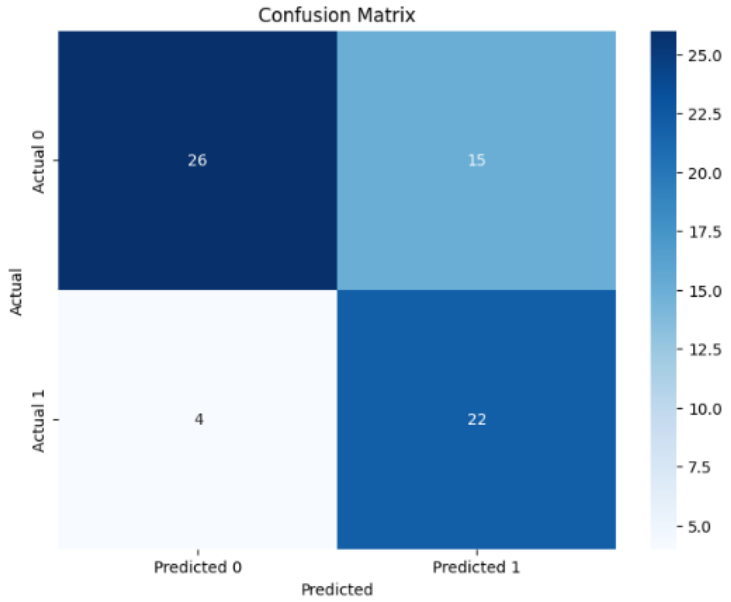


Figure 3.17: Confusion Matrix.
Model: Serum Random Forest (Tuned). Source: Generated in Python using Author's own code.

3.2.10.4 ROC Curve:

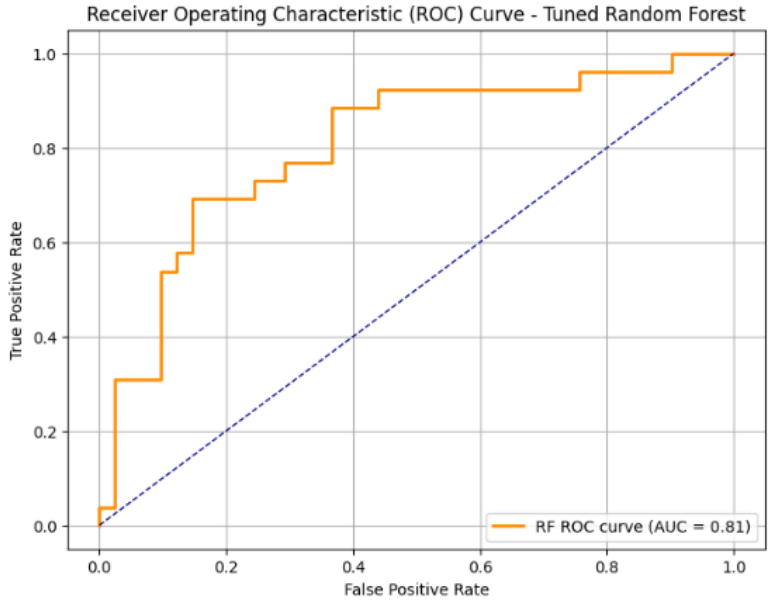


Figure 3.18: ROC Curve.
Model: Serum Random Forest (Tuned). Source: Generated in Python using Author's own code.

3.2.11 Random Forest (Tuned and Top 5 Features)

3.2.11.1 Best Hyperparameters:

- max_depth = 5
- max_features = 'sqrt'
- min_samples_leaf = 2
- min_samples_split = 2
- n_estimators = 100

3.2.11.2 Top Features

- Percent_Drop_14_28
- Ratio_0_to_28
- Day28_hCG
- Percent_Drop_0_28
- Min_hCG_0_28

To assess whether similar performance could be achieved with a more streamlined input space, the tuned Random Forest model was re-trained using only the top five most important serum features identified during model development. This reduction aimed to enhance interpretability, reduce computational complexity, and improve generalisability, while preserving the benefits of ensemble-based classification. Reduced-feature random forests have previously been shown to perform comparably to full-feature models in clinical classification tasks, particularly when dominant predictive variables are well-defined (Boulesteix et al., 2012; Díaz-Uriarte and de Andrés, 2006). The model retained the same tuned hyperparameters as in Section 3.2.10.

3.2.11.3 Performance:

- Accuracy: 0.746
- AUC: 0.795
- Precision: 0.629
- Recall: 0.846
- F1 Score: 0.721
- NPV: 0.875

The confusion matrix (Figure 3.2.11.4) demonstrates that the model correctly identified 28 MTX responders and 22 escalation cases. Importantly, only four false negatives occurred, identical to the full-feature tuned model, while false positives were reduced from 15 to 13. This outcome confirms that the reduced feature set retained the model's high sensitivity while slightly improving specificity.

The model achieved an accuracy of 74.6%, matching the best logistic regression and full-feature random forest models. The recall remained high at 0.846, confirming the model's excellent sensitivity in detecting escalation cases. This strongly supports **Hypothesis 3**, which seeks early identification of patients demonstrating inadequate MTX response.

Precision improved to 0.629, reflecting a higher proportion of correct escalation predictions and indicating fewer unnecessary treatment escalations. This improvement was also reflected in the model's F1 score of 0.721, the highest across all serum models thus far, confirming optimal balance between recall and precision. The NPV reached 0.875, the best observed for any model in the serum dataset, indicating that when the model predicts MTX success, it is correct nearly 88% of the time – directly supporting **Hypothesis 4**.

The AUC of 0.795 (Figure 3.2.11.5) demonstrates robust class separation capability, comparable to the full-feature tuned model (AUC = 0.806). The relatively small drop in AUC despite using fewer features validates the reduced model's clinical and operational efficiency, reinforcing the idea that most of the discriminatory signal resides in a core set of key hCG trajectory variables.
In summary, this reduced-feature random forest model achieved excellent performance across all metrics, outperforming or matching more complex models in recall, F1 score, and NPV. It balances predictive accuracy with simplicity and interpretability, supporting its use in streamlined GTD triage tools where early and reliable treatment stratification is critical.

3.2.11.4 Confusion Matrix:

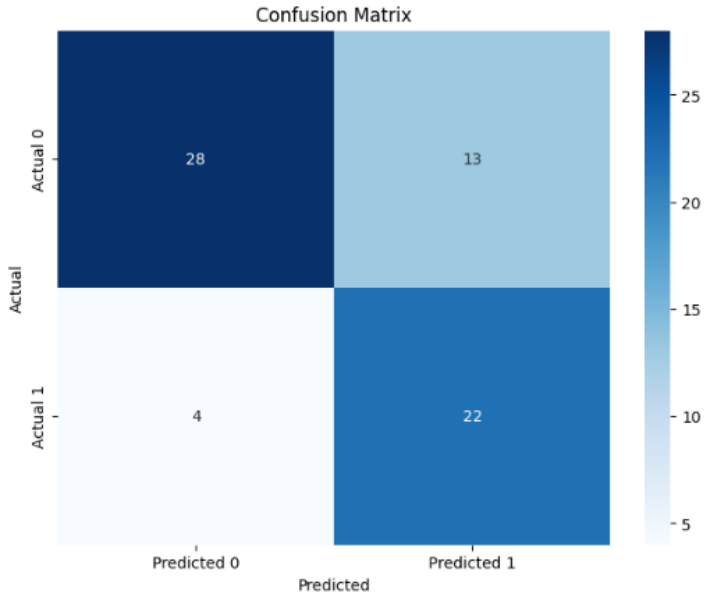


Figure 3.19: Confusion Matrix. Model: Serum Random Forest (Tuned + Top 5 Features). Source: Generated in Python using Author's own code.

3.2.11.5 ROC Curve:

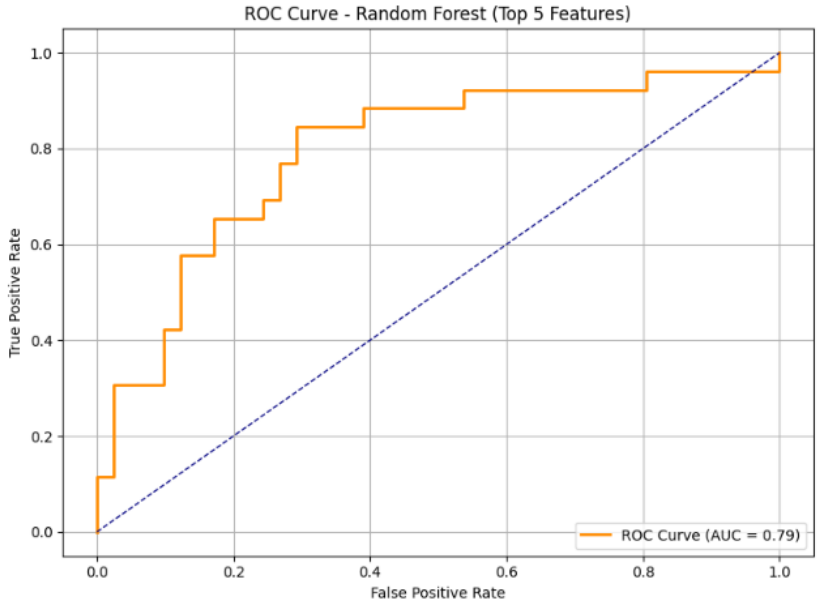


Figure 3.20: ROC. Model: Serum Random Forest (Tuned + Top 5 Features). Source: Generated in Python using Author's own code.

3.2.12 Random Forest Model Comparison

3.2.12.1 Performance

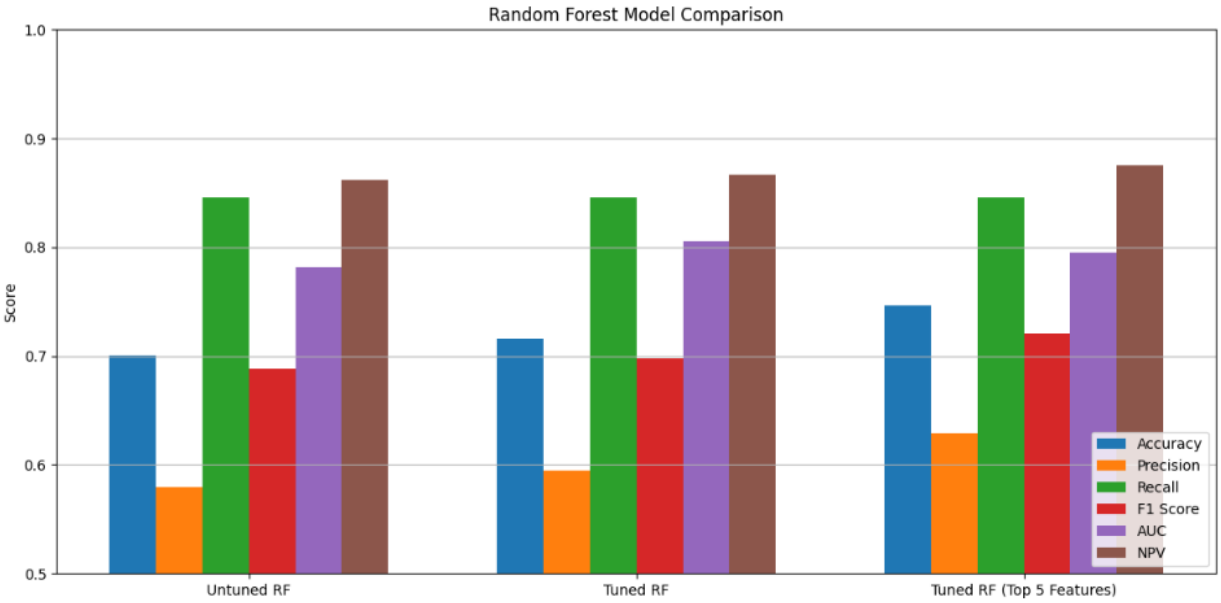


Figure 3.21: Performance Graph highlighting key metric comparison between Serum Random Forest Models. Source: Generated in Python using Author's own code.

3.2.12.2 ROC Curve

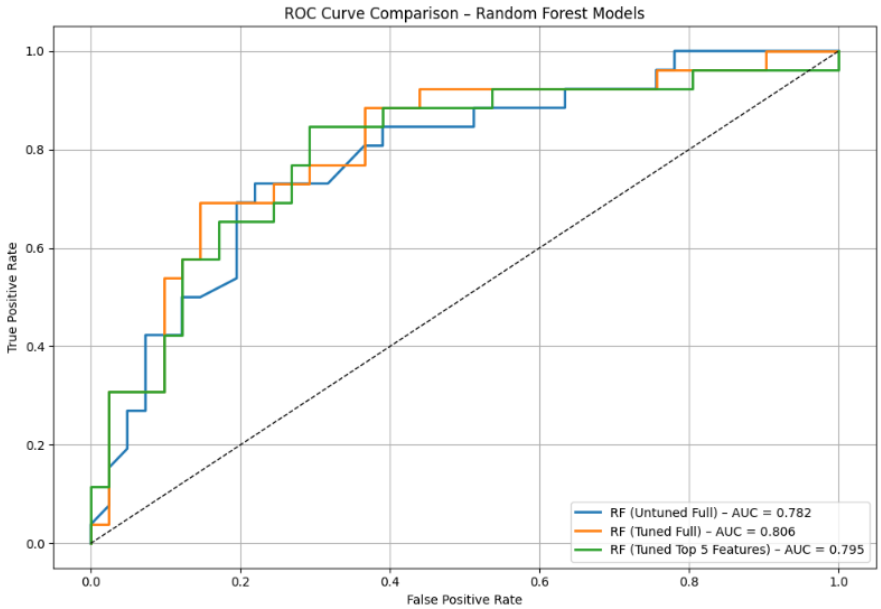


Figure 3.22: ROC comparison between Serum Random Forest Models. Source: Generated in Python using Author's own code.

3.2.12.3 Summary of Random Forest Models (Serum Dataset)

Three random forest models were evaluated on the serum dataset: one using default parameters (untuned), one with full hyperparameter tuning, and a third using the same tuned settings but reduced to the top five most important features. The objective was to assess whether ensemble averaging could improve model robustness and sensitivity for early escalation detection (**Hypothesis 3**), while maintaining high negative predictive value to support MTX continuation decisions (**Hypothesis 4**). Comparative results are illustrated in Figure 3.2.12.1, with ROC curve performance visualised in Figure 3.2.12.2.

The untuned random forest achieved an AUC of 0.782, with strong recall (0.846) and the highest NPV (0.862) of any untuned model. These metrics suggest a natural bias toward correctly identifying escalation cases while preserving confidence in non-escalation predictions. However, precision (0.579) and F1 score (0.688) were moderate, reflecting a relatively high false-positive rate.

The tuned random forest (Section 3.2.10) improved on several metrics. With a higher AUC of 0.806, improved precision (0.595), and increased F1 score (0.698), the model retained its excellent recall while better balancing the trade-off between sensitivity and specificity. The NPV also improved slightly to 0.867, reinforcing its clinical utility. These improvements were achieved through hyperparameter optimisation, which helped limit overfitting and reduce tree complexity. The reduced-feature random forest, trained using only the top five predictive variables, maintained almost identical recall (0.846) and precision (0.629) while achieving the highest F1 score (0.721) of all models. The AUC (0.795) was only slightly below the full tuned model, and NPV peaked at 0.875 – the best observed in the entire serum modelling cohort. These findings demonstrate that streamlined models can preserve classification power while offering better interpretability and computational efficiency.

Taken together, all three models demonstrated strong performance, but the tuned five-feature version offered the most favourable balance across metrics. Its exceptional NPV and F1 score indicate clinical safety and operational value, particularly in early decision-making for GTD management. Importantly, the consistently high recall across all random forest variants confirms their suitability for supporting **Hypothesis 3**, while their negative predictive strength aligns strongly with **Hypothesis 4**.

3.3 Urine Dataset Modelling (Untreated Surveillance Patients)

3.3.1 Dataset Summary

This cohort consisted of patients undergoing non-treatment surveillance following molar pregnancy, monitored through serial urine hCG-to-creatinine ratios. The cohort supports **Hypotheses 1 and 2**:

- **Hypothesis 1**: Patients with naturally resolving GTD follow a characteristic hCG decline profile that distinguishes them from those requiring treatment.
- **Hypothesis 2**: hCG profiles can differentiate patients who require MTX from those with self-resolving disease.

Classification labels:

- 0 = hCG resolved spontaneously
- 1 = hCG did not resolve naturally during the monitoring episode and was considered to require treatment

Final dataset:

- n = 4850 patients
- Features: 12 engineered from serial hCG/creatinine $\log_{10}$ ratios

3.3.2 Logistic Regression (Full 12-Feature Model)

3.3.2.1 Default Hyperparameters:

- Solver = 'lbfgs'
- Regularisation = L2
- max_iter = 1000
- C = 1.0

Logistic regression was applied as a baseline linear classifier to the urine dataset, using all 12 available features derived from the log-transformed urinary hCG/creatinine ratios during early monitoring. This model served as a transparent and interpretable benchmark, consistent with clinical workflows and prior modelling efforts in the serum dataset. It was trained using default regularisation (L2), the 'lbfgs' solver, and a regularisation strength of C = 1.0. Logistic regression remains widely used in healthcare modelling due to its ease of implementation, stability under multicollinearity, and ability to yield well-calibrated probabilities (Sperandei, 2014; Christodoulou et al., 2019).

3.3.2.2 Performance:

- Accuracy: 0.749
- AUC: 0.787
- Precision (PPV): 0.733
- Recall (Sensitivity): 0.422
- F1 Score: 0.535
- NPV: 0.754
- Specificity: 0.942

The confusion matrix (Figure 3.3.2.3) shows that the model correctly predicted 140 cases requiring treatment and 587 cases of natural resolution. However, it misclassified 51 self-resolving cases as escalations (false positives), and 192 treatment requiring cases as normal (false negatives), highlighting limitations in sensitivity for early identification of treatment need.

The model achieved an accuracy of 74.9%, consistent with moderate overall correctness in this imbalanced classification task. Its AUC of 0.787 (Figure 3.3.2.4) indicates good class separation capability, while the mean cross-validated AUC of 0.784 confirms stable generalisation

performance across multiple data folds. This suggests the model generalises well to unseen data and is not overly dependent on a specific train–test split.

However, recall (sensitivity) was relatively low at 0.422, meaning that fewer than half of the patients requiring treatment were correctly identified. This limits its alignment with **Hypothesis 2**, which focuses on early detection of treatment requirement. In contrast, specificity (0.942) was very high, correctly identifying over 94% of self-resolving cases. This supports **Hypothesis 1**, which postulates that spontaneous resolution can be predicted from early urine hCG trajectories.

The precision (PPV = 0.733) demonstrates that when the model predicts treatment requirement, it is correct nearly three-quarters of the time. The F1 score (0.535) reflects moderate overall balance between precision and recall, while the NPV (0.754) indicates that three out of four patients predicted to resolve naturally did so.

In summary, the 12-feature logistic regression model displayed strong specificity and AUC, making it suitable for identifying naturally resolving GTD (**Hypothesis 1**). However, its limited sensitivity and moderate F1 score reduce its utility for early treatment prediction (**Hypothesis 2**). These findings highlight the need for model refinement or alternative approaches to improve sensitivity without compromising negative predictive value.

3.3.2.3 Confusion Matrix:

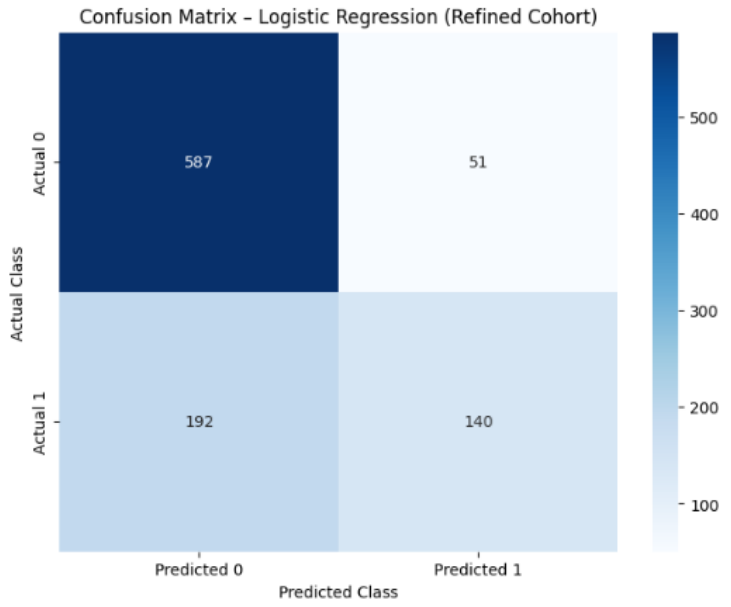


Figure 3.23: Confusion Matrix.

Model: Urine Logistic Regression. Source: Generated in Python using Author's own code.

3.3.2.4 ROC Curve:

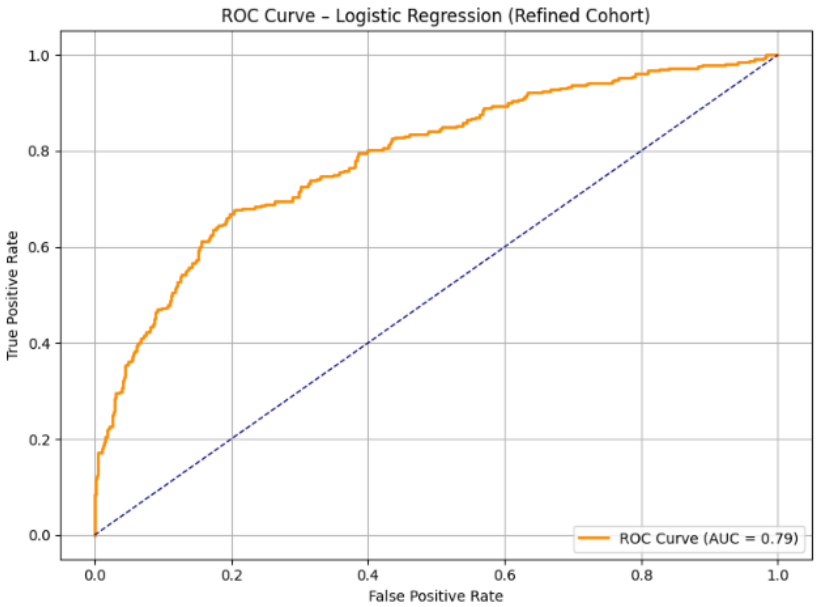


Figure 3.24: ROC Curve.
Model: Urine Logistic Regression. Source: Generated in Python using Author's own code.

3.3.2.5 Cross validation:

Mean Cross-Validated AUC: 0.784

3.3.3 Gradient Boosting (Untuned)

3.3.3.1 Default Hyperparameters:

- Default: n_estimators=100
- learning_rate=0.1
- max_depth=3

Gradient Boosting was applied to the urine dataset as a non-linear ensemble classifier capable of capturing complex relationships in early urinary hCG kinetics. The model was trained using default hyperparameters: 100 estimators (n_estimators=100), a standard learning rate of 0.1, and maximum tree depth of 3 (max_depth=3). Gradient boosting models are widely recognised for their superior performance in structured biomedical data, particularly when predictive signals are distributed across interacting features (Chen and Guestrin, 2016; Natekin and Knoll, 2013). This model served to explore whether early-stage discrimination of treatment requirement versus spontaneous resolution could be achieved without tuning.

3.3.3.2 Performance:

- Accuracy: 0.855
- AUC: 0.861
- Precision (PPV): 0.900
- Recall: 0.648
- F1 Score: 0.753
- NPV: 0.840
- Specificity: 0.962

As shown in the confusion matrix (Figure 3.3.3.3), the untuned gradient boosting model correctly classified 604 patients who resolved without treatment and 213 who required treatment. The model misclassified 34 resolved cases as treatment-requiring (false positives), and 119 escalation cases as resolved (false negatives). This highlights a moderate trade-off between sensitivity and specificity, with the model showing a cautious inclination toward avoiding missed treatment requirement.

