Supporting Medical Decision-Making Using Machine Learning

Richard James Wainwright

A thesis submitted in partial fulfilment of the requirements of Sheffield Hallam University for the degree of Doctor of Philosophy

January 2023

Summary

As the strain on health care continues to grow worldwide, the need for reliable decision-making has never been more apparent. The computerisation of electronic health records has provided a wealth of data that can be applied to various medical use cases. Machine Learning algorithms are exploited to try and assist with making effective decisions. The resulting contributions within this work demonstrate that it is possible to lean on advancements in computer science to develop support tools for medical practitioners which assist in their decision-making processes.

This thesis contributes four core advances to the research domain: Firstly the enhancement of current mortality prediction systems in intensive care units was considered. Comparing multiple Machine Learning classifiers with optimised pipelines produced results that were both comparable and more effective at determining patient mortality than the existing APACHE II model. The most encouraging classifier was Decision Trees whilst being trained using: K-fold cross validation, Grid search hyper-parameter tuning and SMOTE achieving an average AUROC score of 0.93 and accuracy of 0.92. Unlike other mortality prediction systems which are often trained on small cohorts of data, a method of retraining and optimising for different patient cohorts is introduced. Retraining based on a patients age or admission in to the ICU is also considered as a novel approach of keeping support tools up to date.

An ensemble imputation method has been developed that can be used to generate the missing data in a real life dataset. This has produced accuracy and recall results comparable to current state of the art techniques when applied to the Cleveland hospital dataset.

In this work, strategies to rebalance datasets are investigated to predict early onset Sepsis. One promising approach examined in this thesis is the use of the RUSboost algorithm. This enabled the optimisation of a classifier that has a high fidelity without overfitting.

Acknowledgments

I would like to thank my parents for their unwavering support, love, and "can do" attitude that they have passed on to me whilst also encouraging me to try and excel at everything I do. I'd also like to thank my brother Chris, whose gym sessions and support has kept my mind focused throughout this work. I am very grateful and lucky to have you.

I would also like to express thanks to my director of studies, Dr Alex Shenfield, for his advice, ideas and for the opportunity to do this post graduate research within a fantastic institution that has made me feel welcome and part of the team throughout. I am also extremely grateful for my friends who have been so supportive and interested in the work I had undertaken, asking lots of question and suggesting alternative ways of looking at problems.

Finally. I express thanks to Sheffield Hallam University and everyone within C3RI for their financial support.

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.

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

3. I am aware of and understand the University's policy on plagiarism and 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.

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.

5. The word count of the thesis is **30,279** word's.

Name	Richard James Wainwright
Date	January 2023
Award Type.	PhD
Faculty	Industry & Innovation Research Institute
Director(s) of Studies	Dr. Alex Shenfield

The only true wisdom is in knowing you know nothing.

- Socrates

Contents

List of Acronyms

AB	AdaBoost	166
AD	Alzheimer's Disease	67
AGI	Artificial General Intelligence	12
AHC	Agglomerative Hierarchical Clustering	47
AI	Artificial Intelligence	1
AIDS	Acquired Immune Deficiency Syndrome	87
AIMS	Artificial Intelligence Mortality Score	85
ANN	Artificial Neural Network	36
ANNs	Artificial Neural Networks	36
APACHE	Acute Physiology and Chronic Evaluation	81
APACHE II	Acute Physiology and Chronic Evaluation II	3
AR	Augmented Reality	70
AUROC	Area-Under the Receiver Operating Characteristic Curve	81
AWS	Amazon Web Services	182
BMI	Body Mass Index	20
CAT	Computerized Tomography	68
CHD	Coronary Heart Disease	144
CNN	Convolutional Neural Networks	85
COVID-19	Coronavirus Disease	71
CPR	Cardiopulmonary Resuscitation	100
CQC	Care Quality Commission	3
DCNN	Deep Convolutional Neural Networks	69
DHC	Divisive Hierarchical Clustering	48
EEG	Electroencephalography	67
EHRs	Electronic Health Records	48
EIM	Ensemble Imputation Method	7
ELN	Electronics Lab Notebook	65
EMR	Electronic Medical Record(s)	165
ERR	Error Rate	91

ERM	Empirical Risk Minimisation	23
EWS	Early Warning Systems	158
FCNN	Fully Connected Neural Network	160
FDA	Food and Drug Administration	87
FN	False Negative	91
FP	False Positive	91
FPR	False-Positive Rate	92
GDPR	General Data Protection Regulation	76
GPU	Graphics Processing Units	16
HCA	Hierarchical Clustering Analysis	47
ICNARC	Intensive Care National Audit & Research Centre	4
ICU	Intensive Care Unit	3
IFCs	Infection-Fighting Chemicals	158
k-NN	K-Nearest Neighbour	164
kg	Kilograms	20
lb	Pounds	20
LDA	Linear Discriminant Analysis	50
LIMS	Lab Information Management Systems	65
LSTM	Long-Short Term Memory	85
MAR	Missing At Random	134
MCAR	Missing Completely At Random	134
MCP	McCulloch-Pitts Neuron	36
MHRA	Medicine and Healthcare Product Regulatory Agency(s)	65
MIS	Minimally Invasive Surgery	70
ML	Machine Learning	1
MNAR	Missing Not At Random	134
MRI	Magnetic Resonance Imaging	68
NGA	Niche Genetic Algorithm	49
NHS	National Health Service	80
NLP	Natural Language Processing	12
NN	Neural-Network	37

PCA	Principal Component Analysis	49
PRE	Precision	93
QDA	Quadratic Discriminant Analysis	88
qEEG	quantitative Electroencephalography	67
REC	Recall	93
ReLU	Rectified Linear Unit	40
RBF	Radial Basis Function	34
RF	Random Forest	167
ROC	Receiver Operating Characteristic	82
RR	Random Removal	151
RUS	Randomly-Under Sampled	165
SAPS II	2nd Simplified Acute Physiology Score	81
SMOTE	Synthetic Minority Oversampling Techniques	88
SOFA	Sequential Organ Failure Assessment	83
SPL	Suspicious pigmented lesions	68
SR	Systematic Removal	151
SVM	Support Vector Machine	33
SVMs	Support Vector Machines	33
TN	True Negative	91
TP	True Positive	91
TPOT	Tree-based Pipeline Optimization Tool	182
TPR	True-Positive Rate	92
TRIPOD	The Transparent Reporting of a multi-variable prediction model for Individual Prognosis or Diagnosis	66
UK	United Kingdom	144
US	United States	71
USA	United States of America	87

List of Tables

List of Figures

Chapter 1

Introduction

Machine Learning (ML) is a subset of Artificial Intelligence (AI). It is the collective title given to a group of sophisticated modelling techniques capable of modelling extremely complex functions (Singh, 2018a; Petersson, 2021). These modelling techniques are now being applied to a range of problem domains within finance, science, and engineering industries (Frankenfield, 2022a).

Anywhere that there are problems with prediction, classification, or regression, ML techniques can be utilised. However, these modelling techniques contain a large number of potential complex error surfaces. Possible errors can include: local minima, plateaus in the search of landscape, and saddle points. These errors make the process of training models difficult and time consuming and means great care needs to be taken in setting model architectures and parameters. This training process constitutes a crucial part of AI as the performance of these models is dependent on both the training and data provided to them.

Due to the improvements of computers and the reduction in the cost of high end graphical processing units and processing power, AI has seen a resurgence of popularity in recent years and many industries are turning to these techniques to solve complex problems (Singer, 2022). Medicine is one such field of research that is looking to utilise ML to aid the industry (Frankenfield, 2022a).