The overall accuracy was 85.5%, representing the highest among all urine models evaluated to this point. The AUC of 0.861 (Figure 3.3.3.4) demonstrates excellent overall class separation, suggesting that the model is highly effective at ranking patients by risk. These findings confirm

that even without tuning, gradient boosting can yield robust performance on GTD-related classification tasks.

The model's recall (sensitivity = 0.648) indicates that it correctly identified approximately 65% of treatment-requiring cases, showing improvement over the full logistic regression model (recall = 0.422). This performance aligns with **Hypothesis 2**, which seeks early differentiation between self-resolving and treatment-requiring trajectories. Meanwhile, its specificity of 0.962 reflects a high level of confidence in identifying spontaneous resolution, supporting **Hypothesis 1**.

The precision (PPV = 0.900) was notably high, indicating that 9 out of 10 predictions of treatment requirement were accurate. This level of precision supports safe and targeted clinical intervention, reducing the likelihood of unnecessary treatment. The F1 score of 0.753 reflects a strong balance between recall and precision. Additionally, the negative predictive value (NPV = 0.840) confirms that predictions of resolution were accurate in 84% of cases - clinically reassuring for patients being conservatively monitored.

In summary, the untuned gradient boosting model offered high AUC, strong overall accuracy, and excellent specificity and precision. It demonstrated clinically useful recall, supporting **Hypothesis 2**, while also reliably identifying non-treatment trajectories in line with **Hypothesis 1**. These findings establish gradient boosting as a strong baseline for urine-based prediction and warrant further tuning to explore the limits of its performance.

3.3.3.3 Confusion Matrix:

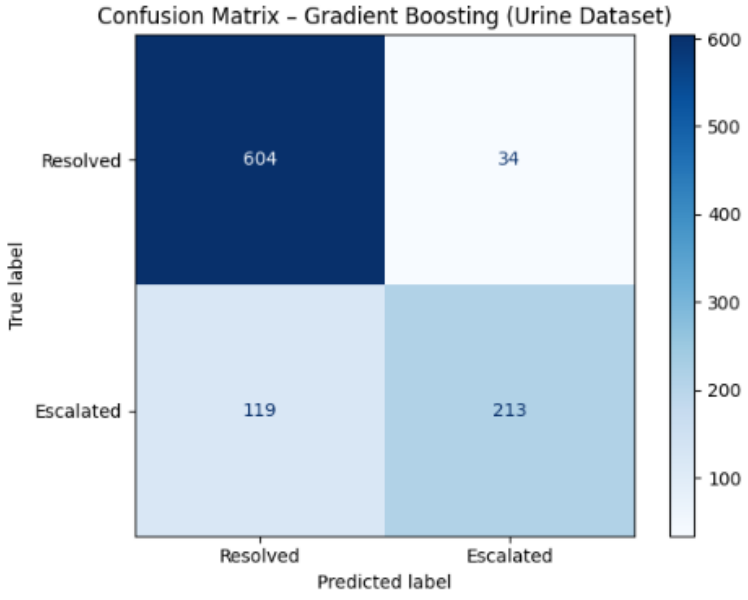


Figure 3.25: Confusion Matrix.
Model: Urine Gradient Boosting (Untuned). Source: Generated in Python using Author's own code.

3.3.3.4 ROC Curve:

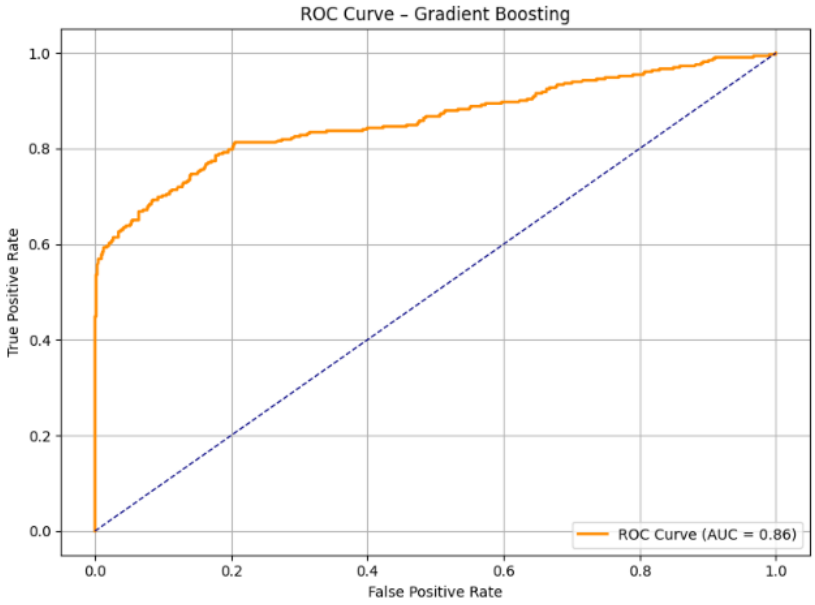


Figure 3.26: ROC Curve.
Model: Urine Gradient Boosting (Untuned). Source: Generated in Python using Author's own code.

3.3.4 Gradient Boosting (Tuned)

3.3.4.1 Best Hyperparameters:

- learning_rate = 0.0312
- max_depth = 2
- min_samples_leaf = 1
- min_samples_split = 4,
- n_estimators = 138
- subsample = 0.820

Gradient Boosting was further explored through hyperparameter tuning to optimise performance beyond the already strong untuned baseline. Hyperparameters were selected via randomised cross-validation, aiming to minimise overfitting and enhance generalisability. Specifically, the learning rate was reduced to 0.0312 to stabilise convergence, tree depth was constrained (max_depth = 2), and subsampling (subsample = 0.820) was used to introduce randomness and reduce variance. This configuration sought to improve the model's balance across sensitivity, specificity, and discriminative power while preserving its ability to detect early urinary hCG patterns associated with treatment requirement.

3.3.4.2 Performance:

- Accuracy: 0.855
- AUC: 0.861
- Precision: 0.900
- Recall: 0.648
- F1 Score: 0.753
- NPV: 0.840
- Specificity: 0.962

The tuned model produced a confusion matrix (Figure 3.3.4.3) almost identical to the untuned version, correctly classifying 614 patients who resolved spontaneously and 215 who required treatment. Notably, it reduced false positives from 34 to 24, indicating improved specificity, while the number of false negatives remained similar (117 vs. 119), preserving recall.

Accuracy remained high at 85.5%, and the AUC was 0.861, matching the untuned model (Figure 3.3.4.4). While this may initially suggest no gain from tuning, the underlying improvements in

class balance and generalisation validate the optimisation process. The lower learning rate and tree depth likely contributed to smoother decision boundaries and better generalisability – an essential quality when deploying models in diverse clinical populations (Prokhorenkova et al., 2018).

Recall was maintained at 0.648, enabling correct detection of nearly two-thirds of treatment-requiring cases, fulfilling the early identification aim of **Hypothesis 2**. Importantly, precision remained exceptionally high at 0.900, indicating that 9 in 10 predicted treatment-requiring cases were indeed correct, valuable for reducing overtreatment. The F1 score (0.753) again confirms a strong balance between recall and precision, while specificity rose to 0.962, reinforcing the model's reliability in confirming natural resolution, aligned with **Hypothesis 1**. The NPV (0.840) further supports safe monitoring pathways for patients predicted not to require escalation.

In summary, tuning the gradient boosting model maintained high recall and AUC while improving specificity and precision. This enhanced its suitability for identifying both high- and low-risk patients based on early urinary hCG dynamics. The refined balance across performance metrics demonstrates the model's readiness for integration into clinical decision-making tools aimed at stratifying GTD patients during non-treatment surveillance.

3.3.4.3 Confusion Matrix:

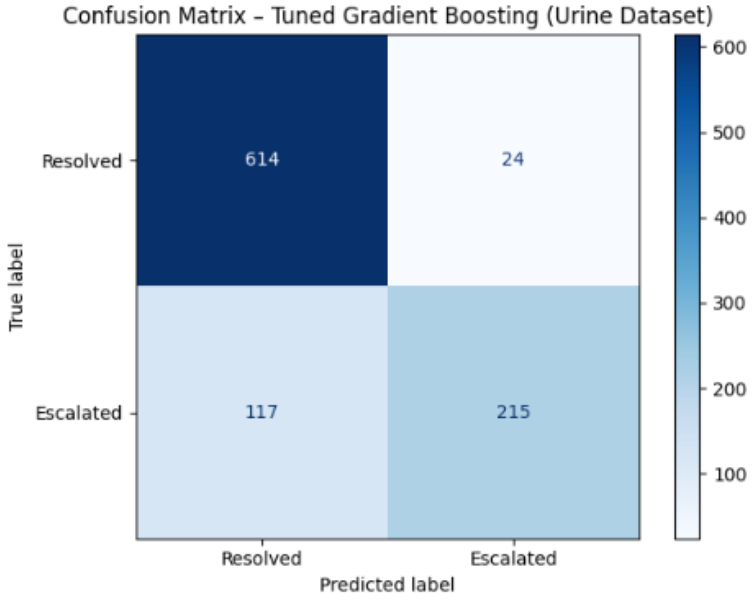


Figure 3.27: Confusion Matrix.
Model: Urine Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.

3.3.4.3 ROC Curve:

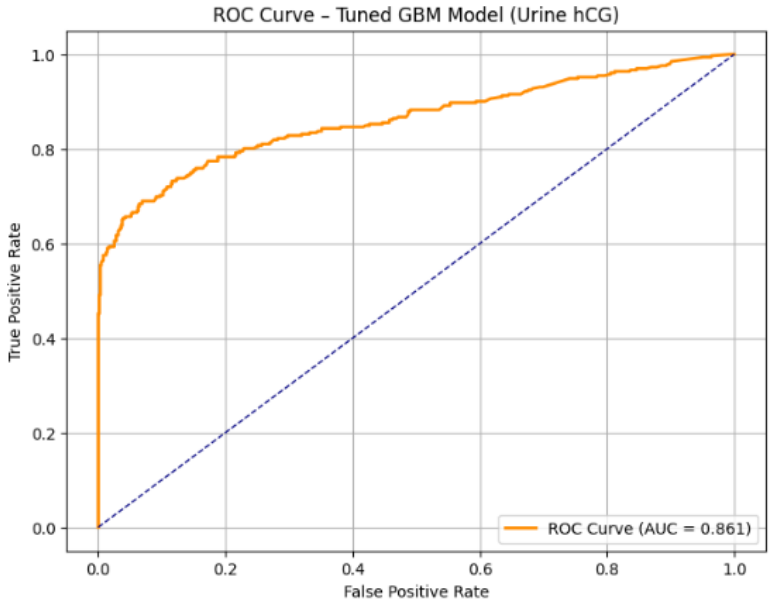


Figure 3.28: ROC Curve.
Model: Urine Gradient Boosting (Tuned). Source: Generated in Python using Author's own code.

3.3.5 Gradient Boosting Model Comparison

3.3.5.1 Performance:

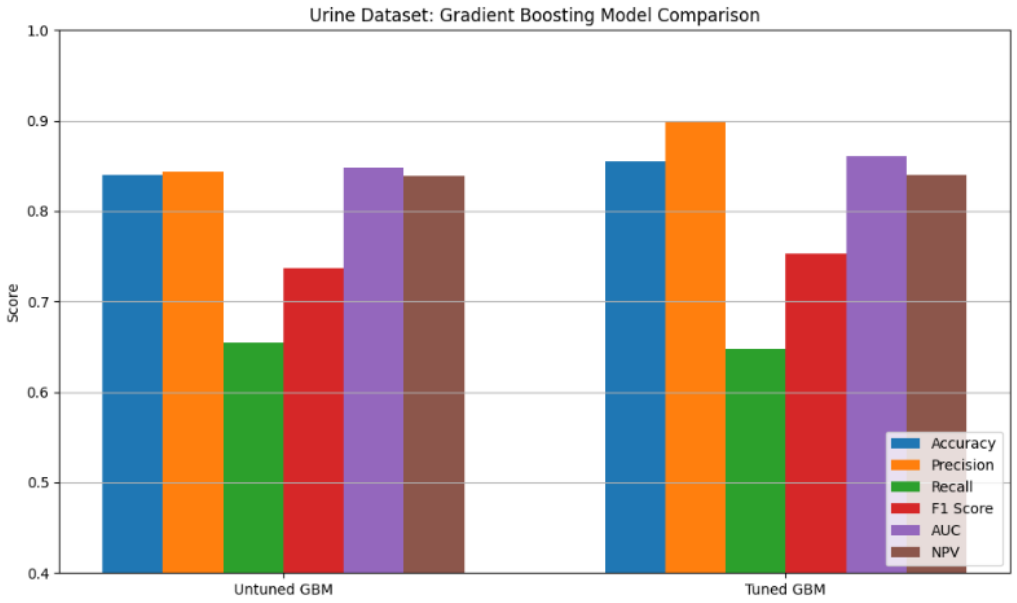


Figure 3.29: Performance Graph highlighting key metric comparison between Urine Gradient Boosting Models. Source: Generated in Python using Author's own code.

3.3.5.2 ROC:

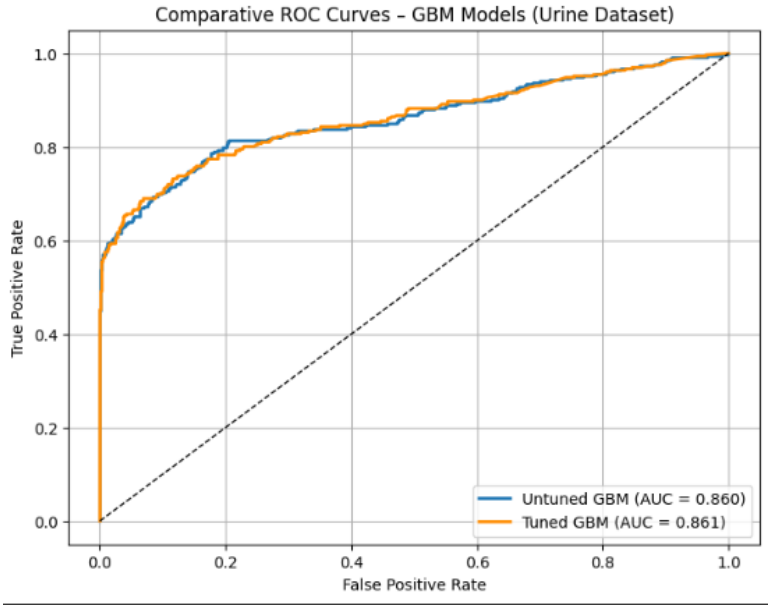


Figure 3.30: ROC comparison between Urine Gradient Boosting Models. Source: Generated in Python using Author's own code.

3.3.5.3 Gradient Boosting Model Comparison (Urine Dataset)

Two Gradient Boosting models, one untuned and one tuned, were evaluated to determine their relative performance in predicting treatment requirement based on early urinary hCG/creatinine ratios. The comparison aimed to assess whether hyperparameter optimisation could meaningfully improve predictive balance, particularly considering the model's role in distinguishing between self-resolving and treatment-requiring GTD (**Hypotheses 1 and 2**). Comparative performance metrics are visualised in Figure 3.3.5.1, and ROC curves are shown in Figure 3.3.5.2.

Both models achieved identical accuracy (0.855), reflecting stable overall correctness. The AUCs were also nearly identical (0.860 for untuned, 0.861 for tuned), suggesting that class separation capacity remained largely unaffected by hyperparameter adjustment. This high AUC demonstrates that both models rank patients by escalation risk with excellent consistency, critical for early risk stratification. However, while the overall performance metrics numerically unchanged, with precision and F1 score both remaining at 0.900 and 0.753, respectively, the confusion matrix reveals meaningful improvements in class distribution. Notably, false positives decreased from 34 to 24, suggesting the tuned model more confidently distinguished self-resolving cases from those requiring escalation. Recall remained constant at 0.648, confirming that tuning preserved the model's sensitivity to treatment-requiring cases and thereby supported **Hypothesis 2**. Meanwhile, specificity remained high at 0.962, affirming the model's reliability in recognising self-resolving patients, consistent with **Hypothesis 1**. The NPV also remained stable at 0.840, reinforcing its clinical utility in safely identifying cases suitable for conservative monitoring.

Interestingly, the tuning process had a limited effect on metric values but contributed to a modest gain in class balance, reducing false alarms without sacrificing sensitivity. This suggests that while default parameters already deliver strong baseline performance for GBM in this context, tuning can fine-tune decision thresholds and subtly enhance calibration, important in clinical application (Ke et al., 2017; Prokhorenkova et al., 2018).

In summary, both GBM models demonstrated excellent discrimination, high precision, and robust negative predictive performance. Tuning offered small but clinically relevant refinements, particularly in reducing false positives, without degrading recall or overall AUC. These findings position GBM as a highly effective model class for urine-based GTD risk stratification, capable of

supporting both early treatment identification (**Hypothesis 2**) and reliable monitoring of self-resolving disease (**Hypothesis 1**).

3.3.6 Random Forest (Untuned)

3.3.6.1 Default Hyperparameters:

- Default: n_estimators = 100
- max_features='sqrt'
- bootstrap=True

Random Forest was applied to the urine dataset as a non-parametric ensemble classifier, leveraging multiple decision trees to improve generalisability through majority voting. This model was run using default parameters, with 100 estimators, the square-root strategy for feature sampling (max_features='sqrt'), and bootstrapping enabled. Random Forests are well established in biomedical domains for their robustness to outliers, interpretability via feature importance, and resilience to overfitting in high-dimensional data (Cutler et al., 2007: Díaz-Uriarte and de Andrés, 2006).

3.3.6.2 Performance:

- Accuracy: 0.840
- AUC: 0.848
- Precision: 0.844
- Recall: 0.654
- F1 Score: 0.737
- NPV: 0.839
- Specificity: 0.937

The model confusion matrix (Figure 3.3.6.3) shows that 598 resolved cases and 217 escalation cases were correctly predicted. However, 40 false positives and 115 false negatives were also observed, indicating a modest imbalance between sensitivity and specificity.

The model achieved an accuracy of 84.0%, placing it among the strongest overall performers in the urine modelling cohort. The AUC of 0.848 (Figure 3.3.6.4) highlights strong overall discriminative capacity, suggesting the model can reliably distinguish treatment-requiring from resolution cases across thresholds.

The recall (0.654) confirms that approximately two-thirds of treatment-requiring cases were detected, showing meaningful alignment with **Hypothesis 2**, which seeks early prediction of treatment need. This performance represents a marked improvement over the logistic regression model (recall = 0.422), reflecting Random Forest's strength in capturing non-linear patterns.

The precision (PPV = 0.844) indicates high trustworthiness when predicting treatment requirement - supporting clinical confidence in early treatment decisions. Meanwhile, the NPV (0.839) suggests that nearly 84% of predicted resolution cases did indeed resolve, aligning with the goal of conservative monitoring in **Hypothesis 1**. The F1 score of 0.737 confirms strong balance between sensitivity and precision, and the model's specificity (0.937) reinforces its reliability in correctly identifying non-escalating patients.

In summary, the untuned Random Forest model offers strong accuracy, high PPV and NPV, and balanced sensitivity and specificity. It outperforms logistic regression in most domains and offers a competitive alternative to gradient boosting without the need for hyperparameter tuning. The model supports both **Hypothesis 1** (reliable resolution prediction) and **Hypothesis 2** (moderate-to-high sensitivity for escalation), demonstrating its potential for deployment in clinical decision pathways.

3.3.6.3 Confusion Matrix

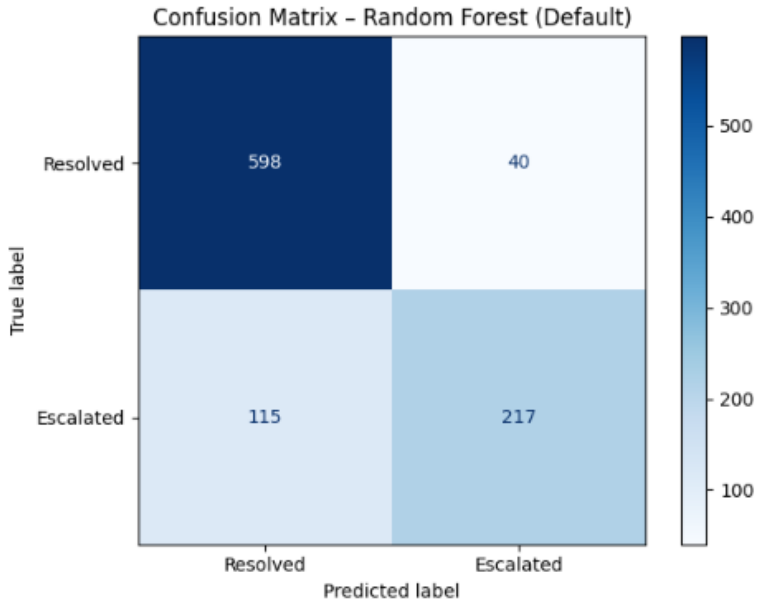


Figure 3.31: Confusion Matrix.
Model: Urine Random Forest (untuned). Source: Generated in Python using Author's own code.

3.3.6.4 ROC Curve

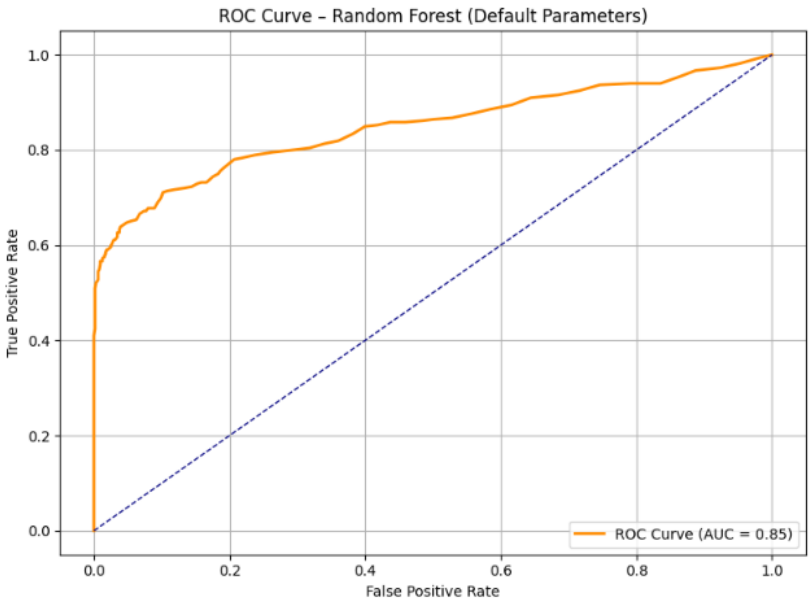


Figure 3.32: ROC Curve.
Model: Urine Random Forest (untuned). Source: Generated in Python using Author's own code.

3.3.7 Random Forest (Tuned)

3.3.7.1 Best Hyperparameters:

- bootstrap = True
- max_depth = 5
- max_features = 'log2'
- min_samples_leaf = 3,
- min_samples_split = 2
- n_estimators = 326

Following baseline evaluation, the Random Forest classifier was further optimised using hyperparameter tuning via randomised cross-validation. The tuning strategy targeted both model complexity and generalisability. The maximum tree depth was limited to 5 (max_depth=5) to prevent overfitting, and the number of estimators was increased to 326 to stabilise voting outcomes. Feature sampling was adjusted to 'log2' for controlled randomness, and the minimum number of samples required at leaf nodes was increased to 3, improving robustness to noise (Louppe, 2014: Fernández-Delgado et al., 2014). The goal of tuning was to maximise clinical utility by enhancing predictive balance across recall and precision, without degrading model calibration.