Due to the ever changing nature of medicine, it is important that research and technologies continue to advance, to compliment such developments within the industry. ML can be utilised as a key driver in the evolution of new and effective support tools. ML can either be used alongside existing support tools to further confirm decisions, or optimised ML pipelines, can in some instances, out perform many of the existing support tools. As a consequence to the vast amount of decisions needed within medicine and the evolving nature of patient cohorts, it has never been more important to swiftly develop ML tools that can assist medical practitioners.

This thesis will look to build on the existing research undertaken on three different medical datasets, to investigate and develop support tools for medical practitioners, whilst also introducing some novel approaches to assist in training models.

1.1 Research Aim

The research undertaken in this body of work overlaps and spans several different disciplines including ML, Data Analysis and Medicine. These areas are combined to develop novel pipelines and techniques that are applied to accurately support medical decision making. The new techniques introduced can be generalised and applied to many different ML tasks, including those that have missing data - a significant challenge for many real-world applications of ML.

This work proposes to address the gap in research by focusing on applying novel ML techniques to the field of medicine on three specific medical datasets that are currently utilised, these datasets are from areas where the current standards are no longer within acceptable tolerances or outdated. It is demonstrated that the ML methodologies, novel pipelines and novel techniques proposed in this body of work can be used as alternative or additional techniques to make patient decisions. The aim of this research is to show that ML techniques can be applied to support medical practitioners in making informed medical decisions for specific scenarios. A key contribution in support of this aim includes the development of a new technique of dealing with missing data as an alternative to existing commonly used imputation methods. Moreover, a new method of predicting mortality in intensive care units is introduced, including the development of an ML pipeline utilising different techniques to rebalance unbalanced datasets. An extensive literature review of current and state of the art ML techniques will also be performed.

Throughout this thesis, various ML techniques will be used in a knowledge discovery process to accurately support medical decision making. The four general research questions addressed in this work are the following:

1. Is it possible to predict the mortality of a patient admitted in to the Intensive Care Unit (ICU) more accurately than the current Acute Physiology and Chronic Evaluation II (APACHE II) tool or the Care

Quality Commission (CQC) Intensive Care National Audit & Research Centre (ICNARC) benchmark when you are using the ICNARC dataset?

2. How can you handle changing patient cohorts admitted in to ICU over time using the same ICNARC dataset?

3. The method presented in this thesis build on existing knowledge within the ML domain by developing a novel method of data imputation using ensemble techniques. This imputation method is applied to the Cleveland Heart Disease classification dataset.

4. What is the most effective way of re-balancing the "PhysioNet Computing in Cardiology Challenge" sepsis datasets?

1.2 Research Objectives

The main objectives of the PhD are given below:

1. Design and develop a novel method of determining mortality within an ICU using the same data collected within the ICNARC dataset.

2. Apply new online ML techniques to deal with different patient cohorts when cohorts are defined by splitting up a dataset by either date of submission to ICU or the age of the patient.

3. Introduce a novel method of dealing with missing data by using ensemble methods.

4. Apply ML rebalancing techniques to allow early detection of sepsis

1.3 Key Contributions

The main contributions of this thesis are:

- **Development of an effective tool to predict mortality in ICU.** A ML model was developed that can outperform current state-of-the-art techniques in predicting mortality. Furthermore, it has been shown to outperform the current standards used in hospitals – the APACHE II and ICNARC scores.

- **Online Machine Learning implementation of mortality prediction.** Utilising online learning approaches it has been possible to demonstrate the benefits of retraining support tool ML models over time (Wainwright and Shenfield, 2023).

- **Applying a novel method of data imputation to predict heart disease.** The ensemble imputation method developed and described in this report has proven it is possible to combine imputation methods in order to effectively fill missing data.

- **6 hour earlier sepsis prediction tool.** Effectively re-balancing a dataset to produce repeatable results is one of the most difficult tasks undertaken by data scientists. The combination of these re-balancing techniques and ML models has allowed the development of a model that can determine the onset of sepsis 6 hours before current methods.