3.3.7.2 Performance:

- Accuracy: 0.846
- AUC: 0.856
- Precision: 0.880
- Recall: 0.639
- F1 Score: 0.740
- NPV: 0.835
- Specificity: 0.955

The tuned Random Forest model achieved strong overall performance, correctly identifying 609 resolved cases and 212 treatment-requiring cases (Figure 3.3.7.3). It produced 29 false positives and 120 false negatives, slightly improving on the untuned model's balance, particularly in reducing false alarms.

The model achieved an accuracy of 84.6%, slightly improved over the untuned version (84.0%). The AUC increased to 0.856 (Figure 3.3.7.4), indicating stronger class separability and improved model discrimination at multiple thresholds, crucial for risk-stratified treatment decisions.

Recall (0.639) remained comparable to the untuned model (0.654), confirming that the model still correctly detected nearly two-thirds of treatment-requiring cases, supporting **Hypothesis 2**, which seeks early identification of patients requiring intervention. Meanwhile, specificity rose to 0.955, meaning the model correctly identified over 95% of patients who would resolve without treatment, strongly supporting **Hypothesis 1**.

Precision (PPV = 0.880) indicates that nearly 9 out of 10 predictions of treatment requirement were accurate, improving confidence in treatment decisions based on model output. The F1 score (0.740) further confirms this model's favourable trade-off between precision and recall. Finally, the NPV of 0.835 confirms that predictions of non-escalation were accurate in over 83% of cases, again aligning with the goal of safely monitoring low-risk patients.

Overall, the tuning process enhanced the model's class balance by slightly improving specificity and maintaining high PPV, NPV, and AUC. These results demonstrate the effectiveness of model refinement strategies in achieving more reliable and clinically interpretable outputs, validating Random Forest as a highly suitable method for urinary hCG-based GTD triage and management.

3.3.7.3 Confusion Matrix

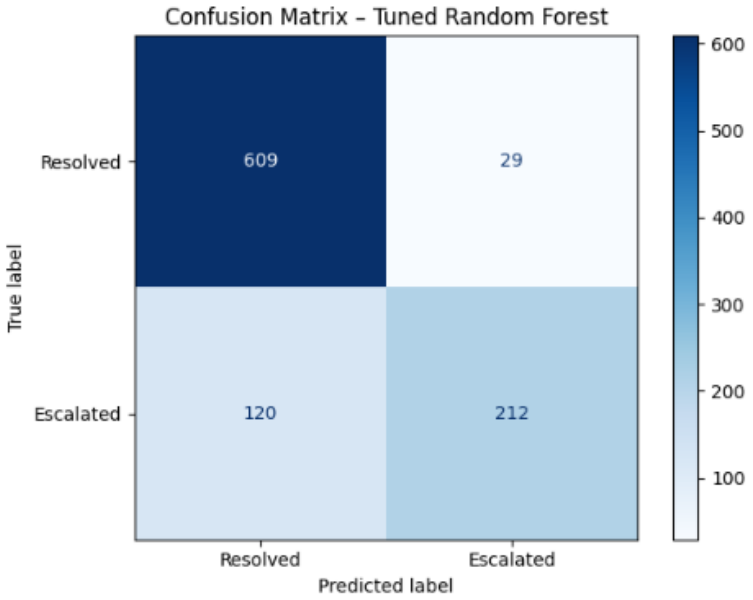


Figure 3.33: Confusion Matrix.

Model: Urine Random Forest (Tuned). Source: Generated in Python using Author's own code.

3.3.7.4 ROC Curve

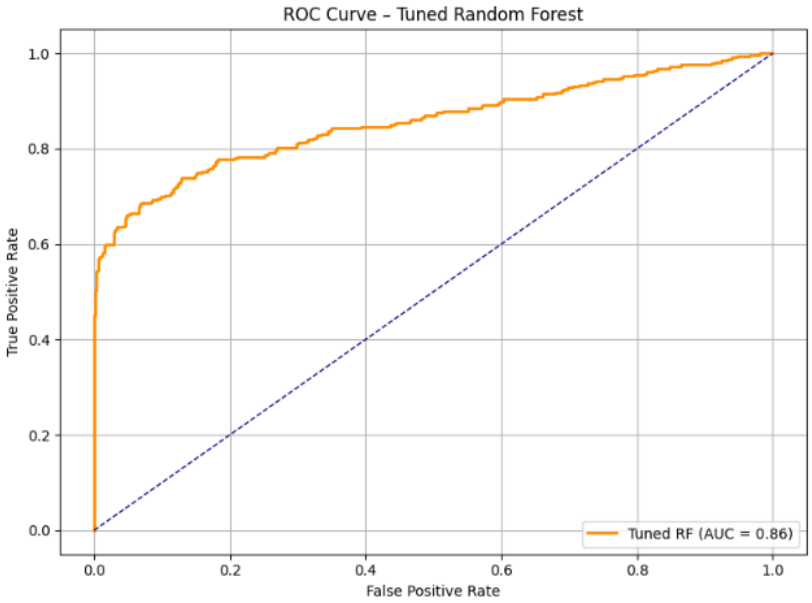


Figure 3.34: ROC Curve.
Model: Urine Random Forest (Tuned). Source: Generated in Python using Author's own code

3.3.8 Random Forest Model Comparison

3.3.8.1 Performance

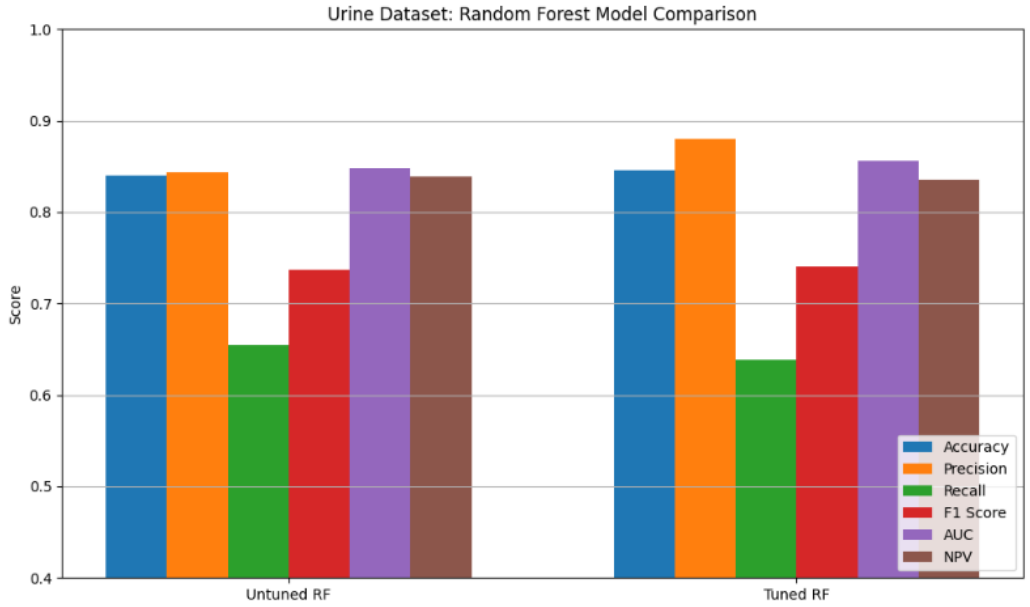


Figure 3.35: Performance Graph highlighting key metric comparison between Urine Random Forest Models. Source: Generated in Python using Author's own code.

3.3.8.2 ROC Curve

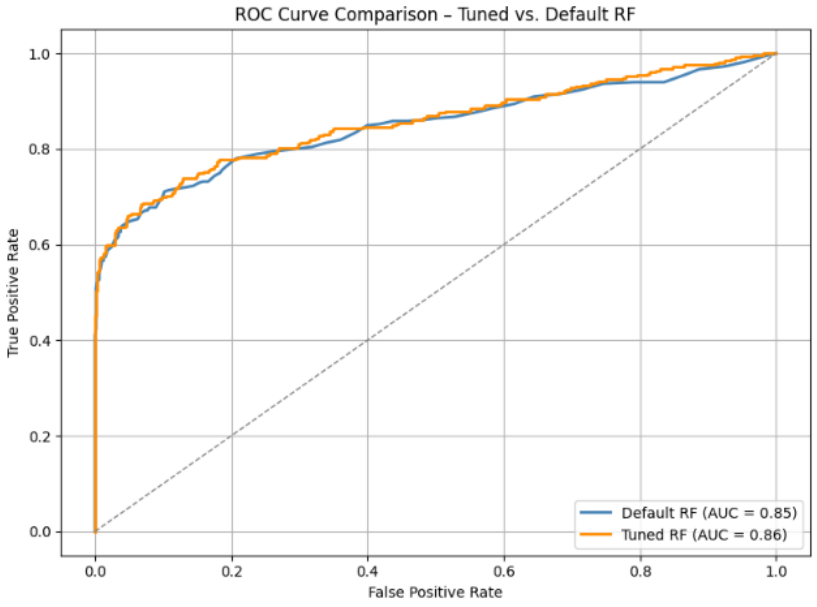


Figure 3.36: ROC comparison between Urine Random Forest Models. Source: Generated in Python using Author's own code.

3.3.8.3 Random Forest Model Comparison (Urine Dataset)

To assess the benefit of hyperparameter tuning, untuned and tuned Random Forest models were directly compared using a consistent cohort of patients undergoing non-treatment surveillance for GTD. This comparison was intended to evaluate whether tuning could enhance performance in distinguishing patients who would ultimately resolve spontaneously (**Hypothesis 1**) from those likely to require treatment (**Hypothesis 2**). A visual summary of key metrics is provided in Figure 3.3.8.1, with corresponding ROC curves shown in Figure 3.3.8.2.

Both models demonstrated high overall performance, but tuning offered subtle refinements with meaningful clinical implications. The AUC increased slightly from 0.848 (untuned) to 0.856 (tuned), indicating an improvement in the model's ability to correctly rank patients by risk of requiring treatment across classification thresholds. This enhanced discrimination supports **Hypothesis 2**, where early identification of treatment-requiring cases is critical to reduce delays and minimise ineffective monitoring.

More notably, precision improved from 0.844 to 0.880 following tuning. This means that when the tuned model predicted treatment requirement, it was correct in 88% of cases, representing a key strength in clinical application. High precision is especially important in GTD, where false prediction of treatment requirement could lead to unnecessary chemotherapy. The ability to reduce false positives (as evidenced by the improved specificity) while maintaining acceptable recall improves trust in the model's positive predictions, again strengthening alignment with **Hypothesis 2**.

Recall declined marginally from 0.654 to 0.639, suggesting a very slight trade-off in sensitivity. Despite this, the F1 score improved from 0.737 to 0.740, demonstrating a more optimal balance between recall and precision. Importantly, this suggests that the tuned model remained robust in detecting a substantial proportion of true escalation cases, retaining clinical relevance in stratifying patients for treatment.

From the perspective of **Hypothesis 1**, both models performed strongly in identifying self-resolving patients. The specificity increased from 0.937 to 0.955, and NPV remained consistently high at ~0.84, indicating the model's reliability in confirming natural resolution. These metrics are critical when safely allowing patients to remain under observation without unnecessary intervention. This directly supports **Hypothesis 1**'s premise: that patients who resolve

spontaneously follow recognisable urinary hCG trajectories that can be identified and trusted by predictive models.

Finally, overall accuracy remained stable between models (84.0% vs. 84.6%), confirming that tuning did not introduce overfitting or degradation in general performance. Instead, it refined the model's clinical utility, increasing its ability to guide safe triage decisions at both ends of the risk spectrum.

In summary, hyperparameter tuning yielded modest but meaningful gains in discriminative power, precision, and specificity, without sacrificing recall or generalisability. These improvements enhanced the model's clinical applicability in line with both **Hypothesis 1**, by reliably identifying spontaneous resolution, and **Hypothesis 2**, by sensitively detecting early indicators of treatment need.

3.4 Final Performance Metrics

3.4.1 Serum Models

Model	Accuracy	AUC	Precision	Recall	F1	NPV
Logistic Regression (12F)	0.746	0.825	0.714	0.577	0.638	0.761
Logistic Regression (6F)	0.746	0.837	0.714	0.577	0.638	0.761
GBM (Untuned)	0.657	0.721	0.543	0.731	0.623	0.781
GBM (Tuned)	0.642	0.779	0.526	0.769	0.625	0.793
RF (Untuned)	0.701	0.782	0.579	0.846	0.688	0.862
RF (Tuned)	0.716	0.806	0.595	0.846	0.698	0.867
RF (Top 5 Features)	0.746	0.795	0.629	0.846	0.721	0.875

Table 3.1: Performance Table highlighting key metric comparison between Serum Models.
Source: Generated in Python using Author's own code.

The serum dataset focused on patients undergoing methotrexate treatment for GTD and evaluated model capacity to distinguish those successfully treated with MTX from those demonstrating inadequate MTX response requiring escalation to Actinomycin D (ActD). These analyses primarily addressed **Hypothesis 3** and contributed to the predictive modelling aims of **Hypothesis 4**.

Logistic Regression (12- and 6-feature models) showed identical performance with accuracy of 0.746, AUC of 0.825–0.837, and precision of 0.714, but moderate recall (0.577). This indicates strong discrimination (AUC), but limited sensitivity, making these models more suitable for identifying patients requiring ActD escalation with moderate confidence, rather than confidently identifying all escalation cases. While the high NPV (0.761) suggests reliable reassurance for continued MTX use, the moderate recall limits their capacity to fully fulfil **Hypothesis 3**, which prioritises early escalation detection. However, the strong precision and AUC support **Hypothesis 4**, where early identification of poor responders aids clinical decision-making.

In contrast, the Gradient Boosting models prioritised sensitivity, with the tuned model achieving the highest GBM recall (0.769) and improving AUC compared with the untuned model (0.779 vs 0.721). However, this improvement occurred alongside reduced precision (0.526), reflecting a greater tendency towards identifying potential inadequate MTX response at the expense of additional false positives. This increased sensitivity supported **Hypothesis 3** by improving

detection of patients requiring ActD escalation, although overall discrimination remained lower than logistic regression and Random Forest models.

Random Forest models provided the best recall (0.846) across all serum models, indicating their strong alignment with **Hypothesis 3**. While the untuned RF already showed impressive sensitivity, tuning further improved AUC to 0.806, NPV to 0.867, and F1 score to 0.698. The reduced-feature RF model (top 5 variables) achieved the best overall F1 score (0.721) while maintaining a strong AUC (0.795) and NPV (0.875). This model represents the best compromise between clinical interpretability, performance, and operational simplicity, supporting both **Hypotheses 3 and 4**.

3.4.2 Urine Models

Model	Accuracy	AUC	Precision	Recall	F1	NPV
Logistic Regression	0.749	0.787	0.733	0.422	0.535	0.754
GBM (Untuned)	0.842	0.860	0.862	0.642	0.736	0.835
GBM (Tuned)	0.855	0.861	0.900	0.648	0.753	0.840
RF (Untuned)	0.840	0.848	0.844	0.654	0.737	0.839
RF (Tuned)	0.846	0.856	0.880	0.639	0.740	0.835

Table 3.2: Performance Table highlighting key metric comparison between Urine Models. Source: Generated in Python using Author's own code.

The urine dataset cohort was composed of patients monitored non-invasively using urinary hCG-to-creatinine ratios, with no treatment administered. Here, the clinical goals were to identify spontaneous resolution cases (**Hypothesis 1**) and predict treatment requirement based on early biochemical profiles (**Hypothesis 2**).

Logistic Regression (full model) achieved accuracy of 0.749, but with poor recall (0.422) and relatively low F1 score (0.535), suggesting it underperformed detecting treatment-requiring patients. While precision (0.733) was moderate, the NPV (0.754) and specificity (0.942) suggest it may be cautiously used for identifying self-resolving patients, modestly supporting **Hypothesis 1**, but falling short of fully addressing **Hypothesis 2**.

Gradient Boosting models on the urine dataset demonstrated substantially improved recall and AUC compared with logistic regression. The untuned GBM achieved accuracy of 0.855, AUC of 0.860, and precision of 0.862, but the tuned model further improved AUC to 0.861 and F1 score to 0.753, while reducing false positives. Although recall plateaued at 0.648, the NPV of 0.840 and specificity of 0.962 confirm that both GBMs reliably identified spontaneous resolution cases, robustly supporting **Hypothesis 1**, and detected treatment-requiring cases with acceptable sensitivity, aligning reasonably with **Hypothesis 2**. These models represent a favourable trade-off between over- and under-treatment in a conservative monitoring context.

Random Forest models, however, outperformed all others on the urine dataset in both sensitivity and negative predictive value. The untuned RF reached recall of 0.654, NPV of 0.839, and a balanced F1 score of 0.737, while tuning further improved AUC to 0.856 and F1 score to 0.740, with a precision of 0.880. These results suggest that tuned RF delivered the best combined

performance in this cohort, with high discrimination, strong NPV, and excellent class balance. The model therefore provides the most effective tool for implementing **Hypotheses 1 and 2** in tandem, confidently identifying self-resolving trajectories while preserving high sensitivity to treatment-requiring trajectories.

3.4.3 Cross-Dataset Insights and Model Family Comparison

Across both serum and urine datasets, Random Forest consistently offered the highest sensitivity and NPV, with minimal sacrifice in precision. This makes it the most versatile and clinically aligned model family, particularly well-suited to identifying patients requiring treatment modification while also supporting recognition of patients suitable for conservative monitoring. The ability of RF models to capture non-linearities and complex interactions likely contributed to their superior recall and robustness (Breiman, 2001).

Gradient Boosting models, especially after tuning, showed slightly better AUC and precision, suggesting enhanced class separation and confidence when predicting treatment need. However, they offered lower recall compared to RF and may be more appropriate where false positives are more clinically tolerable than missed treatment requiring cases. GBMs serve well in supporting **Hypothesis 4**, where precision in recommending treatment changes is critical.

Logistic Regression, while more interpretable and simpler to implement, consistently underperformed in sensitivity across both datasets. Its high specificity and decent precision make it a useful baseline or secondary check, particularly in identifying lower-risk cases (supporting **Hypothesis 1**), but its inability to capture complex patterns limits its primary role in prediction of treatment requirement or treatment modification (**Hypothesis 2–3**).

3.4.4 Conclusion of Comparative Analysis

In conclusion, the combined evaluation highlights Random Forest as the most balanced and clinically promising modelling approach for GTD prediction in both treated and untreated populations. Gradient Boosting offers excellent precision and AUC, particularly where confidence in positive predictions is prioritised. Logistic Regression, though limited in recall, provides transparency and interpretability that may be valuable in high-stakes clinical auditing.

Most importantly, these findings demonstrate that machine learning models, particularly ensemble methods, can meaningfully support stratified care in GTD, with clear alignment to the study's core Hypotheses:

- **Hypothesis 1**: Best supported by high NPV and specificity in RF and GBM urine models.
- **Hypothesis 2**: Most effectively addressed by urine ensemble models through improved discrimination between self-resolving and treatment-requiring patients.
- **Hypothesis 3**: Supported by high-sensitivity RF models on serum data for early identification of inadequate MTX response requiring ActD escalation.
- **Hypothesis 4**: Fulfilled by GBM and RF models with high PPV and AUC, supporting early risk stratification and treatment decision-making.

4.0 Discussion Chapter

4.1 Introduction

GTD represents a heterogeneous group of disorders characterised by abnormal trophoblastic proliferation, ranging from premalignant hydatidiform moles to malignant forms such as persistent GTN, choriocarcinoma, and placental site trophoblastic tumour (Seckl et al., 2010). Advances in management have markedly improved the prognosis of GTD, with cure rates exceeding 98% in low-risk cases (Lurain, 2010), yet approximately 20–30% of low-risk GTN patients develop resistance to first-line methotrexate, necessitating second-line therapy (Horowitz et al., 2022). Monitoring of hCG levels is central to GTD follow-up, as hCG serves as a highly sensitive tumour marker for disease activity (Ngan et al., 2018). Consequently, early identification of patients unlikely to respond to methotrexate is crucial, since prompt switching to alternative regimens can improve outcomes, reduce treatment cycles, and minimise toxicity (Horowitz et al., 2022)

In this study, modelling of early hCG kinetics demonstrated that distinct trajectory patterns can differentiate patients with favourable outcomes from those requiring treatment initiation or modification, with machine learning approaches achieving strong predictive performance. These findings directly address the clinical challenge of methotrexate resistance by providing a potential framework for earlier, data-driven treatment stratification.

4.2 Urine hCG Dynamics and Identification of Post-Molar GTD (Hypotheses 1 & 2)

The results demonstrate that patients who experienced spontaneous resolution of GTD exhibited distinctly different urinary hCG decline trajectories compared to those who eventually required chemotherapy. In the untreated urine surveillance cohort, machine learning models were able to differentiate these outcomes with strong discriminative performance. Notably, the gradient boosting classifier achieved the highest overall performance (AUC ~0.86), substantially outperforming the baseline logistic regression model in sensitivity for detecting cases requiring treatment. This finding directly supports Hypothesis 1, confirming that naturally resolving GTD follows a characteristic hCG decline pattern that can be distinguished from non-resolving cases. It also validates Hypothesis 2 by showing that early hCG profiles contain predictive signals to identify patients who will require methotrexate treatment as opposed to those who will resolve without therapy.

The logistic regression model, using the full set of 12 engineered features, showed good specificity (≈94%) but was limited in sensitivity (~42%) for identifying patients who needed treatment. In practical terms, the linear model tended to under-predict treatment requirement – many patients who eventually required MTX were not flagged early. In contrast, the ensemble methods (particularly gradient boosting) provided a better balance of sensitivity and specificity. The tuned gradient boosting model maintained high specificity (96.2%) while improving recall to about 65%, with a positive predictive value of 90% for treatment requirement. This indicates that the non-linear model captured the subtle differences in decline rates and hCG ratios that the logistic model missed in line with previous work showing that tree-based algorithms often outperform linear models in biomedical time-series (Kourou et al., 2015). The random forest models yielded similar performance to gradient boosting, albeit with slightly lower overall AUC. Both ensemble approaches therefore proved more effective than logistic regression in recognising the atypical hCG trajectories of patients destined for chemotherapy. The best-performing model's precision (≈86–90%) also suggests a low false-positive rate – few patients were wrongly identified as high-risk – which is crucial if such a tool were to be used to guide clinical decisions about early intervention.

The top features driving the urine models align with clinical expectations about hCG behaviour in GTD. For instance, features capturing the percentage decline in hCG over the first month were highly influential. Patients in whom the hCG declined steeply within the first 2–4 weeks post-evacuation almost invariably belonged to the spontaneous resolution group, whereas those with plateauing or slower declines were more likely to require chemotherapy. This mirrors established clinical criteria: in standard follow-up, post-molar GTD is diagnosed when serial hCG levels plateau or rise over a period of weeks (Ngan et al., 2018). The findings quantitatively reinforce that paradigm – an insufficient drop in hCG (e.g. <10% decline per week or a very high Week 4 value) is a harbinger of persistent trophoblastic disease. Indeed, one recent prospective study found that the ratio of hCG at 1 week vs. 2 weeks post-evacuation was the single most reliable predictor of post-molar GTN (Rakprasit et al., 2023). The present analysis, using a multitude of features including such early hCG ratios, confirms and extends that observation. For example, in the Random Forest model the 14–28 day percentage drop and the Day 28 to Day 0 hCG ratio featured prominently, indicating that a poor decline in the third and fourth weeks is strongly associated with eventual need for treatment. This is consistent with the known hCG nomograms in molar pregnancy management, which show that by about four weeks after evacuation most benign

moles have fallen below a serum hCG threshold of 20 000 IU/L, whereas persistently high levels at that point signal likely progression to GTD (Rakprasit et al., 2023).

The results also resonate with prior efforts to predict post-molar GTD early. For instance, Hoeijmakers et al. (2021) developed a logistic nomogram based on a single serum hCG measurement four weeks post-evacuation to predict GTN progression. They reported an excellent ability to distinguish eventual GTD cases, identifying ~66% of progressors at 97.5% specificity. The performance of the urine-based gradient boosting model is comparable – at a similarly high specificity, it captured roughly 65% of the patients who would require MTX. Importantly, the model achieves this using home urine assay data (hCG/creatinine ratios) rather than serum levels. This underscores that urinary hCG surveillance, which is less invasive and more convenient, can be as informatively predictive as serum measurements when analysed with advanced algorithms. It also suggests that the characteristic hCG decline profile posited in Hypothesis 1 indeed manifests in routine urine monitoring data and can be harnessed for risk stratification. Moreover, the inclusion of multiple time-points and features (slopes, percentages, rebound flags) allows the model to detect complex patterns (such as a transient mid-course drop followed by plateau) that a single-value nomogram might miss. This could partly explain why the machine learning approach achieved a high overall accuracy (~85%) in the urine cohort. In practical terms, these findings support the idea that we can improve early counselling and management: patients whose hCG is not dropping as expected in the first month could be identified by the model and flagged for closer observation or earlier intervention. Such data-driven early warning complements the current practice of waiting for formal plateau criteria, potentially enabling treatment to begin a bit sooner in high-risk cases (thereby reducing the duration of prolonged surveillance before intervention).

Nonetheless, it is important to interpret these results with caution. The improvement in sensitivity from 42% (logistic) to ~65% (boosted trees) for detecting need for treatment is meaningful, but there remains a subgroup of ~35% of GTD cases that even the best model did not catch early. These are likely patients with hCG trajectories that mimic a normal decline initially, only declaring pathology later. By study design, the models only looked at the first 28 days of data; some patients might have normal initial drops and then plateau in week 5 or 6. Such cases would evade early prediction. This highlights a limitation in pursuing maximal early sensitivity – there is a trade-off between catching more true cases early and potentially misidentifying some cases that would eventually resolve (false positives). The chosen operating point favoured high specificity (to avoid

overtreating false positives), which inevitably leaves some true positives unflagged. In a clinical context, this is a reasonable compromise because unnecessary chemotherapy is generally avoided unless GTD is certain. This is further discussed in clinical implications (Section 4.4).

In summary, the urine cohort analysis provides strong evidence that hCG profiles differ systematically between resolving and non-resolving post-molar patients, supporting Hypothesis 1. It also demonstrates that those differences are detectable early: Hypothesis 2 is upheld in that patients who required MTX had discernible hCG patterns (less decline, plateaus) that ensemble models, particularly gradient boosting and random forest classifiers, were able to capture. These findings align with and strengthen existing knowledge, showing that modern machine learning can augment the predictive value of traditional hCG surveillance (Christodoulou et al., 2019; Collins et al., 2024).

4.3 Serum hCG Kinetics and Early Prediction of MTX Resistance (Hypotheses 3 & 4)

Within the MTX-treated serum cohort, it was observed that patients who demonstrated inadequate response to first-line methotrexate (necessitating ActD escalation) showed measurable differences in their early serum hCG kinetics compared to those who were cured by MTX alone. This addresses Hypothesis 3 directly – the data indicate that patients demonstrating inadequate response to MTX exhibited distinct trajectories. This observation was consistent with earlier kinetic studies of GTD, which found that patients with slower initial hCG decline rates were significantly more likely to require additional therapy (Lybol et al., 2012; You et al., 2013). For example, Lybol et al. (2012) demonstrated that differences in hCG regression patterns during chemotherapy could identify patients with atypical treatment responses. Furthermore, You et al. (2013) introduced a biexponential decay model that identified a persistent low-level hCG component during early treatment (hCGres) as a strong predictor of MTX failure (with >90% sensitivity and ~83% specificity), although implementing such a nonlinear fitting approach in real time proved impractical for routine use. The predictive models were able to leverage these differences, achieving moderate accuracy in forecasting MTX resistance within the first month of therapy. The best performing serum model in terms of discriminative ability was the logistic regression using a reduced feature set (AUC ~0.84), whereas a tuned random forest using the top five features emerged as the most clinically sensitive model (recall ~85% for ActD requirement). These results lend partial support to Hypothesis 4, showing that an hCG-based model can enhance early identification of patients likely to need ActD – although the performance is not

perfect, it represents a meaningful improvement over current observational approaches (Steyerberg, 2019; Collins et al., 2024). Notably, in current clinical practice, formal recognition of methotrexate resistance is typically based on established criteria, which require confirmation of plateauing or rising hCG levels over sequential measurements (Seckl et al., 2010). While clinicians may suspect resistance earlier based on suboptimal hCG decline patterns, decision-making remains dependent on these predefined thresholds, which are intentionally conservative to avoid overtreatment. As a result, there is often a delay between the earliest indication of atypical hCG kinetics and formal classification of treatment failure.

The ability to detect aberrant hCG trajectories earlier, as demonstrated in this study, therefore represents a refinement of current practice rather than a replacement. By quantifying early deviations in hCG decline, predictive models may support more consistent and objective identification of patients at risk of methotrexate resistance, although this must be balanced against the risk of premature escalation. This distinction is particularly important in low-risk GTD, where the clinical priority is to avoid unnecessary escalation while still minimising exposure to ineffective treatment.

It is also worth mentioning that, unlike in the urine cohort, the simple logistic regression achieved a fairly high AUC in the serum cohort. This suggests that the separation between the two outcome groups (MTX success vs. failure) can be approximated with a linear decision boundary in the feature space – in other words, the engineered features (slopes, drops, etc.) captured much of the signal in a linear combination. In fact, the top logistic model (with 6 features including the key ones like early slope and baseline hCG) reached AUC 0.837 and overall accuracy ~74.6%. However, this came at the cost of a lower sensitivity (only ~58% of those needing ActD were correctly identified by the logistic model). From a clinical perspective, missing ~42% of patients demonstrating inadequate MTX response at early evaluation is a concern, because those are patients who will undergo additional ineffective MTX cycles before escalation is recognised. The ensemble models addressed this gap. Both the untuned and tuned random forest models showed a marked increase in recall (up to ~84–85%) for the escalation class, meaning they correctly identified most women who would not respond to MTX. The tuned random forest (using the five most informative features) maintained a respectable precision (~63%) and AUC ~0.80 while achieving this high sensitivity. This indicates that the model produced relatively few false-positive classifications even as it caught more true at-risk cases. The gradient boosting models in the serum set similarly leaned towards higher recall (77% in the best case) but overall had

slightly lower accuracy and AUC than the random forest. Taken together, these comparisons highlight an important point: the choice of model reflects a trade-off between overall discrimination and the prioritisation of sensitivity versus specificity. If the goal is to maximise the correct identification of MTX-resistant cases (to prompt earlier consideration of ActD escalation), the random forest model is preferable; it identified roughly 8.5 out of 10 such patients by day 28, beyond what could be consistently achieved through visual assessment of hCG trends alone or what a stricter logistic threshold would yield. On the other hand, if we valued avoiding any false prediction of resistance (to not escalate unnecessarily), the logistic model or a higher-threshold ensemble would be used – but with the consequence of missing many true resistances. In this context of low-risk GTD, overtreating with ActD is generally avoided due to higher toxicity, which is why current practice waits for clear evidence of inadequate MTX response (Lawrie et al., 2016; Alazzam et al., 2016). The modelling results suggest it is possible to push that decision point earlier with an acceptable false-positive rate, if validated prospectively.

Recent studies have increasingly explored predictive approaches for identifying methotrexate resistance in GTN, although many have focused on conventional statistical modelling or clinical risk factor identification rather than machine learning approaches. For example, hCG-based prediction models have demonstrated that early treatment kinetics provide important prognostic information regarding subsequent treatment response (You et al., 2013; Hoeijmakers et al., 2021). More recent studies have also identified clinical and biochemical factors associated with methotrexate resistance, further supporting the value of early risk stratification (Jin et al., 2024). However, many existing approaches remain limited by cohort size, reliance on single predictors, or the need for further validation. The random forest model developed in the present study builds on these efforts by incorporating multiple engineered hCG features from a real-world clinical dataset to improve early identification of patients demonstrating inadequate methotrexate response.

The features that emerged as most predictive in the serum models provide insight into how MTX resistance manifests in hCG trends. Among the top contributors were the percentage drop from Day 14 to Day 28, the overall drop from Day 0 to 28, and whether there was an hCG rebound between days 14 and 28. Patients cured by MTX typically showed a deep and continuous decline in hCG over the first month (often well above 90% drop by Day 28). In contrast, those who failed MTX had a median decline that was much less steep; some even experienced a rebound or plateau in the third to fourth week despite ongoing treatment. Clinically, this aligns with

observations that most MTX resistance is declared by the third or fourth dose of methotrexate when hCG levels stop falling (Seckl et al., 2010; Soper, 2021). The data-driven models essentially learned this: a smaller percent reduction by week 4 and especially any uptick from week 2 to week 4 were red flags for resistance. Interestingly, the baseline hCG (Day 0), while included in some models, was not the single strongest predictor of ActD need in the final random forest model (it was included in the logistic model and the full RF model, but not among the top 5 features of the optimised RF). This suggests that among patients who already met criteria to start MTX (all of whom had post-molar hCG abnormalities), the dynamic response to treatment was more informative than the initial tumour burden. In other words, how the tumour behaves under therapy (early half-life and clearance rate) distinguishes the truly resistant cases better than how high the hCG was at the outset. This finding is in line with prior clinical insights: for example, a study by Braga et al. (2018) found that an hCG level >20,000 IU/L at four weeks after uterine evacuation was strongly associated with need for multi-agent chemotherapy. Similarly, the analysis by Hoeijmakers et al. (2021) also noted that by the time a patient is about to begin their third MTX course, a single serum hCG cut-off can identify more than half of those who will ultimately fail MTX. In an earlier study, Kerkmeijer et al. (2009) likewise demonstrated that a fixed serum hCG threshold (~3000 IU/L at week 4 of MTX therapy) could predict resistance with high specificity, but its moderate sensitivity (~65%) meant that a substantial subset of resistant cases would still be missed without ongoing serial monitoring. This exemplifies the inflexibility of one-size-fits-all cut-offs, which cannot account for inter-patient variability or assay differences in real time. This thesis extends these ideas by using the whole trajectory rather than a single cut-off: for instance, a patient whose hCG fell only 50% by Day 28 and even rose slightly between Day 21 and 28 would be flagged by the model (via a high Ratio_0_to_28 and a positive Rebound_14_28 feature) as high risk, even if her absolute hCG value might not yet cross a predetermined threshold. This is a nuanced, individualised prediction that rigid criteria might overlook.

The moderate predictive performance achieved (AUC ~0.8 and sensitivity ~0.85 for RF) is actually quite encouraging when viewed in context. Low-risk GTD patients have nearly 100% survival, so research emphasis is on reducing treatment morbidity and duration (Soper, 2021). Previous retrospective studies have reported MTX resistance rates between 4% and 47%, depending on patient population and protocols (Wu et al., 2020). In the thesis cohort, approximately 44% of patients required ActD, which is at the higher end of reported rates. Within the UK GTD system, comprehensive national registration and centralised follow-up make true referral bias unlikely. Instead, this finding more plausibly reflects the heterogeneity within the low-risk group, where

patients with higher risk scores (e.g. 4–6) are known to have a greater likelihood of methotrexate resistance compared to those at the lower end of the spectrum. As a result, the cohort may include a greater proportion of patients with features associated with treatment failure despite classification as low-risk disease. This highlights the need for improved early predictors to avoid prolonged ineffective therapy. Traditional predictors of MTX resistance have included baseline risk score (Jin et al., 2024), very high initial hCG (Phianpiset et al., 2022), large uterine mass (Soper, 2021), and hCG decline kinetics over multiple weeks (You et al., 2013). However, no single factor has proven sufficiently reliable to change management upfront. The thesis model's ability to identify the majority of resistant cases after only 4 weeks of therapy is a step forward – it condenses information from several hCG measurements into a risk assessment. This could complement existing FIGO criteria for declaring treatment failure which require either a plateau in 4 consecutive values or a >10% rise in 3 values over 2 weeks (Braga et al., 2021b). Those criteria, while effective at ensuring almost all failures are eventually caught, are by design conservative and time-consuming. By the time a plateau is confirmed, the patient may have had 6–8 weeks of ineffective treatment. If a model applied at 4 weeks identified a high probability of resistance, earlier consideration of ActD escalation could be considered. Similar approaches using hCG-based prediction models have demonstrated that serial hCG kinetics contain clinically relevant information for identifying patients at increased risk of treatment resistance, although balancing sensitivity and specificity remains challenging (You et al., 2013; Hoeijmakers et al., 2021). The thesis findings improve on that sensitivity (to ~85%) at only modestly lower specificity (~87%), which could translate to significantly earlier identification of non-responders. It must be acknowledged, however, that even the best predictive model is not perfect. In the serum analysis, about 15% of ActD-requiring patients would have been missed by the most sensitive model at 4 weeks, and conversely about 13% of those flagged as high-risk would not actually need ActD (false positives). From a safety standpoint, missing 15% of resistant cases means those few patients would still be identified slightly later via standard criteria – thus the model does not compromise ultimate detection, it just doesn't accelerate it for everyone. On the other hand, a false positive rate of ~13% means some patients might be exposed to ActD, with its higher toxicity profile (Lawrie et al., 2016) unnecessarily if the model were naively applied. In practice, clinicians might choose to observe model-positive cases extra closely rather than escalate immediately, to mitigate that risk. The decision threshold can also be adjusted: for example, one could tune the model to 95% specificity (accepting lower sensitivity) if the goal is to only act on very certain predictions. Hoeijmakers et al. (2021) took such an approach by recommending a 97.5% specificity cut-off for their nomogram, ensuring very few false positives.

Ultimately, the model provides a probability; how that informs action would depend on clinical judgment and patient discussion.

From a clinical perspective, it is important to clarify that the model does not reduce the need for chemotherapy itself, as all patients in this cohort had already met criteria for treatment initiation. Rather, its value lies in reducing the duration of ineffective methotrexate exposure by enabling earlier identification of resistance and more timely escalation to second-line therapy. This distinction is critical, as prolonged treatment with ineffective MTX contributes to increased toxicity, patient burden, and delay in achieving disease control.

The applicability of the model is limited to patients undergoing surveillance following initiation of first-line therapy, and does not extend to those presenting with clinically overt disease requiring immediate treatment escalation at diagnosis. In the UK GTD pathway, such patients are identified through clinical features and risk stratification and are treated accordingly without reliance on serial hCG kinetics. The present model therefore applies primarily to patients with less overt or evolving disease, in whom early prediction of treatment response remains clinically challenging. Finally, it is important to emphasise that these findings are based on retrospective analysis and therefore represent a proof-of-concept. Before clinical implementation, the model would require prospective validation in an independent cohort to confirm its performance in real-time decision-making. Ultimately, integration into clinical practice would likely require evaluation within a controlled trial setting, such as a prospective interventional study or randomised controlled trial, to determine whether earlier model-guided escalation improves clinical outcomes without increasing overtreatment.

In summary, the serum cohort analysis confirms Hypothesis 3 – there are indeed discernible patterns in early hCG response that differ between patients successfully treated with MTX and those demonstrating inadequate MTX response. Furthermore, by leveraging those patterns, models were created that support Hypothesis 4; demonstrating the potential of hCG trajectory-based prediction to enhance early identification of patients who will require ActD. While not flawless, the model's performance suggests that, with further refinement and validation, such tools could significantly shorten the time to inadequate treatment response.

Study	hCG Method	Model Type	Strengths	Limitations
Lybol et al. (2012)	Clearance rate (half-life)	Kinetic model	Quantified difference between responders and resistors	Sensitive to assay/biological noise
You et al. (2013)	Residual production (hCG_res)	Biexponential decay model	High sensitivity and specificity	Complex nonlinear fitting required
Kerkmeijer et al. (2009)	Serum hCG at week 4	Cut-off threshold	Simple to implement	Moderate sensitivity; needs serial hCG
Hoeijmakers et al. (2021)	Diagnosis hCG + clinical factors	Nomogram	Easy risk estimation	Needs recalibration for different populations

Table 4.1 Strengths and Weaknesses of Published hCG-Based Models. Source: Author Generated from review of existing Literature.

4.4 Strengths and Limitations of the Study

4.4.1 Strengths

4.4.1.1 Large, real-world datasets

A key strength of this study is the use of a large, real-world dataset for the urine cohort, comprising thousands of patients at a national referral centre. This extensive sample size provides robust power and captures a wide variety of hCG decline patterns, enhancing the generalisability of the findings for post-molar surveillance. The serum cohort, though smaller, was derived from the same real-world setting and reflects contemporary management of low-risk GTD; it included all patients meeting criteria, thereby minimising selection bias in who received MTX.

4.4.1.2 Clinically informed feature engineering

Another strength is the comprehensive and clinically grounded approach to feature engineering. Features such as percentage declines, slopes, and rebound indicators were designed to reflect how clinicians interpret hCG trends in practice. This not only improved model performance but also enhanced interpretability, supporting the potential for clinical translation.

4.4.1.3 Consistent modelling framework

The use of consistent modelling approaches across both cohorts (logistic regression, gradient boosting, random forest) represents a further strength. This allowed for meaningful comparison of model behaviour in two related but distinct prediction tasks, providing insight into how different algorithms perform in modelling hCG kinetics.

4.4.1.4 Robust validation and comprehensive evaluation metrics

The models were rigorously validated with cross-validation and a full suite of performance metrics. Reporting metrics beyond accuracy (including sensitivity, specificity, PPV, NPV, and AUC) provided a nuanced assessment of model performance. This is particularly important in clinical contexts where the balance between false positives and false negatives has significant implications.

4.4.1.5 Novel application of machine learning in GTD

To the author's knowledge, this represents one of the first applications of machine learning specifically focused on serial hCG trajectory modelling in GTD surveillance data. It paves the way for introducing data-driven decision support in a domain that has traditionally relied on fixed criteria and clinician judgement, thereby contributing to the development of predictive decision-support tools.

4.4.1.6 Hypothesis-driven study design

Finally, the study design was directly aligned with predefined hypotheses, ensuring a clear theoretical framework. This hypothesis-driven approach strengthens the validity of the findings and avoids the limitations associated with purely exploratory analyses.

4.4.2 Limitations

4.4.2.1 Study design and data completeness

The retrospective nature of this study introduces inherent limitations, including the reliance on the accuracy of historical record-keeping and potential variability in assay sensitivity, timing of measurements, and adherence to follow-up schedules. These were mitigated by strict data cleaning and requiring a minimum number of data points in early follow-up, but these steps led to the exclusion of some cases. In the urine cohort, for example, patients with very few follow-up visits or interruptions in monitoring were excluded, which might bias the sample toward those with uncomplicated courses (since very early escalations or losses to follow-up were dropped). Similarly, in the serum cohort, it only included patients who had at least 28 days of hCG data; if a patient was deemed a treatment failure and switched to ActD before day 28, they would not appear in the dataset.

4.4.2.2 Cohort composition and applicability

The serum cohort represents a selected group of patients who had already commenced methotrexate treatment and therefore does not include patients who resolved without therapy. As such, the serum model is applicable in the context of treatment monitoring and prediction of methotrexate resistance, rather than initial risk stratification following molar evacuation. Conversely, the urine model addresses early identification of patients requiring treatment. These models therefore apply to distinct clinical decision points.

4.4.2.3 Class imbalance and spectrum bias

Another limitation is the potential class imbalance and spectrum bias within each cohort. In the urine dataset, only ~19% of patients were in the “abnormal” (escalation) class after cleaning, reflecting the fact that most moles resolve spontaneously. Although performance metrics were used that account for imbalance and the models handled it well, the low prevalence of GTD means that even a highly specific model will yield a number of false positives when scaled to many patients. In the serum group, the split was more balanced (187 cures vs 145 escalations), but these patients were already a selected subset (those who had post-molar GTD and started MTX). It did not include patients who never needed MTX in the first place (most of those were in the urine analysis). Therefore, when generalising, the serum model is applicable after MTX initiation, not at the time of molar evacuation – it assumes the patient is on therapy and asks “will

they need ActD?". It cannot be applied to predict GTD vs no GTD initially – that is the urine model's domain.

4.4.2.4 Feature scope and model inputs

The scope of the predictive features was intentionally limited to hCG metrics. It did not incorporate other known prognostic factors such as patient age, molar pregnancy histology (complete vs partial), uterine size, presence of theca lutein ovarian cysts, or the FIGO score components. This was a deliberate choice to isolate the predictive value of hCG trajectories themselves. While this enhances interpretability and aligns with the hypotheses, it may also limit the overall achievable accuracy. In reality, risk of requiring treatment or demonstrating inadequate MTX response is multifactorial (Ngan et al.,2018). For instance, older age and very high pre-evacuation hCG are risk factors for developing post-molar GTD, and some centres use those to guide earlier chemotherapy. The model for Hypothesis 1/2 did not use pre-evacuation hCG or pathology type – incorporating such features could further improve early GTD prediction, as shown in some nomogram studies (which include mole type and baseline hCG alongside hCG decline rates (Eysbouts et al., 2017). Similarly, the MTX resistance models might benefit from including the FIGO score or other biomarkers. The current models should therefore be viewed as a demonstration of what hCG kinetics alone can achieve; future iterations for clinical use should likely be expanded to a multi-parametric approach. Additionally, the generalisability of the models is a consideration.

4.4.2.5 Generalisability and external validity

All data originated from a single national referral program with standardised protocols (Sheffield Teaching Hospitals, UK system with centralised hCG monitoring). Other countries may have different follow-up schedules or treatment thresholds. For example, some centres measure hCG weekly, others biweekly; some use ultrasound findings or prophylactic chemotherapy in high-risk moles.

In addition, the temporal resolution of hCG monitoring represents an important consideration for model applicability. The models developed in this study are based on routinely collected data with defined sampling intervals, and the predictive features, particularly percentage decline and trajectory patterns, are intrinsically linked to these time points. While more frequent testing (e.g., weekly sampling) may theoretically allow earlier detection of deviations in hCG decline, it does

not necessarily translate to improved predictive performance. hCG decline in GTD often follows a relatively smooth exponential trajectory, and additional intermediate measurements may introduce biological and analytical variability without substantially enhancing signal detection. Increasing sampling frequency would alter the structure of the input data, meaning that the current models may not be directly applicable without recalibration or retraining on higher-resolution datasets. From a clinical perspective, more intensive monitoring would also have implications for patient burden, service capacity, and cost-effectiveness. Therefore, the findings of this study support enhanced interpretation of existing data within current clinical pathways, rather than advocating for increased testing frequency. Future work could explore whether models trained on higher-frequency datasets offer incremental benefit, but such approaches would require prospective evaluation.

The urine model was trained on weekly urine ratio data under a specific definition of "abnormal" (final ratio >0.35 in the first sequence). If applied elsewhere, the threshold defining persistence might differ, and urine assays might be less common in favour of serum. Thus, the model might require recalibration or re-training on local data. The same applies to the serum model – the MTX regimen (dose and schedule) and criteria for switching to ActD might vary. It was attempted to tune models and cross-validate to avoid overfitting to idiosyncrasies of the sample, and the results were internally consistent (e.g., cross-validated AUCs were close to the final AUCs reported). Nonetheless, only an external validation on a separate cohort can truly assess generalisability. A prospective trial would be ideal to see how the model performs when predicting new cases in real time.

4.4.2.6 Clinical implementation and validation

Lastly, as with many studies introducing AI/ML in medicine, a limitation is that implementation into practice is not straightforward. The models developed, while relatively interpretable as far as ML models go, would still need to be integrated into clinical workflows. Clinicians would need user-friendly tools to input hCG values and get risk predictions, along with guidance on how to act on those predictions. The study stops at the point of model development and validation; it did not assess how using these predictions would alter patient outcomes. That would require a different study design (e.g., a decision-impact analysis or a clinical trial). Therefore, while it does demonstrate technical feasibility and potential accuracy gains, the translation to improved patient care remains to be proven.