Additional contributions that have arisen from this body of work but have not been described in this thesis are:

- **Human Activity Recognition Making Use of Long-Short Term Memory Techniques.** Using an open source dataset, a model to predict human activities and postural transitions has been developed. (Wainwright and Shenfield, 2019).

- **Building Actionable Personas Using Machine Learning Techniques.** Children survey data from 22 countries globally was utilised to develop personas that can be used by marketing professionals to better understand their target audience. Different clustering algorithms were introduced and a novel way of segmenting children was developed (Farrukh et al., 2022).

1.4 Thesis Structure

Chapter 2 presents a thorough review of ML, including theoretical concepts, different types of learning, methods and a general taxonomy of AI. A brief history of AI is also given, and some of the key issues with choosing the correct approach to use are discussed. Chapter 2 also presents an in depth introduction to different commonly used classifiers with many different techniques and concepts introduced. The current state of ML and how it is currently used in the medical domain is outlined in chapter 3.

Chapter 4 uses the techniques introduced in Chapter 2 to accurately predict the mortality of patients in ICU. The model is developed using a dataset provided by the ICNARC that has electronic health records for patients admitted into ICU in a selection of London Hospitals between 2012 and 2014. The full ML pipeline is examined with techniques for standardising, sampling and modelling the problem experimented on. The results are compared with pre-existing literature and the APACHE II and ICNARC scores. Patient cohorts, medicines, and treatments change dramatically over time. As a direct result, existing support tools can become quickly outdated and provide inaccurate results. Building on the knowledge introduced in Chapter 2 and the concept of online learning, a real time ML model is developed that can be trained over different time periods.

Chapter 5 builds on the limited techniques described in literature for dealing with missing data. An in depth description of the different types of missing data is conveyed before a complete dataset has data removed using different combinations of systematic and random removal techniques. The Ensemble Imputation Method (EIM) technique is introduced and applied to detect heart disease in patients where missing data is introduced. Chapter 6 compares the outcomes of different dataset re-balancing techniques before hypothesising and demonstrating a new method of re-balancing datasets. The new method builds on existing techniques and combines them together. An early onset sepsis detection dataset that is heavily imbalanced is utilised.

Finally, Chapter 7 presents the conclusions for this thesis and also outlines some potential further work and research that could be worthy of investigation.

Chapter 2

Review Of Machine Learning

2.1 Introduction

This Chapter will provide a thorough review of the field of AI with an emphasis on fundamental ML concepts. Section 2.2 provides an introduction into AI and describes some of the key research discoveries and historical points to date. After giving an outline of AI, some common advantages and disadvantages are discussed.

ML is introduced in section 2.7 with an explanation of some of the core theoretical concepts and a discussion of popular classification techniques. There are many applications for ML, and some common uses in medicine are outlined throughout this Chapter. Some medical domain uses of ML techniques are then discussed in section 3.

2.2 A Short History of Artificial Intelligence

> *"We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best" (Turing, 1950)*[1]

Emulating computer systems can learn, reason, and self-correct in order to complete many different tasks - in some cases more quickly and accurately than humans. These tasks can include providing personalised music recommendations, translation of languages, and recognising speech (Rouse and Botelho, 2018; Childs, 2011). AI was inspired by the work undertaken by Alan Turing, Norbert Weiner, and Claude Shannon which showed that it might be possible to develop an electronic brain using existing knowledge of neurology and how the human brain uses electronic signals (McCorduck and Cfe, 2004; Crevier, 1993b).

AI is a varied field of study, from applications trained to play board games all the way through to complex machines that can carry out difficult classification tasks (such as detecting cancers in X-rays and the detection of obstacles for autonomous vehicles). AI has become intrinsic in the way we interact with computers in modern society with AI techniques built-in to aspects of everyday life (e.g. computational photography in mobile phones and smart digital assistants) (Lewis, 2014).