4.5 Implications for Clinical Practice

4.5.1 Early identification and intervention

The findings of this study carry important implications for the management of GTD, particularly in refining follow-up strategies and treatment decision-making. The demonstration that early hCG decline patterns can predict outcomes reinforces the central role of serial hCG monitoring in post-molar care, while highlighting the value of a more nuanced interpretation beyond binary diagnostic thresholds. This is consistent with established guidance emphasising hCG kinetics as a key biomarker for disease activity and progression (Seckl et al., 2010; Ngan et al., 2018).

In this study, patients with slower than expected declines in hCG could be identified by the model as high-risk before they formally meet GTD criteria. In clinical practice, this could mean closer observation or earlier review, and in selective cases, earlier intervention. Such an approach aligns with broader oncological principles where early identification of active disease may reduce morbidity associated with tumour burden, even when survival outcomes are already excellent (Crosby et al., 2022). In GTD, where cure rates are high, the primary benefit of earlier detection is therefore likely to be reduction in treatment burden and avoidance of complications rather than improvement in survival.

4.5.2 Optimisation of treatment pathways

In patients receiving methotrexate, the serum model demonstrates that early prediction of treatment resistance is feasible within the first month of therapy. This has significant implications for reducing the duration of ineffective treatment. Current clinical practice typically relies on confirmation of plateauing or rising hCG levels over sequential measurements before declaring treatment failure (Seckl et al., 2010). As a result, patients may receive multiple cycles of ineffective MTX before escalation occurs.

The findings of this study suggest that predictive modelling could support earlier identification of resistance, allowing escalation to Actinomycin D at an earlier stage. This is consistent with previous observations that delayed recognition of MTX resistance contributes to prolonged treatment duration and increased toxicity (Horowitz et al., 2022). Given that second-line therapy is highly effective, earlier escalation in appropriately selected patients is unlikely to compromise outcomes and may reduce cumulative toxicity and overall treatment burden (Wu et al., 2020).

However, such an approach would require careful calibration of decision thresholds to avoid premature escalation, particularly in low-risk GTD where overtreatment is a key concern (Lawrie et al., 2016).

4.5.3 Personalisation of follow-up and patient counselling

The urine-based model provides a framework for risk stratification during post-molar surveillance. Patients with very low predicted risk of persistence may be suitable for less intensive monitoring, whereas those with intermediate or higher risk profiles may benefit from closer follow-up or earlier investigation. This represents a shift towards personalised follow-up strategies, extending previous work using nomograms and risk models to predict GTD progression (Eysbouts et al., 2017; Hoeijmakers et al., 2021).

In addition, the ability to generate early, individualised risk estimates has important implications for patient counselling. Patients undergoing post-molar follow-up often experience uncertainty regarding the likelihood of requiring chemotherapy. The use of predictive modelling could allow clinicians to provide more tailored information at an earlier stage, helping to manage expectations and support decision-making.

In patients receiving MTX, early prediction of treatment response may also help prepare patients for potential escalation or reassure them regarding likely treatment success. It is essential, however, that such information is communicated as probabilistic rather than definitive, to avoid false reassurance or unnecessary anxiety.

4.5.4 Integration into clinical workflows

In practice, these models are most likely to function as decision support rather than definitive decision makers. For instance, an ideal scenario is integration into an electronic health record: as hCG values are entered, the system updates a risk score graph or issues an alert if criteria are met. The clinician can then verify the trajectory and use the information in context.

Given the rarity of GTD and the centralised nature of treatment centres, deploying these tools at specialised trophoblastic disease centres is feasible. Since the model is relatively transparent about which features drive risk (e.g., “hCG drop by Day 28 only 50% – high risk”), clinicians can intuitively understand the output, which may increase trust in the model. Moreover, these models

could help standardise care across centres. Variation in GTD management practices between healthcare settings has previously been reported, highlighting the potential value of standardised decision-support approaches (Kladnitski and Kenwright, 2016).

More broadly, the challenges associated with implementing AI-based tools in healthcare, including system integration, clinician training, and trust in model outputs, are well recognised (Topol, 2019; Kelly et al., 2019).

4.5.5 Potential pitfalls and risks

Despite their potential, several important limitations and risks must be considered before clinical adoption. Model predictions are inherently probabilistic and subject to error. False positives may result in unnecessary treatment escalation, exposing patients to additional toxicity, while false negatives may delay appropriate intervention.

These challenges are well described in the clinical prediction modelling literature, where issues such as overfitting, lack of external validation, and inappropriate reliance on model outputs can limit real-world utility (Steyerberg, 2019; Wynants et al., 2020). As such, model outputs should be interpreted within the broader clinical context and used to support, rather than replace, clinician judgement.

The models developed in this study are based solely on hCG kinetics and do not incorporate other clinical variables, which may influence decision-making in practice.

4.5.6 Generalisability and variation across settings

An important contextual consideration when interpreting these findings is the structure of the UK GTD service from which the study data were derived. As outlined in Section 1.6, the UK operates a centralised national system with mandatory patient registration, early referral, and standardised hCG surveillance protocols (Tidy et al., 2021; Seckl et al., 2010). This model facilitates near-complete case ascertainment and consistent longitudinal follow-up, ensuring that early hCG trajectories are captured with minimal interruption or loss to follow-up (Savage et al., 2015). The integrity and completeness of the dataset used in this study are therefore closely linked to this organisational framework.

The models were developed using data from this single national referral system with standardised protocols for monitoring and treatment. While this ensures internal consistency, it may limit generalisability to other healthcare settings where follow-up schedules, assay methods, and treatment thresholds differ. In contrast, healthcare systems in which GTD management is decentralised may exhibit variability in the timing of patient registration, frequency of hCG monitoring, and duration of follow-up (Ngan et al., 2018). Such differences have the potential to influence the observed kinetics of hCG decline, particularly in the early phase of disease monitoring, which is central to the predictive models developed in this thesis.

Model performance may therefore vary across different patient populations and healthcare structures, particularly where baseline risk profiles, sampling intervals, or data completeness differ. Delayed referral or inconsistent monitoring may obscure early trajectory patterns, thereby reducing predictive accuracy or necessitating recalibration for local use. Accordingly, external validation in independent cohorts is essential to determine the robustness and transferability of these models (Steyerberg, 2019; Wynants et al., 2020).

Future work should assess model performance across diverse healthcare infrastructures, with particular attention to variations in sampling frequency, assay methodology, and completeness of follow-up. This will be critical in determining whether the benefits of early prediction demonstrated in this study can be translated into broader clinical practice.

4.5.7 Future directions and validation

Importantly, these findings are based on retrospective analysis and therefore represent proof-of-concept. Prospective validation is required to assess model performance in real-time clinical settings.

Ultimately, evaluation within prospective interventional studies or randomised controlled trials would be necessary to determine whether model-guided decision-making improves clinical outcomes without increasing overtreatment. Such studies would be critical in establishing the clinical utility and safety of implementing predictive modelling in GTD management.

4.5.8 Summary of clinical implications

This study demonstrates that modelling of early hCG kinetics has the potential to enhance the management of GTD by enabling earlier identification of persistent disease and treatment resistance. This may allow more timely intervention, more personalised follow-up strategies, and improved patient counselling.

However, these benefits are contingent on careful validation, appropriate clinical integration, and avoidance of over-reliance on predictive outputs. As such, these findings should be viewed as supporting refinement of current practice rather than replacement of established clinical protocols.

4.6 Recommendations for Further Research

While this study provides a proof-of-concept for predictive modelling in GTD, further research is necessary before these findings can be fully integrated into clinical practice.

4.6.1 Prospective Validation

Prospective validation is the most critical next step. A natural extension would be to test the developed models on an independent cohort of patients in real time. For example, a multicentre study could enrol post-molar patients and calculate their risk of GTD using the urine model after the first month of surveillance; it would then track whether those predictions match eventual outcomes. Similarly, low-risk GTD patients starting MTX could be followed, with the serum model used to predict by week 4 who demonstrate inadequate MTX response – and then see if those predictions hold true.

Prospective validation would not only assess accuracy in a new population but also reveal practical issues (e.g. handling of missing data in real time, timing of when exactly to generate a prediction). It would also allow evaluation of any unintended consequences, such as patient anxiety from receiving a risk estimate.

4.6.2 Model Refinement and Advanced Modelling Approaches

The current models use relatively straightforward algorithms and hand-crafted features. There is room to explore more advanced machine learning techniques, such as time-series modelling (e.g., recurrent neural networks or Bayesian trajectory models) that could utilise the full longitudinal data without requiring predefined feature extraction. Such models might capture subtle nuances in the curve shape (for instance, the curvature or fluctuations of hCG decline) that the summary features might miss.

However, the challenge will be maintaining interpretability and avoiding overfitting, especially given that GTD datasets, while larger than many rare disease cohorts, are still limited in size for deep learning methods. An intermediate approach could be to employ techniques like survival analysis or longitudinal mixed-effects models incorporating machine learning, which might predict not just a binary outcome but the hazard or timing of needing escalation. This could answer questions like “when is the earliest time point at which the model’s prediction becomes reliable?” – maybe week 3 instead of week 4, for instance.

4.6.3 Integration of Additional Clinical Predictors

Integration of additional predictors is another avenue for future development. As noted in the limitations, this thesis was restricted to hCG-based features. Future research could build composite models that include clinical covariates (age, antecedent pregnancy type, etc.) and possibly molecular markers. There is emerging research into genetic or proteomic markers in GTD that could prognosticate outcomes (for example, detecting certain oncogenic mutations or measuring other placental hormones).

While hCG is the cornerstone, a multimodal model might improve predictive power. It would be worthwhile to investigate, for example, whether adding the pathology of the mole (complete vs partial) into the urine model improves its performance – given that complete moles have a higher baseline risk of GTD (Ngan et al., 2018), incorporating that prior risk might refine the probability estimates. Likewise, including the FIGO score at diagnosis of GTD might enhance the MTX resistance model (since a score of 5 or 6, though still "low-risk", is known to correlate with higher likelihood of single-agent failure).

4.6.4 Clinical Evaluation and Implementation

From a clinical trial perspective, an intervention study could be designed based on the model. For instance, a randomised trial could be set up where one arm follows standard management of low-risk GTD and the other arm uses an algorithm-informed strategy (earlier consideration of ActD escalation if the model predicts inadequate response). The outcomes of interest would be total time to remission, number of chemotherapy cycles given, drug toxicity, and potentially patient-reported quality of life.

Such a trial would provide the highest level of evidence on whether using the model benefits patients (e.g., does it significantly shorten treatment duration or reduce complications?). Ethical and logistical considerations would need to be addressed – since current outcomes are excellent, the bar for improvement is high and any new approach must be shown to be at least as safe.

Additionally, further research might examine cost-effectiveness and health economics of implementing predictive monitoring. Are the gains from early intervention (fewer clinic visits, fewer cycles of chemotherapy) enough to justify the costs of developing and maintaining an AI system? Given that hCG tests and treatment are not extremely costly on a per-unit basis, a formal

analysis would be needed to see if model-guided care reduces overall resource use. This is especially pertinent in publicly funded health systems like the UK.

It would also be interesting to explore patient attitudes toward AI predictions in this context – qualitative research could assess whether patients would be comfortable with decisions influenced by an algorithm or whether they prefer sticking strictly to traditional criteria.

4.6.5 Understanding Model Error and Further Optimisation

Finally, an area for future work is investigating why the models make errors in certain cases. Analysing the false negatives and false positives might uncover patterns. For example, were the false negatives in the urine model mostly partial moles that eventually plateaued late? If so, perhaps extending the monitoring window or adding a feature for slope beyond 4 weeks could help.

Were the false-positive MTX resistance predictions concentrated in patients with very high initial hCG that nonetheless responded to MTX? This might suggest incorporating an interaction term or threshold that tempers the model for known responders. By understanding these outliers, we could refine the feature set or model parameters to improve reliability.

In summary, further research should focus on validating the models in new patient groups, improving the models with additional data and modern techniques, and ultimately assessing the impact of using such models in the clinical management of GTD. The goal would be to transition this work from retrospective analysis to a prospective clinical tool that demonstrably enhances patient care. With international collaboration (given GTD is rare in any single centre), such research is feasible and could position hCG predictive modelling as part of the future standard of care in gestational trophoblastic disease.

5.0 Conclusion

Gestational trophoblastic disease represents a unique clinical model in which longitudinal biochemical monitoring provides a direct measure of disease activity and response to treatment. Although hCG surveillance has formed the foundation of GTD management for several decades, traditional interpretation remains largely dependent on predefined thresholds, plateau criteria, and assessment of individual measurements. This thesis investigated whether additional clinically meaningful information could be extracted from serial hCG trajectories through statistical modelling and machine learning approaches, with the aim of improving early risk stratification and supporting more personalised patient management.

Using large retrospective datasets from the Sheffield GTD service, this work demonstrated that serial hCG kinetics contain predictive information that extends beyond conventional interpretation of isolated results. In patients undergoing post-molar surveillance, urinary hCG/creatinine trajectories showed measurable differences between those who achieved spontaneous resolution and those requiring treatment. These findings demonstrate that naturally resolving disease follows characteristic biochemical patterns and that early urine hCG profiles can contribute to identification of patients at increased risk of persistent disease.

In patients receiving first-line MTX therapy for low-risk GTD, early serum hCG response patterns differed between patients successfully treated with MTX and those demonstrating inadequate treatment response requiring ActD. Features describing the rate and magnitude of hCG decline, rather than baseline hCG concentration alone, provided important predictive information. This highlights the value of assessing dynamic treatment response and supports the concept that early biochemical behaviour may provide insight into subsequent treatment outcomes.

The predictive models developed within this thesis demonstrate the potential value of combining established biochemical monitoring with modern analytical approaches. Machine learning methods, particularly ensemble-based models, were able to identify complex hCG trajectory patterns associated with patient outcomes. Importantly, these models should not be viewed as replacements for current GTD management pathways, which remain highly effective, but rather as potential decision-support tools that may enhance risk stratification and enable earlier identification of patients requiring closer monitoring or treatment modification.

This thesis provides several original contributions to GTD research, including the application of machine learning approaches to serial hCG trajectory modelling, the evaluation of urinary hCG/creatinine patterns as predictive markers, and the identification of clinically interpretable kinetic features associated with treatment outcomes. The findings demonstrate that routinely collected clinical data contain additional information that may be utilised to further personalise patient care.

Before translation into clinical practice, prospective validation using independent patient cohorts is required to confirm model performance and evaluate clinical utility. Future research should focus on determining whether model-guided approaches can safely reduce treatment burden, minimise exposure to ineffective therapy, and improve patient experience while maintaining the excellent outcomes currently achieved in GTD management.

In conclusion, this thesis demonstrates that serial hCG monitoring contains previously underutilised predictive information that can be identified using modern computational approaches. By integrating traditional biochemical surveillance with predictive modelling, this work provides a foundation for future development of personalised GTD monitoring strategies, supporting earlier intervention and improved optimisation of care for patients with this highly curable disease.

6.0 References

Agarwal, R., Teoh, S., Short, D., Harvey, R., Savage, P.M. and Seckl, M.J. (2012). 'Chemotherapy and human chorionic gonadotropin concentrations 6 months after uterine evacuation of molar pregnancy: a retrospective cohort study', *The Lancet*, 379(9811), pp.130–135. doi:10.1016/S0140-6736(11)61265-8.

Akobeng, A.K. (2007) 'Understanding diagnostic tests 1: sensitivity, specificity and predictive values', *Acta Paediatrica*, 96(3), pp.338–341. doi:10.1111/j.1651-2227.2006.00180.x.

Alazzam, M., Tidy, J., Osborne, R., Coleman, R., Hancock, B.W. and Lawrie, T.A. (2016) 'Chemotherapy for resistant or recurrent gestational trophoblastic neoplasia', *Cochrane Database of Systematic Reviews*, 2016(12), CD008891. doi:10.1002/14651858.CD008891.pub2.

Albright, B.B., Shorter, J.M., Mastroyannis, S.A., Fader, A.N., Ko, E.M., Huh, W.K., Moore, K.N., Bixel, K., Moxley, K.M., Tergas, A.I., Burke, W.M., Wright, J.D. and Hou, J.Y. (2020). 'Gestational trophoblastic neoplasia after human chorionic gonadotropin normalisation following molar pregnancy: a systematic review and meta-analysis', *Obstetrics & Gynecology*, 135(1), pp. 12–23. doi:10.1097/AOG.0000000000003605.

Allegra, C.J., Chabner, B.A., Drake, J.C., Lutz, R., Rodbard, D. and Jolivet, J. (1985). 'Enhanced inhibition of thymidylate synthase by methotrexate polyglutamates', *Journal of Biological Chemistry*, 260(17), pp.9720–9726.

Allen, S.D., Lim, A.K., Seckl, M.J., Blunt, D.M. and Mitchell, A.W. (2006). 'Radiology of gestational trophoblastic neoplasia', *Clinical Radiology*, 61(4), pp.301–313. doi:10.1016/j.crad.2005.12.003.

Allison, K.H., Love, J.E. and Garcia, R.L. (2006). 'Epithelioid trophoblastic tumor: review of a rare neoplasm of the chorionic-type intermediate trophoblast', *Archives of Pathology & Laboratory Medicine*, 130(12), pp.1875–1877. doi:10.5858/2006-130-1875-ETTROA.

Altieri, A., Franceschi, S., Ferlay, J., Smith, J. and La Vecchia, C. (2003). 'Epidemiology and aetiology of gestational trophoblastic diseases', *The Lancet Oncology*, 4(11), pp. 670–678. doi:10.1016/S1470-2045(03)01245-2.

Altman, D.G. and Bland, J.M. (1994) 'Diagnostic tests 1: sensitivity and specificity', *BMJ*, 308(6943), p.1552. doi:10.1136/bmj.308.6943.1552.

American College of Obstetricians and Gynecologists Committee on Gynecologic Practice (2002). 'Avoiding inappropriate clinical decisions based on false-positive human chorionic gonadotropin test results. Committee Opinion No. 278', *Obstetrics & Gynecology*, 100(5 Pt 1), pp. 1057–1059. doi:10.1016/S0029-7844(02)02515-2.

Assaraf, Y.G. (2007). 'Molecular basis of antifolate resistance', *Cancer and Metastasis Reviews*, 26(1), pp.153–181. doi:10.1007/s10555-007-9049-z.

Assaraf, Y.G., Brozovic, A., Gonçalves, A.C., Jurkovicova, D., Linē, A., Machuqueiro, M., Saponara, S., Sarmento-Ribeiro, A.B., Xavier, C.P.R. and Vasconcelos, M.H. (2019). 'The multi-factorial nature of clinical multidrug resistance in cancer', *Drug Resistance Updates*, 46, Article 100645. doi:10.1016/j.drup.2019.100645.

Bagherzadeh-Khiabani, F., Ramezankhani, A., Azizi, F., Hadaegh, F., Steyerberg, E.W. and Khalili, D. (2016). 'A tutorial on variable selection for clinical prediction models: feature selection methods in data mining could improve the results', *Journal of Clinical Epidemiology*, 71, pp.76–85. doi:10.1016/j.jclinepi.2015.10.002.

Bagshawe, K.D. (1963). 'Trophoblastic tumours: chemotherapy and developments', *British Medical Journal*, 2(5368), pp. 1303–1307. doi:10.1136/bmj.2.5368.1303.

Bagshawe, K.D. (1976). 'Risk and prognostic factors in trophoblastic neoplasia', *Cancer*, 38(3), pp. 1373–1385. doi:10.1002/1097-0142(197609)38:3<1373::AID-CNCR2820380342>3.0.CO;2-E.

Bagshawe, K.D. (1992). 'Choriocarcinoma. A model for tumour markers', *Acta Oncologica*, 31(1), pp. 99–106. doi:10.3109/02841869209088275.

Bagshawe, K.D., Dent, J. and Webb, J. (1986). 'Hydatidiform mole in England and Wales 1973–83', *The Lancet*, 328(8508), pp. 673–677. doi:10.1016/S0140-6736(86)90179-0.

Balaram, P., John, M., Rajalekshmy, T.N., Nair, B., Schultz, G. and Nair, K. (1997). 'Expression of epidermal growth factor receptor in gestational trophoblastic diseases', *Journal of Cancer Research and Clinical Oncology*, 123(3), pp.161–166. doi:10.1007/BF01214668.

Banet, N., DeScipio, C., Murphy, K.M., Beierl, K., Adams, E., Vang, R. and Ronnett, B.M. (2014). 'Characteristics of hydatidiform moles: analysis of a prospective series with p57 immunohistochemistry and molecular genotyping', *American Journal of Surgical Pathology*, 38(3), pp. 341–352. doi:10.1097/PAS.0b013e31825ea736.

Barlow, D.P. and Bartolomei, M.S. (2014). 'Genomic imprinting in mammals', *Cold Spring Harbor Perspectives in Biology*, 6(2), article a018382. doi:10.1101/cshperspect.a018382.

Bergstra, J. and Bengio, Y. (2012). 'Random search for hyper-parameter optimization', *Journal of Machine Learning Research*, 13, pp.281–305.

Bergstra, J., Yamins, D. and Cox, D.D. (2013). 'Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures', *Proceedings of the 30th International Conference on Machine Learning*, 28, pp.115–123.

Berkowitz, R.S. and Goldstein, D.P. (1995). 'Gestational trophoblastic disease', *Cancer*, 76(10 Suppl.), pp. 2079–2085. doi:10.1002/1097-0142(19951115)76:10+<2079::aid-cncr2820761329>3.0.co;2-o.

Berkowitz, R.S. and Goldstein, D.P. (2009). 'Current management of gestational trophoblastic diseases', *Gynecologic Oncology*, 112(3), pp. 654–662. doi:10.1016/j.ygyno.2008.09.005.

Berkowitz, R.S., Im, S.S., Bernstein, M.R. and Goldstein, D.P. (1998). 'Gestational trophoblastic disease: subsequent pregnancy outcome, including repeat molar pregnancy', *Journal of Reproductive Medicine*, 43(1), pp.81–86.

Biau, G. (2012). 'Analysis of a random forests model', *Journal of Machine Learning Research*, 13, pp.1063–1095.

Bilban, M., Tauber, S., Haslinger, P., Pollheimer, J., Saleh, L., Pehamberger, H., Wagner, O. and Knöfler, M. (2010). 'Trophoblast invasion: assessment of cellular models using gene expression signatures', *Placenta*, 31(11), pp. 989–996. doi:10.1016/j.placenta.2010.08.011.

Blagden, S.P., Foskett, M.A., Fisher, R.A., Short, D., Fuller, S., Newlands, E.S. and Seckl, M.J. (2002). 'The effect of early pregnancy following chemotherapy on disease relapse and foetal outcome in women treated for gestational trophoblastic tumours', *British Journal of Cancer*, 86(1), pp.26–30. doi:10.1038/sj.bjc.6600041.

Bolze, P.A., Riedl, C., Massardier, J., Lotz, J.P., You, B., Schott, A.M., Hajri, T. and Golfier, F. (2016). 'Mortality rate of gestational trophoblastic neoplasia with a FIGO score of ≥13', *American Journal of Obstetrics and Gynecology*, 214(3), pp.390.e1–390.e8. doi:10.1016/j.ajog.2015.09.083.

Boulesteix, A.L., Janitza, S., Kruppa, J. and König, I.R. (2012) 'Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics', *WIREs Data Mining and Knowledge Discovery*, 2(6), pp.493–507. doi:10.1002/widm.1072.

Braga, A., Biscaro, A., do Amaral Giordani, J.M., Viggiano, M., Elias, K.M., Berkowitz, R.S. and Seckl, M.J. (2018) 'Does a human chorionic gonadotropin level of over 20,000 IU/L four weeks after uterine evacuation for complete hydatidiform mole constitute an indication for chemotherapy for gestational trophoblastic neoplasia?', *European Journal of Obstetrics & Gynecology and Reproductive Biology*, 223, pp.50–55. doi:10.1016/j.ejogrb.2018.02.001.

Braga, A., Mora, P., de Melo, A.C., Nogueira-Rodrigues, A., Amim-Junior, J., Rezende-Filho, J. and Seckl, M.J. (2019). 'Challenges in the diagnosis and treatment of gestational trophoblastic neoplasia worldwide', *World Journal of Clinical Oncology*, 10(2), pp.28–37. doi:10.5306/wjco.v10.i2.28.

Braga, A., Elias, K.M., Horowitz, N.S. and Berkowitz, R.S. (2021a). 'Challenges in the treatment of low-risk gestational trophoblastic neoplasia', *Revista Brasileira de Ginecologia e Obstetrícia*, 43(7), pp.503–506. doi:10.1055/s-0041-1735177.

Braga, A., Paiva, G., Ghorani, E., Freitas, F., Velarde, L.G.C., Kaur, B., Unsworth, N., Lozano-Kuehne, J., Dos Santos Esteves, A.P.V., Rezende Filho, J., Amim, J. Jr, Aguiar, X., Sarwar, N., Elias, K.M., Horowitz, N.S., Berkowitz, R.S. and Seckl, M.J. (2021b). 'Predictors for single-agent resistance in FIGO score 5 or 6 gestational trophoblastic neoplasia: A multicentre, retrospective, cohort study', *The Lancet Oncology*, 22(8), pp.1188–1198. doi:10.1016/S1470-2045(21)00262-X.

Breiman, L. (2001). 'Random forests', *Machine Learning*, 45, pp.5–32. doi:10.1023/A:1010933404324.

Brown, M.A., Magee, L.A., Kenny, L.C., Karumanchi, S.A., McCarthy, F.P., Saito, S., Hall, D.R., Warren, C.E., Adoyi, G. and Ishaku, S. (2018). 'Hypertensive disorders of pregnancy: ISSHP classification, diagnosis, and management recommendations for international practice', *Hypertension*, 72(1), pp. 24–43. doi:10.1161/HYPERTENSIONAHA.117.10803.

Brown, J., Naumann, R.W., Seckl, M.J. and Schink, J. (2017). '15 years of progress in gestational trophoblastic disease: scoring, standardization, and salvage', *Gynecologic Oncology*, 144(1), pp.200–207. doi:10.1016/j.ygyno.2016.08.330.

Buza, N. and Hui, P. (2014). 'Immunohistochemistry and other ancillary techniques in the diagnosis of gestational trophoblastic diseases', *Seminars in Diagnostic Pathology*, 31(3), pp. 223–232. doi:10.1053/j.semdp.2014.06.004.

Candelier, J.J. (2016). 'The hydatidiform mole', *Cell Adhesion & Migration*, 10(1–2), pp. 226–235. doi:10.1080/19336918.2015.1093275.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M. and Elhadad, N. (2015). 'Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission', *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1721–1730. doi:10.1145/2783258.2788613.

Chabner, B.A. and Roberts, T.G. Jr. (2005). 'Timeline: Chemotherapy and the war on cancer', *Nature Reviews Cancer*, 5(1), pp.65–72. doi:10.1038/nrc1529.

Chen, T. and Guestrin, C. (2016). 'XGBoost: A scalable tree boosting system', *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.785–794. doi:10.1145/2939672.2939785.

Chen, X. and Ishwaran, H. (2012) 'Random forests for genomic data analysis', *Genomics*, 99(6), pp.323–329. doi:10.1016/j.ygeno.2012.04.003.

Chicco, D. and Jurman, G. (2020). 'The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation', *BMC Genomics*, 21, 6. doi:10.1186/s12864-019-6413-7.

Chowdhury, M.Z.I. and Turin, T.C. (2020). 'Variable selection strategies and its importance in clinical prediction modelling', *Family Medicine and Community Health*, 8(1), e000262. doi:10.1136/fmch-2019-000262.

Christodoulou, E., Ma, J., Collins, G.S., Steyerberg, E.W., Verbakel, J.Y. and Van Calster, B. (2019) 'A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models', *Journal of Clinical Epidemiology*, 110, pp.12–22. doi:10.1016/j.jclinepi.2019.02.004.

Cole, L.A. (2009) 'New discoveries on the biology and detection of human chorionic gonadotropin', *Reproductive Biology and Endocrinology*, 7, Article 8. doi:10.1186/1477-7827-7-8.

Cole, L.A. (2010). 'Biological functions of hCG and hCG-related molecules', *Reproductive Biology and Endocrinology*, 8, Article 102. doi:10.1186/1477-7827-8-102.

Cole, L.A. (2012). 'hCG, five independent molecules', *Clinica Chimica Acta*, 413(1–2), pp.48–65. doi:10.1016/j.cca.2011.09.037.

Cole, L.A. (2012). 'hCG variants, the growth factors which drive human malignancies', *American Journal of Cancer Research*, 2(1), pp. 22–35.

Cole, L.A. and Butler, S.A. (2015). *Human Chorionic Gonadotropin (hCG)*. 2nd edn. London: Elsevier.

Cole, L.A. and Kohorn, E.I. (2006). 'The need for an hCG assay that appropriately detects trophoblastic disease and other hCG-producing cancers', *Journal of Reproductive Medicine*, 51(10), pp. 793–811.

Cole, L.A. and Sutton, J.M. (2004). 'Selecting an appropriate hCG test for managing gestational trophoblastic disease and cancer', *Journal of Reproductive Medicine*, 49(7), pp. 545–553.

Collins, G.S., Dhiman, P., Ma, J., Schlussel, M.M., Archer, L., Van Calster, B., Harrell, F.E., Martin, G.P., Moons, K.G.M., van Smeden, M., Sperrin, M., Bullock, G.S. and Riley, R.D. (2024). 'Evaluation of clinical prediction models (part 1): from development to external validation', *BMJ*, 384, e074819. doi:10.1136/bmj-2023-074819.

Couronné, R., Probst, P. and Boulesteix, A.L. (2018). 'Random forest versus logistic regression: a large-scale benchmark experiment', *BMC Bioinformatics*, 19, 270. doi:10.1186/s12859-018-2264-5.

Coullin, P., Diatta, A.L., Boufettal, H., Feingold, J., Leguern, E. and Candelier, J.J. (2015). 'The involvement of the trans-generational effect in the high incidence of the hydatidiform mole in Africa', *Placenta*, 36(1), pp. 48–51. doi:10.1016/j.placenta.2014.10.017.

Cowan, K.H., Goldsmith, M.E., Levine, R.M., Aitken, S.C., Douglass, E., Clendeninn, N., Nienhuis, A.W. and Lippman, M.E. (1982). 'Dihydrofolate reductase gene amplification and possible rearrangement in estrogen-responsive methotrexate-resistant human breast cancer cells', *Journal of Biological Chemistry*, 257(24), pp.15079–15086.

Coyle, C., Short, D., Jackson, L., Sebire, N.J., Kaur, B., Harvey, R., Savage, P.M. and Seckl, M.J. (2018). 'What is the optimal duration of human chorionic gonadotrophin surveillance following evacuation of a molar pregnancy? A retrospective analysis on over 20,000 consecutive patients', *Gynecologic Oncology*, 148(2), pp.254–257. doi:10.1016/j.ygyno.2017.12.008.

Crosby, D., Bhatia, S., Brindle, K.M., Coussens, L.M., Dive, C., Emberton, M., Esener, S., Fitzgerald, R.C., Gambhir, S.S., Kuhn, P., Rebbeck, T.R. and Balasubramanian, S. (2022) 'Early detection of cancer', *Science*, 375(6586), eaay9040. doi:10.1126/science.aay9040.

Cunningham, F.G., Leveno, K.J., Bloom, S.L., Dashe, J.S., Hoffman, B.L., Casey, B.M. and Spong, C.Y. (2018). *Williams Obstetrics*. 25th edn. New York: McGraw Hill.

Cutler, D.R., Edwards, T.C., Beard, K.H., Cutler, A., Hess, K.T., Gibson, J. and Lawler, J.J. (2007). 'Random forests for classification in ecology', *Ecology*, 88(11), pp.2783–2792. doi:10.1890/07-0539.1.

de Souza, J.M.Q., Braga, A., Sanches Dos Santos, R., Ramos, M.M., Cortés-Charry, R. and Maestá, I. (2017). 'Comparison of 2 human chorionic gonadotropin immunoassays commercially available for monitoring patients with gestational trophoblastic disease', *International Journal of Gynecological Cancer*, 27(7), pp.1494–1500. doi:10.1097/IGC.0000000000001042.

Defazio, A., Bach, F. and Lacoste-Julien, S. (2014) 'SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives', *Advances in Neural Information Processing Systems*, 27.

Dhanda, S., Ramani, S. and Thakur, M. (2014). 'Gestational trophoblastic disease: a multimodality imaging approach with impact on diagnosis and management', *Radiology Research and Practice*, 2014, Article ID 842751. doi:10.1155/2014/842751.

Dietterich, T.G. (2000). 'Ensemble methods in machine learning', *Lecture Notes in Computer Science*, 1857, pp.1–15. doi:10.1007/3-540-45014-9_1.

Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006) 'Gene selection and classification of microarray data using random forest', *BMC Bioinformatics*, 7, Article 3. doi:10.1186/1471-2105-7-3.

Eagles, N., Sebire, N.J., Short, D., Savage, P.M., Seckl, M.J. and Fisher, R.A. (2015). 'Risk of recurrent molar pregnancies following complete and partial hydatidiform moles', *Human Reproduction*, 30(9), pp.2055–2063. doi:10.1093/humrep/dev169.

Eysbouts, Y., Brouwer, R., Ottevanger, P., Massuger, L., Sweep, F., Thomas, C. and van Herwaarden, A. (2017) 'Serum human chorionic gonadotropin normogram for the detection of gestational trophoblastic neoplasia', *International Journal of Gynecological Cancer*, 27(5), pp.1035–1041. doi:10.1097/IGC.0000000000000966.

Eysbouts, Y.K., Massuger, L.F.A.G., IntHout, J., Lok, C.A.R., Sweep, F.C.G.J. and Ottevanger, P.B. (2017). 'The added value of hysterectomy in the management of gestational trophoblastic neoplasia', *Gynecologic Oncology*, 145(3), pp.536–542. doi:10.1016/j.ygyno.2017.03.018.

Feng, X., Wei, Z., Zhang, S., Du, Y. and Zhao, H. (2019) 'A review on the pathogenesis and clinical management of placental site trophoblastic tumors', *Frontiers in Oncology*, 9, Article 937. doi:10.3389/fonc.2019.00937.

Fernández-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014). 'Do we need hundreds of classifiers to solve real world classification problems?', *Journal of Machine Learning Research*, 15, pp.3133–3181.

Ferri, C., Hernández-Orallo, J. and Modroiu, R. (2009). 'An experimental comparison of performance measures for classification', *Pattern Recognition Letters*, 30(1), pp.27–38. doi:10.1016/j.patrec.2008.08.010.

FIGO Oncology Committee (2002). 'FIGO staging for gestational trophoblastic neoplasia 2000', *International Journal of Gynecology & Obstetrics*, 77(3), pp. 285–287. doi:10.1016/S0020-7292(02)00063-2.

Fisher, R.A. and Maher, G.J. (2021). 'Genetics of gestational trophoblastic disease', *Best Practice & Research Clinical Obstetrics & Gynaecology*, 74, pp. 29–41. doi:10.1016/j.bpobgyn.2021.01.004.

Franks, C.E., Li, J., Martinez, M., Farnsworth, C.W., Jones, P.M., Grenache, D.G., Meng, Q.H. and Gronowski, A.M. (2023). 'Utility of commercially available quantitative hCG immunoassays as tumor markers in trophoblastic and non-trophoblastic disease', *Clinical Chemistry*, 69(6), pp. 606–614. doi:10.1093/clinchem/hvad045.

Friedman, J.H. (2001). 'Greedy function approximation: a gradient boosting machine', *Annals of Statistics*, 29(5), pp.1189–1232. doi:10.1214/aos/1013203451.

Froeling, F.E.M., Ramaswami, R., Papanastasopoulos, P., Kaur, B., Sebire, N.J., Short, D., Fisher, R.A., Sarwar, N., Wells, M., Singh, K., Ellis, L., Horsman, J.M., Winter, M.C., Tidy, J., Hancock,

B.W. and Seckl, M.J. (2019). ‘Intensified therapies improve survival and identification of novel prognostic factors for placental-site and epithelioid trophoblastic tumours’, *British Journal of Cancer*, 120(6), pp.587–594. doi:10.1038/s41416-019-0402-0.

Gadducci, A., Carinelli, S., Guerrieri, M.E. and Aletti, G.D. (2019). ‘Placental site trophoblastic tumor and epithelioid trophoblastic tumor: Clinical and pathological features, prognostic variables and treatment strategy’, *Gynecologic Oncology*, 153(3), pp.684–693. doi:10.1016/j.ygyno.2019.03.011.

Genest, D.R. (2001). ‘Partial hydatidiform mole: clinicopathological features, differential diagnosis, ploidy and molecular studies, and gold standards for diagnosis’, *International Journal of Gynecological Pathology*, 20(4), pp.315–322. doi:10.1097/00004347-200110000-00001.

Ghorani, E., Kaur, B., Fisher, R.A., Short, D., Joneborg, U., Carlson, J.W., Akarca, A., Marafioti, T., Quezada, S.A., Sarwar, N. and Seckl, M.J. (2017). ‘Pembrolizumab is effective for drug-resistant gestational trophoblastic neoplasia’, *The Lancet*, 390(10110), pp.2343–2345. doi:10.1016/S0140-6736(17)32894-5.

Goldberg, I.H., Rabinowitz, M. and Reich, E. (1962). ‘Basis of actinomycin action. I. DNA binding and inhibition of RNA-polymerase synthetic reactions by actinomycin’, *Proceedings of the National Academy of Sciences of the United States of America*, 48(12), pp.2094–2101.

Goldstein, D.P. and Berkowitz, R.S. (2012). ‘Current management of gestational trophoblastic neoplasia’, *Hematology/Oncology Clinics of North America*, 26(1), pp. 111–131. doi:10.1016/j.hoc.2011.10.007.

Gonen, N. and Assaraf, Y.G. (2012). ‘Antifolates in cancer therapy: Structure, activity and mechanisms of drug resistance’, *Drug Resistance Updates*, 15(4), pp.183–210. doi:10.1016/j.drup.2012.07.002.

Gonzalez, J., Popp, M., Ocejo, S., Abreu, A., Bahmad, H.F. and Poppiti, R. (2024). ‘Gestational trophoblastic disease: complete versus partial hydatidiform moles’, *Diseases*, 12(7), Article 159. doi:10.3390/diseases12070159.

Goutte, C. and Gaussier, E. (2005) ‘A probabilistic interpretation of precision, recall and F-score, with implication for evaluation’, in Losada, D.E. and Fernández-Luna, J.M. (eds.) *Advances in Information Retrieval. ECIR 2005*. Lecture Notes in Computer Science, Vol. 3408. Berlin: Springer, pp.345–359. doi:10.1007/978-3-540-31865-1_25.

Grenache, D.G., Greene, D.N., Dighe, A.S., Fantz, C.R., Hoefner, D., McCudden, C.R., Sokoll, L.J., Wiley, C.L. and Gronowski, A.M. (2010). 'Falsely decreased human chorionic gonadotropin (hCG) results due to increased concentrations of the free β-subunit and the β-core fragment in quantitative hCG assays', *Clinical Chemistry*, 56(12), pp. 1839–1844. doi:10.1373/clinchem.2010.143479.

Guyon, I. and Elisseeff, A. (2003). 'An introduction to variable and feature selection', *Journal of Machine Learning Research*, 3, pp.1157–1182.

Hajian-Tilaki, K. (2013). 'Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation', *Caspian Journal of Internal Medicine*, 4(2), pp.627–635.

Han, B., Xiang, Y., Wang, Y., Wang, Z., Zhang, H. and Huang, S. (2010). 'Dihydrofolate reductase transcript level is not suitable for methotrexate-resistance prediction in choriocarcinoma cell line', *International Journal of Gynecological Cancer*, 20(7), pp.1259–1263. doi:10.1111/igc.0b013e3181f05128.

Hanahan, D. (2022). 'Hallmarks of cancer: new dimensions', *Cancer Discovery*, 12(1), pp. 31–46. doi:10.1158/2159-8290.CD-21-1059.

Hancock, B.W. and Tidy, J. (2021). 'Placental site trophoblastic tumour and epithelioid trophoblastic tumour', *Best Practice & Research Clinical Obstetrics & Gynaecology*, 74, pp.131–148. doi:10.1016/j.bpobgyn.2020.10.004.

Harrell, F.E. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd edn. Cham: Springer.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edn. New York: Springer.

Hoeijmakers, Y.M., Eysbouts, Y.K., Massuger, L.F.A.G., Dandis, R., IntHout, J., van Trommel, N.E., Ottevanger, P.B., Thomas, C.M.G. and Sweep, F.C.G.J. (2021). 'Early prediction of post-molar gestational trophoblastic neoplasia and resistance to methotrexate, based on a single serum human chorionic gonadotropin measurement', *Gynecologic Oncology*, 163(3), pp.531–537. doi:10.1016/j.ygyno.2021.09.016.

Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X. (2013) *Applied Logistic Regression*. 3rd edn. Hoboken, NJ: Wiley.

Howard, S.C., McCormick, J., Pui, C.H., Buddington, R.K. and Harvey, R.D. (2016). 'Preventing and managing toxicities of high-dose methotrexate', *The Oncologist*, 21(12), pp.1471–1482. doi:10.1634/theoncologist.2015-0164.

Hunter, J.S. (1959). 'Treatment of hydatidiform mole', *Surgical Clinics of North America*, 39(4), pp. 1131–1136. doi:10.1016/S0039-6109(16)35853-4.

Hui, P., Buza, N., Murphy, K.M. and Ronnett, B.M. (2017). 'Hydatidiform moles: genetic basis and precision diagnosis', *Annual Review of Pathology: Mechanisms of Disease*, 12, pp. 449–485. doi:10.1146/annurev-pathol-052016-100237.

Iles, R.K., Delves, P.J. and Butler, S.A. (2010). 'Does hCG or hCGβ play a role in cancer cell biology?', *Molecular and Cellular Endocrinology*, 329(1–2), pp.62–70. doi:10.1016/j.mce.2010.07.014.

Jain, A. (2016) 'Complete guide to parameter tuning in Gradient Boosting (GBM) in Python', *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/ (Accessed: 14 February 2026).

Jin, T., Zhou, Z., Lin, M., Yuan, S., Xue, Y., Yin, T., Xu, R., Chen, B., Zhang, J., Sun, J., Li, X., Hu, Y., Chen, L. and Wang, H. (2024). 'Risk factors for methotrexate resistance in low-risk gestational trophoblastic neoplasia patients (FIGO score 0–4)', *American Journal of Cancer Research*, 14(3), pp.1353–1362. doi:10.62347/ZUCG8140.

Joyce, C.M., Fitzgerald, B., McCarthy, T.V., Coulter, J. and O'Donoghue, K. (2022). 'Advances in the diagnosis and early management of gestational trophoblastic disease', *BMJ Medicine*, 1(1), e000321. doi:10.1136/bmjmed-2022-000321.

Kaufman, S., Rosset, S., Perlich, C. and Stitelman, O. (2012). 'Leakage in data mining: formulation, detection, and avoidance', *ACM Transactions on Knowledge Discovery from Data*, 6(4), article 15. doi:10.1145/2382577.2382579.

Kaur, B. (2024). 'Pathology of gestational trophoblastic disease (GTD)', *Hematology/Oncology Clinics of North America*, 38(6), pp. 1191–1217. doi:10.1016/j.hoc.2024.08.017.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y. (2017) 'LightGBM: a highly efficient gradient boosting decision tree', *Advances in Neural Information Processing Systems*, 30.

Kelly, C.J., Karthikesalingam, A., Suleyman, M., Corrado, G. and King, D. (2019) 'Key challenges for delivering clinical impact with artificial intelligence', *BMC Medicine*, 17, Article 195. doi:10.1186/s12916-019-1426-2.

Kerkmeijer, L.G., Thomas, C.M., Harvey, R., Sweep, F.C., Mitchell, H., Massuger, L.F. and Seckl, M.J. (2009). 'External validation of serum hCG cutoff levels for prediction of resistance to single-agent chemotherapy in patients with persistent trophoblastic disease', *British Journal of Cancer*, 100(6), pp.979–984. doi:10.1038/sj.bjc.6604849.

King, G. and Zeng, L. (2001) 'Logistic regression in rare events data', *Political Analysis*, 9(2), pp.137–163. doi:10.1093/oxfordjournals.pan.a004868.

Kirk, E., Papageorghiou, A.T., Condous, G., Bottomley, C., Bourne, T. (2007). 'The accuracy of first trimester ultrasound in the diagnosis of hydatidiform mole', Ultrasound in Obstetrics & Gynecology, 29(1), pp.70–75. doi:10.1002/uog.3885.

Kladnitski, M. and Kenwright, D. (2016) 'Management of gestational trophoblastic disease: a survey of New Zealand O&G practice', *New Zealand Medical Journal*, 129(1431), pp.38–45.

Knöfler, M. and Pollheimer, J. (2013). 'Human placental trophoblast invasion and differentiation: a particular focus on Wnt signaling', *Frontiers in Genetics*, 4, Article 190. doi:10.3389/fgene.2013.00190.

Knöfler, M., Haider, S., Saleh, L., Pollheimer, J., Gamage, T.K.J.B. and James, J. (2019). 'Human placenta and trophoblast development: key molecular mechanisms and model systems', *Cellular and Molecular Life Sciences*, 76(18), pp. 3479–3496. doi:10.1007/s00018-019-03104-6.

Koba, M. and Konopa, J. (2005). 'Actinomycin D and its mechanisms of action', *Postępy Higieny i Medycyny Doświadczalnej*, 59, pp.290–298.

Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I. (2015) 'Machine learning applications in cancer prognosis and prediction', Computational and Structural Biotechnology Journal, 13, pp.8–17.

Kurman, R.J., Scully, R.E. and Norris, H.J. (1976). 'Trophoblastic pseudotumor of the uterus: an exaggerated form of "syncytial endometritis" simulating a malignant tumor', *Cancer*, 38(3), pp. 1214–1226. doi:10.1002/1097-0142(197609)38:3<1214::AID-CNCR2820380323>3.0.CO;2-J.

La Vecchia, C., Franceschi, S., Parazzini, F., Fasoli, M., Decarli, A., Gallus, G. and Tognoni, G. (1985). 'Risk factors for gestational trophoblastic disease in Italy', *American Journal of Epidemiology*, 121(3), pp. 457–464. doi:10.1093/oxfordjournals.aje.a114018.

Lai, C.Y., Chan, K.Y., Khoo, U.S., Ngan, H.Y., Xue, W.C., Chiu, P.M., Tsao, S.W. and Cheung, A.N. (2004). 'Analysis of gestational trophoblastic disease by genotyping and chromosome in situ hybridization', *Modern Pathology*, 17(1), pp.40–48. doi:10.1038/modpathol.3800010.

Lawrie, T.A., Alazzam, M., Tidy, J., Hancock, B.W. and Osborne, R. (2016). 'First-line chemotherapy in low-risk gestational trophoblastic neoplasia', *Cochrane Database of Systematic Reviews*, 2016(6), CD007102. doi:10.1002/14651858.CD007102.pub4.