[1] A quote taken from Alan Turing's published work from 1950 called "Computing Machinery and Intelligence"

Concepts used in modern AI can be seen throughout civilisation over the last 1800 years where initially philosophers attempted to describe human thinking by introducing a symbolic system. However, it was not until the 1950s that the field of AI was formalised. *"Can machines learn?"* - is a question taken from the paper 'Computer Machinery and Intelligence' published in 1950 by Alan Turing (Turing, 1950). The paper proposed "The Imitation Game" which later developed into "The Turing Test" a measure of the ability of a machine/computers to think in a human-like fashion (Crevier, 1993a).

The term "Artificial Intelligence" was introduced in 1956 by John McCarthy when the first academic conference on the topic was held at Dartmouth College in Hanover, New Hampshire. An attendee at the conference, Marvin Minsky, is quoted as saying *"Within a generation [. . .] the problem of creating AI will substantially be solved" (Crevier, 1993a).* AI research continued to develop in to the 1960s with the creation of new programming languages, robots, and automatons. Science fiction cinema frequently showed artificially intelligent beings with TV shows and movies growing in popularity over the following decade, resulting in researchers being more attracted to study in the field.

Regardless of the well-funded global research effort over several decades, computer science researchers found it increasingly difficult to develop intelligent machines. In order to create successful applications, such as computer vision, there was a requirement for powerful machines capable of processing enough data. At the time these machines were not readily available, and the hardware was not capable of what was required. As a direct result, governments lost faith in AI and saw it as a lost cause. Therefore, from the

1970s to the early 1990s, an *"AI Winter"* took place where researchers dealt with a shortage of funding (Schuchmann, 2019).

The three main advances of AI in the last sixty years have been (Lewis, 2014):

1. Search algorithms
2. Machine Learning algorithms
3. Statistical analysis in understanding the world at large

Eugene Goostman is the name given to the chatbot that successfully passed The Turing Test in 2014 by convincing 33% of the panel that they were having a conversation with a real boy for 5 minutes (Warwick and Shah, 2015). Although this accomplishment hasn't been without controversy, with AI experts saying that only a third of the panel were fooled and that the bot was allowed to not answer a number of questions by claiming English was a second language (Sample and Hern, 2014). The Turing Test is widely recognised as insufficient for measuring intelligence in machines, as it only considers external behaviour. There is a wider field of research in assessing machine intelligence, this research considers a complete measure of machine intelligence and not just an updated version of the Turing Test (Menager, 2018; Aron, 2015).

The adoption of AI has been slow and has only begun to see real growth and improvement since 2003, with the introduction of more advanced computer systems. These systems are capable of handling vast amounts of data and are able to solve complex mathematical calculations in a timely manner

(Smith, 2006). Exponential gains in computer processing power and the increasing availability of cheap computer storage has meant that large technology companies such as (but not limited to) Amazon, Google, and Baidu use AI techniques to their commercial advantage. Potential applications include enabling them to monitor potential customers by targeting adverts and understanding consumer behaviour when shopping and interacting online.