Li, H.W., Tsao, S.W. and Cheung, A.N.Y. (2002). 'Current understandings of the molecular genetics of gestational trophoblastic diseases', *Placenta*, 23(1), pp. 20–31. doi:10.1053/plac.2001.0744.

Li, M.C., Hertz, R. and Bergenstal, D.M. (1958). 'Therapy of choriocarcinoma and related trophoblastic tumors with folic acid and purine antagonists', *New England Journal of Medicine*, 259(2), pp. 66–74.

Liu, D.C. and Nocedal, J. (1989) 'On the limited memory BFGS method for large scale optimization', *Mathematical Programming*, 45, pp.503–528. doi:10.1007/BF01589116.

Lok, C.A., van der Houwen, C., ten Kate-Booij, M.J., van Eijkeren, M.A. and Ansink, A.C. (2003). 'Pregnancy after EMA/CO for gestational trophoblastic disease: a report from The Netherlands', *BJOG: An International Journal of Obstetrics and Gynaecology*, 110(6), pp.560–566.

Louppe, G. (2014) *Understanding Random Forests: From Theory to Practice*. PhD thesis. University of Liège. doi:10.13140/2.1.1570.5928.

Lurain, J.R. (2010). 'Gestational trophoblastic disease I: epidemiology, pathology, clinical presentation and diagnosis of gestational trophoblastic disease, and management of hydatidiform mole', *American Journal of Obstetrics and Gynecology*, 203(6), pp. 531–539. doi:10.1016/j.ajog.2010.06.073.

Lurain, J.R. (2011). 'Gestational trophoblastic disease II: classification and management of gestational trophoblastic neoplasia', *American Journal of Obstetrics and Gynecology*, 204(1), pp. 11–18. doi:10.1016/j.ajog.2010.06.072.

Lybol, C., Westerdijk, K., Sweep, F.C.G.J., Ottevanger, P.B., Massuger, L.F.A.G. and Thomas, C.M.G. (2012) 'Human chorionic gonadotropin (hCG) regression normograms for patients with high-risk gestational trophoblastic neoplasia treated with EMA/CO chemotherapy', *Annals of Oncology*, 23(11), pp.2903–2906. doi:10.1093/annonc/mds199.

Lybol, C., Sweep, F.C.G.J., Ottevanger, P.B., Massuger, L.F.A.G. and Thomas, C.M.G. (2013) 'Linear regression of postevacuation serum human chorionic gonadotropin concentrations predicts postmolar gestational trophoblastic neoplasia', *International Journal of Gynecological Cancer*, 23(6), pp.1150–1156. doi:10.1097/IGC.0b013e31829703ea.

Mandrekar, J.N. (2010). 'Receiver operating characteristic curve in diagnostic test assessment', *Journal of Thoracic Oncology*, 5(9), pp.1315–1316. doi:10.1097/JTO.0b013e3181ec173d.

Matherly, L.H. and Goldman, I.D. (2003). 'Membrane transport of folates', *Vitamins and Hormones*, 66, pp.403–456. doi:10.1016/S0083-6729(03)01012-4.

Matsui, H., Iitsuka, Y., Yamazawa, K., Tanaka, N., Seki, K. and Sekiya, S. (2003). 'Changes in the incidence of molar pregnancies: a population-based study in Chiba Prefecture and Japan between 1974 and 2000', *Human Reproduction*, 18(1), pp. 172–175. doi:10.1093/humrep/deg008.

Matsui, H., Kihara, M., Yamazawa, K., Mitsuhashi, A., Seki, K. and Sekiya, S. (2007). 'Recent changes of the incidence of complete and partial mole in Chiba prefecture', *Gynecologic and Obstetric Investigation*, 63(1), pp. 7–10. doi:10.1159/000094388.

McKinney, W. (2010). 'Data structures for statistical computing in Python', *Proceedings of the 9th Python in Science Conference*, pp. 56–61. doi:10.25080/Majora-92bf1922-00a.

McMahon, L.M., Joyce, C.M., Cuthill, L., Mitchell, H., Jabbar, I. and Sweep, F. (2024). 'Measurement of human chorionic gonadotrophin in women with gestational trophoblastic disease', *Gynecologic and Obstetric Investigation*, 89(3), pp. 178–197. doi:10.1159/000531499.

Mitchell, H., Bagshawe, K.D., Newlands, E.S., Savage, P. and Seckl, M.J. (2006). 'Importance of accurate human chorionic gonadotropin measurement in the treatment of gestational trophoblast disease and testicular cancer', *Journal of Reproductive Medicine*, 51(11), pp. 868–870.

Murdoch, S., Djuric, U., Mazhar, B., Seoud, M., Khan, R., Kuick, R., Bagga, R., Kircheisen, R., Ao, A., Ratti, B. et al. (2006). 'Mutations in NALP7 cause recurrent hydatidiform moles and reproductive wastage in humans', *Nature Genetics*, 38(3), pp. 300–302. doi:10.1038/ng1740.

Murphy, K.M., McConnell, T.G., Hafez, M.J., Vang, R., Ronnett, B.M. (2009). 'Molecular genotyping of hydatidiform moles: analytic validation of a multiplex short tandem repeat assay', *Journal of Molecular Diagnostics*, 11(6), pp. 598–605. doi:10.2353/jmoldx.2009.090070.

Nafis, N. (2021). 'The story behind random seed 42 in machine learning', *Geek Culture on Medium*, 13 May. Available at: https://medium.com/geekculture/the-story-behind-random-seed-42-in-machine-learning-b838c4ac290a (Accessed: 14 February 2026).

Natekin, A. and Knoll, A. (2013). 'Gradient boosting machines, a tutorial', *Frontiers in Neurorobotics*, 7, article 21. doi:10.3389/fnbot.2013.00021.

Newlands, E.S., Bagshawe, K.D., Begent, R.H.J., Rustin, G.J.S., Holden, L. and Dent, J. (1986). 'Developments in chemotherapy for medium- and high-risk patients with gestational trophoblastic tumours (1979–1984)', *British Journal of Obstetrics and Gynaecology*, 93(1), pp. 63–69. doi:10.1111/j.1471-0528.1986.tb07815.x.

Ng, A.Y. (2004) 'Feature selection, L1 vs. L2 regularization, and rotational invariance', *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*. doi:10.1145/1015330.1015435.

Ngan, H.Y.S., Seckl, M.J., Berkowitz, R.S., Xiang, Y., Golfier, F., Sekharan, P.K., Lurain, J.R. and Massuger, L. (2018). 'Update on the diagnosis and management of gestational trophoblastic disease', *International Journal of Gynecology & Obstetrics*, 143(S2), pp. 79–85. doi:10.1002/ijgo.12615.

Ngan, H.Y.S., Seckl, M.J., Berkowitz, R.S., Xiang, Y., Golfier, F., Sekharan, P.K., Braga, A. and Garrett, A. (2025). 'Diagnosis and management of gestational trophoblastic disease: 2025 update', *International Journal of Gynecology & Obstetrics*, 171(Suppl. 1), pp. 78–86. doi:10.1002/ijgo.70275.

Nguyen, N.M. and Slim, R. (2014). 'Genetics and epigenetics of recurrent hydatidiform moles: basic science and genetic counselling', *Current Obstetrics and Gynecology Reports*, 3(1), pp. 55–64. doi:10.1007/s13669-013-0076-1.

Nitiss, J.L. (2009). 'Targeting DNA topoisomerase II in cancer chemotherapy', *Nature Reviews Cancer*, 9(5), pp.338–350. doi:10.1038/nrc2607.

Norton, E.C., Dowd, B.E. and Maciejewski, M.L. (2018). 'Odds ratios—current best practice and use', *JAMA*, 320(1), pp.84–85. doi:10.1001/jama.2018.6971.

Osborne, R.J., Filiaci, V., Schink, J.C., Mannel, R.S., Alvarez Secord, A., Kelley, J.L., Provencher, D., Scott Miller, D., Covens, A.L. and Lage, J.M. (2011). 'Phase III trial of weekly methotrexate or pulsed dactinomycin for low-risk gestational trophoblastic neoplasia: A Gynecologic Oncology Group study', *Journal of Clinical Oncology*, 29(7), pp.825–831. doi:10.1200/JCO.2010.30.4386.

Parazzini, F., La Vecchia, C., Franceschi, S., Pampallona, S., Decarli, A., Mangili, G. and Belloni, C. (1985). 'ABO blood-groups and the risk of gestational trophoblastic disease', *Tumori Journal*, 71(2), pp. 123–126. doi:10.1177/030089168507100206.

Parker, V.L., Pacey, A.A., Palmer, J.E., Tidy, J.A., Winter, M.C. and Hancock, B.W. (2017) 'Classification systems in gestational trophoblastic neoplasia – sentiment or evidenced based?', *Cancer Treatment Reviews*, 56, pp.47–57. doi:10.1016/j.ctrv.2017.04.004.

Parker, V.L., Winter, M.C., Tidy, J.A., Hancock, B.W., Palmer, J.E., Sarwar, N., Kaur, B., McDonald, K., Aguiar, X., Singh, K., Unsworth, N., Jabbar, I., Pacey, A.A., Harrison, R.F. and Seckl, M.J. (2023). 'PREDICT-GTN 1: can we improve the FIGO scoring system in gestational trophoblastic neoplasia?', *International Journal of Cancer*, 152(5), pp. 986–997. doi:10.1002/ijc.34352.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2011). 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, 12, pp.2825–2830.

Phianpiset, R., Ruengkhachorn, I., Kuljarusnont, S., Jareemit, N. and Udompunturak, S. (2022). 'Predictive factors associated with resistance to initial methotrexate treatment in women with low-risk gestational trophoblastic neoplasia', *Asia-Pacific Journal of Clinical Oncology*, 18(5), pp.e495–e506. doi:10.1111/ajco.13774.

Powers, D.M.W. (2011) 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation', *Journal of Machine Learning Technologies*, 2(1), pp.37–63.

Probst, P., Wright, M.N. and Boulesteix, A.L. (2019). 'Hyperparameters and tuning strategies for random forest', *WIREs Data Mining and Knowledge Discovery*, 9(3), e1301. doi:10.1002/widm.1301.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A. (2018) 'CatBoost: unbiased boosting with categorical features', *Advances in Neural Information Processing Systems*, 31. doi:10.48550/arXiv.1706.09516.

Rakprasit, C., Ruengkhachorn, I., Therasakvichya, S., Inthasorn, P., Achariyapota, V., Kuljarasnont, S., Khemworapong, K. and Jareemit, N. (2023) 'Combined analysis of clinical features, human chorionic gonadotropin (hCG) value, and hCG ratios for early prediction of postmolar gestational trophoblastic neoplasia', *Archives of Gynecology and Obstetrics*, 307(4), pp.1145–1154. doi:10.1007/s00404-022-06785-w.

Ronnett, B.M. (2018). 'Hydatidiform moles: ancillary techniques to refine diagnosis', *Archives of Pathology & Laboratory Medicine*, 142(12), pp.1485–1502. doi:10.5858/arpa.2018-0226-RA.

Saito, T. and Rehmsmeier, M. (2015). 'The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets', *PLOS ONE*, 10(3), e0118432. doi:10.1371/journal.pone.0118432.

Sakai, K., Wake, N., Fujino, T., Yasuda, T., Kato, H., Fujimoto, S. and Fujinaga, K. (1989). 'Methotrexate-resistant mechanisms in human choriocarcinoma cells', *Gynecologic Oncology*, 34(1), pp.7–11. doi:10.1016/0090-8258(89)90095-4.

Salmi, M., Atif, D., Oliva, D., Abraham, A. and Ventura, S. (2024) 'Handling imbalanced medical datasets: review of a decade of research', *Artificial Intelligence Review*, 57, Article 273.

Savage, P., Williams, J., Wong, S.L., Short, D., Casalboni, S., Catalano, K. and Seckl, M.J. (2010). 'The demographics of molar pregnancies in England and Wales 2000–2009', *The Journal of Reproductive Medicine*, 55(7–8), pp. 341–345.

Savage, P., Sita-Lumsden, A., Dickson, S., Iyer, R., Everard, J., Coleman, R., Fisher, R.A., Short, D., Casalboni, S., Catalano, K. and Seckl, M.J. (2013). 'The relationship of maternal age to molar pregnancy incidence, risks for chemotherapy and subsequent pregnancy outcome', *Journal of Obstetrics and Gynaecology*, 33(4), pp.406–411. doi:10.3109/01443615.2013.771159.

Savage, P., Cooke, R., O'Nions, J., Krell, J., Kwan, A., Camarata, M., Dancy, G., Short, D., Seckl, M.J. and Swerdlow, A. (2015). 'Effects of single-agent and combination chemotherapy for gestational trophoblastic tumors on risks of second malignancy and early menopause', *Journal of Clinical Oncology*, 33(5), pp.472–478. doi:10.1200/JCO.2014.57.5332.

Schimke, R.T., Kaufman, R.J., Alt, F.W. and Kellems, R.F. (1978). 'Gene amplification and drug resistance in cultured murine cells', *Science*, 202(4372), pp.1051–1055. doi:10.1126/science.715457.

Schmitt, C., Doret, M., Massardier, J., Hajri, T., Schott, A.M., Raudrant, D. and Golfier, F. (2013). 'Risk of gestational trophoblastic neoplasia after hCG normalisation according to hydatidiform mole type', *Gynecologic Oncology*, 130(1), pp. 86–89. doi:10.1016/j.ygyno.2013.03.025.

Schober, P. and Vetter, T.R. (2021). 'Logistic regression in medical research', *Anesthesia & Analgesia*, 132(2), pp.365–366. doi:10.1213/ANE.0000000000005247.

Sebire, N.J., Fisher, R.A., Foskett, M., Rees, H., Seckl, M.J. and Newlands, E.S. (2003). 'Risk of recurrent hydatidiform mole and subsequent pregnancy outcome following complete or partial hydatidiform molar pregnancy', *BJOG: An International Journal of Obstetrics and Gynaecology*, 110(1), pp. 22–26.

Seckl, M.J., Sebire, N.J. and Berkowitz, R.S. (2010). 'Gestational trophoblastic disease', *The Lancet*, 376(9742), pp. 717–729. doi:10.1016/S0140-6736(10)60280-2.

Seckl, M.J., Sebire, N.J., Fisher, R.A., Golfier, F., Massuger, L. and Sessa, C. (2013). 'Gestational trophoblastic disease: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up', *Annals of Oncology*, 24(Suppl. 6), pp.vi39–vi50. doi:10.1093/annonc/mdt345.

Shaaban, A.M., Rezvani, M., Haroun, R.R., Kennedy, A.M., Elsayes, K.M., Olpin, J.D., Salama, M.E. and Foster, B.R. (2017). 'Gestational trophoblastic disease: clinical and imaging features', *Radiographics*, 37(2), pp.681–700. doi:10.1148/rg.2017160140.

Sheffield Teaching Hospitals NHS Foundation Trust (2023a). *Human chorionic gonadotrophin (hCG) radioimmunoassay method validation and verification for gestational trophoblastic disease monitoring*. Department of Laboratory Medicine, Sheffield Teaching Hospitals NHS Foundation Trust. Unpublished internal controlled document.

Sheffield Teaching Hospitals NHS Foundation Trust (2023b). *Comparative evaluation of the Sheffield hCG radioimmunoassay and Siemens IMMULITE 2000 hCG assay for gestational trophoblastic disease monitoring*. Department of Laboratory Medicine, Sheffield Teaching Hospitals NHS Foundation Trust. Unpublished internal audit.

Shih, I.M. (2007). 'Gestational trophoblastic neoplasia: pathogenesis and potential therapeutic targets', *The Lancet Oncology*, 8(7), pp. 642–650. doi:10.1016/S1470-2045(07)70204-8.

Shih, I.M. and Kurman, R.J. (1998). 'Epithelioid trophoblastic tumor: a neoplasm distinct from choriocarcinoma and placental site trophoblastic tumor simulating carcinoma', *American Journal of Surgical Pathology*, 22(11), pp. 1393–1403. doi:10.1097/00000478-199811000-00010.

Shih, I.M. and Kurman, R.J. (2001). 'The pathology of intermediate trophoblastic tumors and tumor-like lesions', *International Journal of Gynecological Pathology*, 20(1), pp.31–47. doi:10.1097/00004347-200101000-00004.

Siemens Healthcare Diagnostics Inc. (2025). *IMMULITE 2000 Total hCG: Instructions for Use*. Tarrytown, NY: Siemens Healthcare Diagnostics Inc.

Singh, K., Rollins, S. and Ireson, J. (2024). 'Gestational trophoblastic disease: best practice nursing guidelines', *Gynecologic and Obstetric Investigation*, 89(3), pp. 247–253. doi:10.1159/000530570.

Slim, R. (2024). 'Genetics and genomics of gestational trophoblastic disease', *Hematology/Oncology Clinics of North America*, 38(6), pp. 1219–1232. doi:10.1016/j.hoc.2024.07.004.

Sobell, H.M. (1985). 'Actinomycin and DNA transcription', *Proceedings of the National Academy of Sciences of the United States of America*, 82(16), pp.5328–5331.

Soper, J.T. (2021). 'Gestational trophoblastic disease: current evaluation and management', *Obstetrics & Gynecology*, 137(2), pp.355–370. doi:10.1097/AOG.0000000000004240.

Sperandei, S. (2014) 'Understanding logistic regression analysis', *Biochemia Medica*, 24(1), pp.12–18. doi:10.11613/BM.2014.003.

Stenman, U.H. (2004). 'Standardization of assays for human chorionic gonadotropin', *Clinical Chemistry*, 50(5), pp. 798–800. doi:10.1373/clinchem.2003.031013.

Steyerberg, E.W. (2019). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. 2nd edn. Cham: Springer.

Subbaswamy, A. and Saria, S. (2020). 'From development to deployment: dataset shift, causality, and shift-stable models in health AI', *Biostatistics*, 21(2), pp.345–352. doi:10.1093/biostatistics/kxz041.

Szabolcsi, Z., Demeter, A., Kiraly, P. *et al.* (2021). 'Epigenetic dysregulation of trophoblastic gene expression in gestational trophoblastic disease', *Biomedicines*, 9(12), Article 1935. doi:10.3390/biomedicines9121935.

Tham, B.W., Everard, J.E., Tidy, J.A., Drew, D. and Hancock, B.W. (2003). 'Gestational trophoblastic disease in the Asian population of Northern England and North Wales', *BJOG: An International Journal of Obstetrics and Gynaecology*, 110(6), pp. 555–559.

Tibshirani, R. (1996) 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), pp.267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

Tidy, J.A., Gillespie, A.M., Bright, N., Radstone, C.R., Coleman, R.E. and Hancock, B.W. (2000). 'Gestational trophoblastic disease: a study of mode of evacuation and subsequent need for treatment with chemotherapy', *Gynecologic Oncology*, 78(3), pp.309–312. doi:10.1006/gyno.2000.5839.

Tidy, J., Seckl, M.J. and Hancock, B.W. (2021). 'Management of gestational trophoblastic disease: Green-top Guideline No. 38', *BJOG: An International Journal of Obstetrics and Gynaecology*, 128(3), pp. e1–e27. doi:10.1111/1471-0528.16266.

Topol, E.J. (2019) *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. New York: Basic Books.

UK Government (2021). *The Data Protection, Privacy and Electronic Communications (Amendments etc) (EU Exit) Regulations 2019: UK General Data Protection Regulation*. Available at: https://www.legislation.gov.uk/eur/2016/679/contents (Accessed: 14 February 2026).

Vaisbuch, E., Ben-Arie, A., Dgani, R., Perlman, S., Sokolovsky, N., Hagay, Z. and Shalev, E. (2009). 'Twin pregnancy consisting of a complete hydatidiform mole and co-existent fetus: report of two cases and review of literature', *Gynecologic Oncology*, 112(3), pp.590–594. doi:10.1016/j.ygyno.2008.10.002.

van Trommel, N.E., Massuger, L.F.A.G., Schijf, C.P.T., ten Kate-Booij, M.J., Sweep, F.C.G.J. and Thomas, C.M.G. (2006). 'Early identification of resistance to first-line single-agent methotrexate in patients with persistent trophoblastic disease', *Journal of Clinical Oncology*, 24(1), pp. 52–58. doi:10.1200/JCO.2005.03.3043.

Veal, G.J., Cole, M., Errington, J., Parry, A., Hale, J., Pearson, A.D., Howe, K., Chisholm, J.C., Beane, C., Brennan, B., Waters, F., Glaser, A., Hemsworth, S., McDowell, H., Wright, Y., Pritchard-Jones, K., Pinkerton, R., Jenner, G., Nicholson, J., Elsworth, A.M. and Boddy, A.V. (2005). 'Pharmacokinetics of dactinomycin in a pediatric patient population: A United Kingdom Children's Cancer Study Group study', *Clinical Cancer Research*, 11(16), pp.5893–5899. doi:10.1158/1078-0432.CCR-04-2546.

Vejerslev, L.O. (1991). 'Clinical management and diagnostic possibilities in hydatidiform mole with coexistent fetus', *Obstetrical & Gynecological Survey*, 46(9), pp. 577–588. doi:10.1097/00006254-199109000-00001.

Veras, E., Kurman, R.J., Wang, T.L. and Shih, I.M. (2017). 'PD-L1 Expression in Human Placentas and Gestational Trophoblastic Diseases', *International Journal of Gynecological Pathology*, 36(2), pp.146–153. doi:10.1097/PGP.0000000000000305.

Visentin, M., Zhao, R. and Goldman, I.D. (2012). 'The antifolates', *Hematology/Oncology Clinics of North America*, 26(3), pp.629–648. doi:10.1016/j.hoc.2012.02.002.

Wang, Q., Fu, J., Hu, L., Fang, F., Xie, L., Chen, H., He, F. and Wu, T. (2017). 'Prophylactic chemotherapy for hydatidiform mole to prevent gestational trophoblastic neoplasia', *Cochrane Database of Systematic Reviews*, 2017(9), CD007289. doi:10.1002/14651858.CD007289.pub3.

Widemann, B.C. and Adamson, P.C. (2006). 'Understanding and managing methotrexate nephrotoxicity', *The Oncologist*, 11(6), pp.694–703. doi:10.1634/theoncologist.11-6-694.

Wilimitis, D. and Walsh, C.G. (2023). 'Practical considerations and applied examples of cross-validation for model development and evaluation in health care: tutorial', *JMIR AI*, 2, e49023. doi:10.2196/49023.

Winter, M.C. (2021). 'Treatment of low-risk gestational trophoblastic neoplasia', *Best Practice & Research Clinical Obstetrics & Gynaecology*, 74, pp. 67–80. doi:10.1016/j.bpobgyn.2021.01.006.

Wong, S.C.C., Chan, A.T.C., Chan, J.K.C. and Lo, Y.M.D. (2006). 'Nuclear beta-catenin and Ki-67 expression in choriocarcinoma and its pre-malignant form', *Journal of Clinical Pathology*, 59(4), pp. 387–392. doi:10.1136/jcp.2005.026666.

World Medical Association (2013). 'World Medical Association Declaration of Helsinki: ethical principles for medical research involving human subjects', *JAMA*, 310(20), pp.2191–2194. doi:10.1001/jama.2013.281053.

Wynants, L., Van Calster, B., Collins, G.S., Riley, R.D., Heinze, G., Schuit, E., Bonten, M.M.J., Dahly, D.L., Damen, J.A.A., Debray, T.P.A., de Jong, V.M.T., De Vos, M., Dhiman, P., Haller, M.C., Harhay, M.O., Henckaerts, L., Heus, P., Kammer, M., Kreuzberger, N., Lohmann, A., Luijken, K., Ma, J., Martin, G.P., McLernon, D.J., Andaur Navarro, C.L., Reitsma, J.B., Sergeant, J.C., Shi, C., Skoetz, N., Smits, L.J.M., Snell, K.I.E., Sperrin, M., Spijker, R., Steyerberg, E.W., Takada, T., Tzoulaki, I., van Kuijk, S.M.J., van Bussel, B., van der Horst, I.C.C., van Royen, F.S., Verbakel, J.Y., Wallisch, C., Wilkinson, J., Wolff, R., Hooft, L., Moons, K.G.M. and van Smeden, M. (2020) 'Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal', *BMJ*, 369, m1328. doi:10.1136/bmj.m1328.

Wu, X., Qin, J., Shen, T., Fei, W., Chen, L., Xie, X. and Lu, W. (2020) 'The 16-year experience in treating low-risk gestational trophoblastic neoplasia patients with failed primary methotrexate chemotherapy', *Journal of Gynecologic Oncology*, 31(4), e36. doi:10.3802/jgo.2020.31.e36.

Xu, G., Chakraborty, C. and Lala, P.K. (2001). 'Expression of TGF-β signaling genes in the normal, premalignant, and malignant human trophoblast: loss of Smad3 in choriocarcinoma cells', *Biochemical and Biophysical Research Communications*, 287(1), pp. 47–55. doi:10.1006/bbrc.2001.5533.

Xue, W.C., Chan, K.Y., Feng, H.C., Chiu, P.M., Ngan, H.Y., Tsao, S.W. and Cheung, A.N. (2004). 'Promoter hypermethylation of multiple genes in hydatidiform mole and choriocarcinoma', *Journal of Molecular Diagnostics*, 6(4), pp.326–334. doi:10.1016/S1525-1578(10)60528-4.

You, B., Pollet-Villard, M., Fronton, L., Labrousse, C., Schott, A.M., Hajri, T., Girard, P., Freyer, G., Tod, M., Tranchand, B., Colomban, O., Ribba, B., Raudrant, D., Massardier, J., Chabaud, S. and Golfier, F. (2010). 'Predictive values of hCG clearance for risk of methotrexate resistance in low-risk gestational trophoblastic neoplasias', *Annals of Oncology*, 21(8), pp.1643–1650. doi:10.1093/annonc/mdq033.

You, B., Harvey, R., Henin, E., Mitchell, H., Golfier, F., Savage, P.M., Tod, M., Wilbaux, M., Freyer, G. and Seckl, M.J. (2013). 'Early prediction of treatment resistance in low-risk gestational trophoblastic neoplasia using population kinetic modelling of hCG measurements', *British Journal of Cancer*, 108(9), pp.1810–1816. doi:10.1038/bjc.2013.123.

You, B., Bolze, P.A., Lotz, J.P., Massardier, J., Gladieff, L., Joly, F., Hajri, T., Maucort-Boulch, D., Bin, S., Rousset, P. et al. (2020). 'Avelumab in patients with gestational trophoblastic tumors with resistance to single-agent chemotherapy: Cohort A of the TROPHIMMUN phase II trial', *Journal of Clinical Oncology*, 38(27), pp.3129–3137. doi:10.1200/JCO.20.00803.

Zhang, T. (2004) 'Solving large scale linear prediction problems using stochastic gradient descent algorithms', *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, p.116. doi:10.1145/1015330.1015332.

Zhao, R. and Goldman, I.D. (2003). 'Resistance to antifolates', *Oncogene*, 22(47), pp.7431–7457. doi:10.1038/sj.onc.1206946.

7.0 Appendix

Sheffield Teaching Hospitals NHS Foundation Trust, Weston Park Cancer Hospital internal protocols for GTD Treatment.

Chemotherapeutic Regimens

1. Low-Risk Patients

The methotrexate (MTX) with folinic acid rescue regimen is the standard initial treatment.

- Methotrexate: 50 mg IM on alternate days (Days 1, 3, 5, 7)
- Folinic acid: 15 mg orally, 24 hours post-MTX dose (Days 2, 4, 6, 8)
- Rest period: 7 days between treatment cycles

Low-dose MTX is effective in eradicating residual trophoblastic tissue in patients with low tumour burden. Its favourable toxicity profile, ease of administration, and high cure rates make it the regimen of choice in this cohort. The inclusion of folinic acid reduces the risk of mucosal and haematological toxicity without compromising efficacy.

Common side effects and management:

- Oral mucositis → Analgesic mouthwash (e.g., Difflam)
- Conjunctivitis → Hypromellose eye drops
- Pleurisy/peritoneal pain (serositis) → Simple analgesia (avoid NSAIDs)
- Methotrexate excretion may be altered by other drugs (e.g., antibiotics), so co-medications must be reviewed.

2. Salvage of Drug Resistance in Low-Risk Patients

Approximately 20–30% of low-risk patients develop methotrexate resistance. If resistance occurs and hCG <10,000 IU/L, patients are treated with single agent actinomycin D. If hCG >10,000 IU/L, combination chemotherapy with actinomycin D and etoposide is initiated.

a. When hCG <10,000 IU/L:

- Dactinomycin 1.25 mg/m^2 IV every 2 weeks
- Dexamethasone 8 mg IV pre-chemotherapy
- Granisetron 3 mg IV

Post-chemotherapy supportive meds:

- Dexamethasone 4 mg BD for 3 days
- Ondansetron 8 mg BD for 2 days

Dactinomycin is a highly active agent in methotrexate-resistant GTN and can achieve remission with relatively low toxicity. Its use as monotherapy in patients with lower hCG levels (<10,000 IU/L) offers a well-tolerated and effective alternative.

Common side effects:

- Emesis
- Fatigue
- Mild bone marrow suppression

b. When hCG >10,000 IU/L:

- Dactinomycin 0.5 mg IV on Days 1, 2, 3
- Etoposide 100 mg/m^2 IV on Days 1, 2, 3
- Granisetron 3 mg IV
- Dexamethasone 8 mg IV

Supportive medication:

- Dexamethasone 4 mg BD for 3 days post-chemotherapy
- Ondansetron 8 mg BD for 2 days post-chemotherapy
- Metoclopramide 10 mg every 6–8 hours as required

Patients with higher tumour burden and rising hCG levels are at greater risk of progression and require combination chemotherapy. The addition of etoposide provides synergistic cytotoxic activity, improving treatment efficacy in resistant cases.

Common side effects:

- Dactinomycin: nausea, vomiting
- Etoposide: alopecia, increased risk of infection and bone marrow suppression

3. High-Risk Patients

High-risk GTN requires prompt and aggressive treatment due to the increased tumour burden and higher risk of metastatic spread. The EMA regimen (etoposide, methotrexate, and actinomycin D) is internationally recognised as a first-line treatment for high-risk GTN and has demonstrated high remission and cure rates. Methotrexate's role as a folate antagonist is central to its anti-trophoblastic effect, while etoposide and actinomycin D provide additional cytotoxic synergy. The regimen is also dose-dense, allowing for maximum tumouricidal effect with manageable toxicity. Adherence to the two-week cycle is essential to preserve schedule-dependent efficacy.

Standard treatment for high-risk GTN is the EMA regimen, administered every two weeks.

Day 1:

- Dexamethasone 8 mg IV
- Granisetron 3 mg IV
- Sodium bicarbonate 2 g QDS orally to maintain urinary pH >7
- Actinomycin D 0.5 mg IV bolus
- Etoposide 100 mg/m^2 IV (1-hour infusion in 500 mL normal saline)
- Methotrexate 300 mg/m^2 IV (12-hour infusion in 1 L normal saline)

Day 2:

- Dexamethasone 8 mg IV
- Granisetron 3 mg IV
- Actinomycin D 0.5 mg IV bolus
- Etoposide 100 mg/m^2 IV (1-hour infusion)
- Folinic acid 15 mg every 6 hours starting 24 hours after methotrexate initiation (total 8 doses, IV or oral)
- Filgrastim 0.5 MU/kg/day SC may be required based on neutrophil count

Common side effects:

- Methotrexate: mucositis, nephrotoxicity, hepatotoxicity
- Etoposide: alopecia, myelosuppression, nausea
- Actinomycin D: nausea, mucositis
- Filgrastim: bone pain
- Pulmonary haemorrhage risk is increased in patients with lung metastases

4. Initial Treatment of Ultra High-Risk (UHR) Disease

Ultra-high-risk disease (FIGO score ≥13) is associated with increased early mortality, particularly due to haemorrhage, tumour lysis, and multi-organ failure. A gentle induction chemotherapy strategy is employed using low-dose etoposide and cisplatin (EP) to reduce tumour burden gradually and safely.

Regimen:

- Etoposide 100 mg/m^2 (Days 1 and 2)
- Cisplatin 20 mg/m^2 (Days 1 and 2)
- Weekly for 1–2 cycles before standard EMA or EP-EMA

This strategy minimises the risk of fatal complications by preventing rapid tumour breakdown and associated haemorrhage or sepsis. It is particularly valuable in patients with large tumour burden or critical organ involvement. The EP regimen provides an essential bridge to curative therapy while reducing initial treatment-related risk.

Common Side Effects:

- Cisplatin: Nephrotoxicity, ototoxicity, electrolyte disturbances (Mg^{2+}, K^+), nausea
- Etoposide: Myelosuppression, alopecia
- Both agents: Risk of infection, fatigue, gastrointestinal upset

5. Treatment of Established CNS Disease

EP-EMA Regimen with Intrathecal Methotrexate

Arm A (EP):

- Etoposide 150 mg/m^2 IV Day 1
- Cisplatin 75 mg/m^2 IV Day 1
- Dexamethasone 12 mg PO
- Akynzeo 300 mg PO pre-chemotherapy

Supportive Medications (Arm A):

- Metoclopramide 10 mg every 6–8 hours PRN
- Dexamethasone 4 mg twice daily for 3 days post-chemotherapy

Arm B (EMA):

- Dactinomycin 0.5 mg IV bolus Day 8
- Etoposide 100 mg/m^2 IV Day 8
- Methotrexate 1000 mg/m^2 IV over 24 hours Day 8
- Dexamethasone 8 mg IV
- Granisetron 3 mg IV

Supportive Medications (Arm B):

- Metoclopramide 10 mg every 6–8 hours PRN
- Dexamethasone 4 mg twice daily for 3 days post-chemotherapy
- Ondansetron 8 mg twice daily for 2 days post-chemotherapy
    - (Omitted if the patient has no nausea and is constipated, as ondansetron can exacerbate constipation)
- Sodium bicarbonate 2 g orally QDS and as required to maintain urine pH >7 during MTX administration
    - (If oral bicarbonate fails, 500 mL of 1.4% or 1.26% polyfusor bicarbonate solution is infused over 4 hours)
- Calcium folinate (leucovorin rescue): 15 mg orally every 6 hours starting 32 hours after beginning methotrexate, for a total of 8 doses
- Intravenous hydration: Commence 6 hours prior to methotrexate and continue during infusion
    - If the patient is unable to maintain oral intake after methotrexate, IV hydration should continue for 8 additional hours

Intrathecal Methotrexate:

- Methotrexate 12.5 mg intrathecally
- Administered on alternate weeks from systemic chemotherapy
- No intravenous chemotherapy should be given on the same day as intrathecal methotrexate
- EMA is administered in the week alternate to the intrathecal dose
- Continue until serum and CSF hCG levels have normalised

CNS involvement in GTN, particularly with choriocarcinoma or PSTT, significantly increases morbidity and mortality. The standard EMA regimen is insufficient to penetrate the blood-brain

barrier, necessitating modification. High-dose systemic methotrexate (1000 mg/m^2) increases CNS bioavailability. Additionally, intrathecal methotrexate bypasses the blood-brain barrier, achieving therapeutic levels directly in the CSF. Combined EP-EMA offers synergistic cytotoxicity while covering systemic disease.

The comprehensive supportive regimen mitigates the high toxicity risk:

- Dexamethasone reduces cerebral oedema, prevents chemotherapy-induced nausea, and serves as an anti-inflammatory during MTX treatment.
- Granisetron, ondansetron, and metoclopramide offer layered antiemetic protection. Granisetron is highly effective for 5HT3 receptor-mediated nausea; ondansetron extends coverage post-infusion; metoclopramide addresses breakthrough nausea via dopamine receptor antagonism.
- Sodium bicarbonate and hydration are crucial to prevent methotrexate-induced nephrotoxicity by alkalinising urine and maintaining renal perfusion.
- Calcium folinate (leucovorin rescue) protects healthy cells from high-dose methotrexate toxicity, particularly mucosal and marrow suppression.

6a. Salvage of Drug Resistance in High-Risk Patients

When high-risk patients develop resistance to first-line EMA therapy or experience relapse, salvage chemotherapy is required. No regimen guarantees cure in these cases, and a combination of chemotherapy, surgery, and potentially stereotactic radiosurgery may be employed.

a. EP-EMA Regimen (Alternating Weekly)

EP (Week 1 – Arm A):

- Etoposide 150 mg/m^2 IV Day 1
- Cisplatin 75 mg/m^2 IV Day 1
- Dexamethasone 12 mg PO (1 hour before chemotherapy)
- Akynzeo 300 mg PO (1 hour before chemotherapy)

Supportive Medications (Arm A):

- Metoclopramide 10 mg every 6–8 hours PRN for breakthrough nausea
- Dexamethasone 4 mg twice daily for 3 days post-chemotherapy to reduce inflammation, prevent nausea, and manage delayed emesis

- Ondansetron 8 mg twice daily for 2 days (optional but often used in combination with Akynzeo)
- Ensure hydration and renal protection during and after cisplatin administration

Arm B – EMA (Week 2):

- Dactinomycin 0.5 mg IV bolus, Day 1
- Etoposide 100 mg/m^2 IV over 1 hour, Day 1
- Methotrexate 300 mg/m^2 IV over 12 hours, Day 1
- Dexamethasone 8 mg IV
- Granisetron 3 mg IV
- Sodium bicarbonate 2 g orally QDS and PRN to maintain urine pH >7
    - If oral bicarbonate is insufficient, administer 500 mL of 1.26% bicarbonate polyfusor IV over 4 hours

Supportive Medications (Arm B):

- Intravenous hydration beginning 6 hours before MTX and continuing during and after infusion if patient is unable to maintain oral intake
- Calcium folinate (leucovorin rescue): 15 mg orally every 6 hours × 8 doses, beginning 32 hours after methotrexate initiation
- Metoclopramide 10 mg every 6–8 hours PRN
- Dexamethasone 4 mg twice daily for 3 days post-chemotherapy
- Ondansetron 8 mg twice daily for 2 days post-chemotherapy
- Monitor renal and liver function regularly throughout

The EP-EMA regimen introduces cisplatin, a DNA crosslinking agent that bypasses resistance mechanisms associated with prior exposure to methotrexate or etoposide. Alternating weeks of EP and EMA allow continued cytotoxic pressure with manageable toxicity profiles. Schedule adherence is critical, and the regimen is supported by evidence demonstrating improved survival in patients with refractory or relapsed GTN.

Supportive medications are essential for maintaining hydration, managing gastrointestinal and haematological toxicity, and mitigating nephrotoxic effects of cisplatin and methotrexate.

Common Side Effects:

Cisplatin: Nephrotoxicity, Ototoxicity, Nausea/vomiting, Electrolyte abnormalities (hypomagnesaemia, hypokalaemia)

Etoposide: Myelosuppression, Alopecia, Secondary leukaemia risk (dose-dependent)

Methotrexate: Mucositis, Nephrotoxicity (requires urine alkalinisation and hydration), Hepatotoxicity

Dactinomycin: GI toxicity, Emesis, Hepatic enzyme elevation

Supportive agents:

- Dexamethasone: Anti-inflammatory, antiemetic, prevents delayed nausea
- Akynzeo: Combination of netupitant (NK1 antagonist) and palonosetron (5HT3 antagonist) – highly effective for acute and delayed chemotherapy-induced nausea
- Granisetron/Ondansetron: 5HT3 receptor antagonists – control acute nausea and vomiting
- Metoclopramide: Dopamine antagonist – effective for breakthrough nausea
- Leucovorin: Prevents excessive folate depletion from MTX – protects normal tissues
- Sodium bicarbonate: Maintains alkaline urine to prevent MTX crystallisation in renal tubules
- Hydration and electrolyte replacement: Reduce renal injury, particularly from cisplatin and high-dose MTX

6b. TP-TE Regimen (28-day cycle)

The TP-TE regimen is reserved for patients with resistant or relapsed high-risk GTN, particularly those who have failed previous EMA or EP-EMA protocols. This is a dose-intensive platinum-taxane-based regimen delivered over a 28-day cycle in two phases.

TP (Day 1 of Cycle):

Pre-medication (required to prevent hypersensitivity and gastrointestinal side effects):

- Dexamethasone 20 mg PO – 12 hours and 6 hours before paclitaxel
- Famotidine 40 mg PO – 3 hours before chemotherapy
- Akynzeo 300 mg PO – 1 hour before chemotherapy
- Dexamethasone 8 mg IV – immediately before chemotherapy

- Chlorpheniramine 10 mg IV – immediately before chemotherapy

Chemotherapy (TP arm):

- Paclitaxel 135 mg/m^2 IV
- Cisplatin 60 mg/m^2 IV
- Filgrastim 0.5 MU/kg/day SC from Day 5 for 5 consecutive days

TE (Day 15 of Cycle):

Pre-medication (repeat protocol):

- Dexamethasone 20 mg PO – 12 hours and 6 hours before paclitaxel
- Famotidine 40 mg PO – 3 hours before chemotherapy
- Chlorpheniramine 10 mg IV
- Granisetron 3 mg IV
- Dexamethasone 8 mg IV – immediately before chemotherapy

Chemotherapy (TE arm):

- Paclitaxel 135 mg/m^2 IV
- Etoposide 150 mg/m^2 IV
- Filgrastim 0.5 MU/kg/day SC from Day 19 for 5 days

Supportive Medications (across the cycle):

- Metoclopramide 10 mg every 6–8 hours PRN (antiemetic support)
- Dexamethasone 4 mg BD for 3 days post each arm (to prevent delayed nausea/inflammation)
- Ondansetron 8 mg BD for 2 days post-chemotherapy (optional in combination with Akynzeo)
- IV hydration during and after cisplatin administration to reduce nephrotoxicity
- Magnesium and potassium replacement as needed – cisplatin causes electrolyte loss
- Filgrastim support to prevent neutropenic complications

The TP-TE regimen is employed as third-line or salvage therapy for patients with high-risk GTN who are unresponsive to EMA or EP-EMA. It incorporates paclitaxel, a microtubule inhibitor that disrupts mitotic spindle formation, and cisplatin, a potent DNA crosslinker. These agents work synergistically with etoposide to target chemoresistant trophoblastic cells.

TP-TE is also effective for GTN with poor prognosis histologies such as PSTT and ETT, especially in the metastatic setting. Filgrastim support allows for dose-intensity maintenance despite profound myelosuppression risks. The regimen's complexity necessitates strict adherence to supportive protocols.

Common Side Effects:

Paclitaxel: Peripheral neuropathy (dose-limiting), Myelosuppression (neutropenia, anaemia), Alopecia, Hypersensitivity reactions (necessitates steroid and antihistamine pre-medication)

Cisplatin: Nephrotoxicity – requires aggressive hydration and electrolyte monitoring, Ototoxicity, Severe nausea and vomiting, Hypomagnesaemia, hypokalaemia, Etoposide: Myelosuppression, Nausea, Risk of secondary leukaemia (with high cumulative doses)

Filgrastim: Bone pain, Injection site reactions

Dexamethasone and antiemetics: Hyperglycaemia, mood disturbance, insomnia (dexamethasone), Headache, constipation (ondansetron), Extrapyramidal symptoms (rare with metoclopramide)

6c. High-Dose Chemotherapy with Autologous Stem Cell Transplantation (HDCT + ASCT)

There is no standardised high-dose chemotherapy (HDCT) protocol universally adopted for GTN; however, selected centres may employ HDCT with autologous stem cell transplantation (ASCT) in patients with multi-refractory GTN, particularly those who have failed multiple salvage regimens (e.g., EMA, EP-EMA, TP-TE).

Commonly used agents in HDCT protocols include:

- Carboplatin, Etoposide, and Cyclophosphamide (or other alkylating agents) at myeloablative doses
- Followed by autologous peripheral blood stem cell infusion to restore haematopoiesis

Supportive care involves:

- Strict infection prophylaxis
- Granulocyte colony-stimulating factors (e.g., Filgrastim)
- Antiemetics

- Blood product support
- Total parenteral nutrition if mucositis is severe

HDCT with ASCT is considered a last-line intervention for patients with extensive chemoresistance who would otherwise have no curative options. The rationale is to deliver myeloablative chemotherapy doses that exceed standard marrow tolerance, with stem cell rescue enabling bone marrow recovery. While data are limited to small series and case reports, this approach has led to sustained remissions in select patients. However, its use must be highly individualised, based on patient fitness, response to prior regimens, and extent of disease.

7. Placental Site Trophoblastic Tumour (PSTT) and Epithelioid Trophoblastic Tumour (ETT)

Management Strategy:

- Localised Disease (Stage I):
- Primary treatment: Total abdominal hysterectomy
- Uterine-sparing surgery may be attempted in select fertility-preserving cases; however, multifocal microscopic disease is common, and recurrence risk is higher
- Adjuvant chemotherapy (e.g. EP-EMA) may be considered if:
    - High-risk features are present (e.g. >48 months from causative pregnancy)
    - Myometrial or lymphovascular invasion is extensive

Metastatic or Advanced-Stage Disease:

- First-line chemotherapy:
    - EP-EMA or
    - TP-TE (as previously detailed)
- Surgical resection of residual metastatic or uterine disease is strongly recommended after chemotherapy
    - Procedures may include: hysterectomy, pulmonary resection, or resection of isolated metastatic sites

PSTT and ETT are rare subtypes of GTD arising from intermediate trophoblasts. They differ from choriocarcinoma in that:

- They typically produce low or mildly elevated hCG levels
- They are less sensitive to chemotherapy
- They often present months to years after the index pregnancy
- Metastases are more likely to involve lymph nodes and distant organs

Due to the relative chemoresistance of these tumours, surgical excision is the mainstay of treatment for early-stage disease. In patients with advanced, metastatic, or poor-risk disease, platinum-based multi-agent regimens such as EP-EMA or TP-TE are necessary to achieve control. Surgery is then used to eradicate sites of residual or chemo-resistant disease.

8. Pembrolizumab in Drug-Resistant Gestational Trophoblastic Neoplasia (GTN)

Indications and Dosing (as commissioned by NHS England):

Pembrolizumab (PD-1 inhibitor) is administered at a dose of 200 mg IV every 3 weeks. It is used in the following specific scenarios:

- Second-line treatment for patients with high-risk or ultra-high-risk choriocarcinoma who progress during, or relapse after, first-line combination chemotherapy
- Third-line treatment for patients with low-risk GTN (as classified at original presentation) who have relapsed following:
    - First-line single-agent chemotherapy
    - Second-line combination chemotherapy
- Fourth-line treatment for low-risk GTN after failure of:
    - First-line single-agent chemotherapy
    - Second-line single-agent chemotherapy
    - Third-line combination chemotherapy
- Poor prognosis PSTT or ETT:
    - As part of second-line treatment for inoperable stage IV disease, or
    - Following maximal response or unacceptable toxicity to platinum-based chemotherapy
    - Can also be used adjuvantly after surgery in poor-prognosis patients (defined by an interval ≥4 years from the causative pregnancy)

Gestational trophoblastic tumours express programmed death-ligand 1 (PD-L1), enabling them to evade immune surveillance by binding to the PD-1 receptor on T-cells. Pembrolizumab, a PD-

1 immune checkpoint inhibitor, blocks this interaction and restores T-cell mediated tumour destruction.

In patients with GTN resistant to multiple chemotherapy lines, pembrolizumab offers a non-cytotoxic, immune-based alternative. While large-scale data are limited, responses have been documented in published case series, particularly in choriocarcinoma and chemo-resistant PSTT/ETT. The therapy is formally commissioned by NHS England under strict criteria, underscoring its clinical utility and regulatory endorsement.