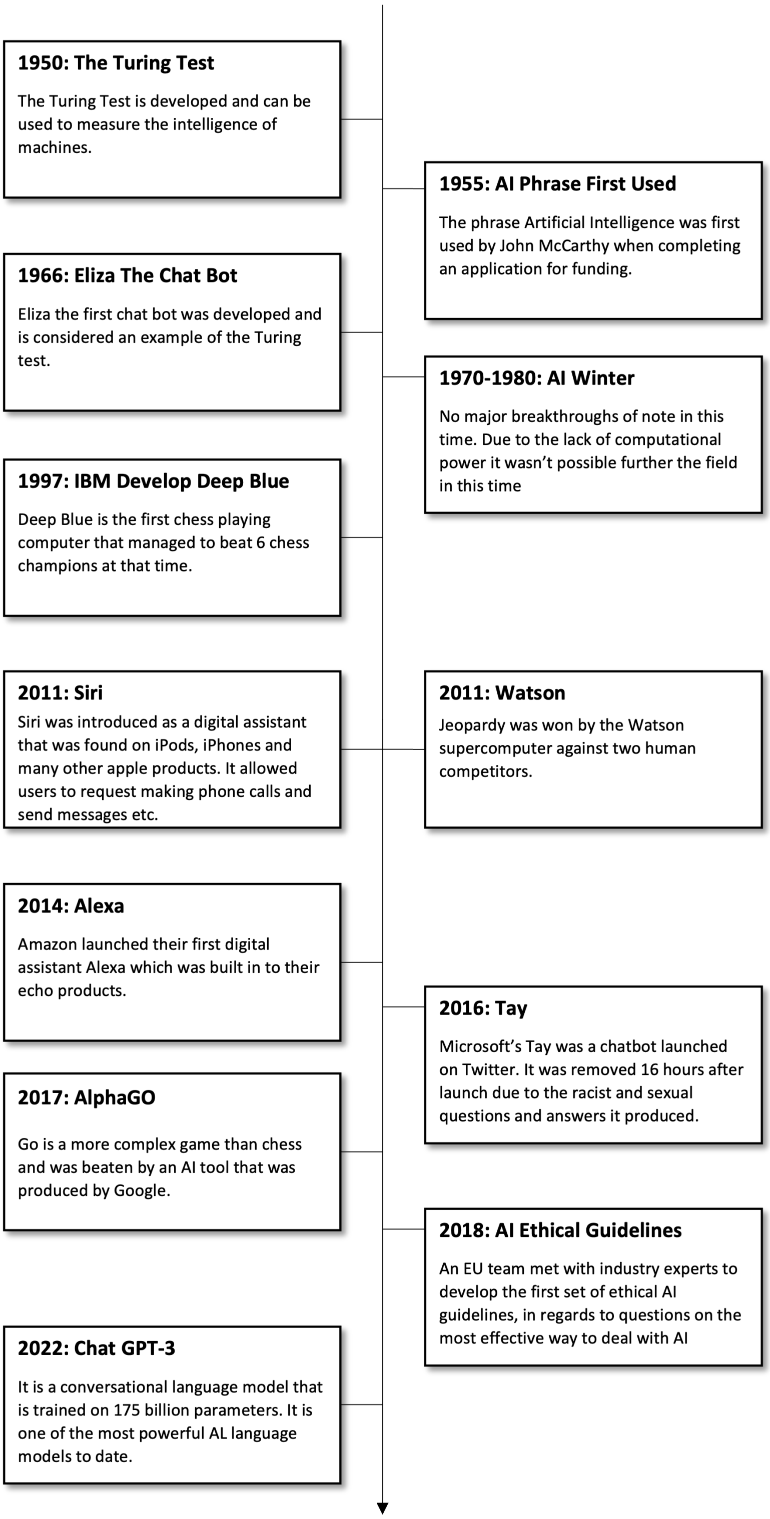
Weak AI describes models that are trained to solve a very specific problem and, in many cases, can outperform human capabilities. DeepMind's AlphaGo and Deep Blue from IBM are examples of Weak AI systems that are capable of playing board-games better than human players (Han et al., 2019). Deep Blue achieved this feat in 1997 when it defeated the Russian Grandmaster Garry Kasparov. However, AI systems are not flexible and cannot be applied to a different problem once developed (Goodrich, 2021).

The theoretical concept of Strong AI sometimes referred to as Artificial General Intelligence (AGI) describes systems that have the flexibility of humans and can combine this flexibility with the advantages of a computer by storing large amounts of data. This combination could result in more reliable answers and reduced risks (Wang, 2019) by making use of clustering and associations to process data and not just classification to find the more appropriate answer (Wang, 2019). However, currently strong AI doesn't exist, it is just a theoretical form of machine intelligence and researchers disagree as to whether such systems are even possible (Frankenfield, 2022b).

In 2022, AI advances were introduced at an unparalleled rate. Somethings that will be improved in 2023 are chatbots, as increased Natural Language Processing (NLP) abilities pave the way for intelligent apps and virtual assistants to understand the users requests and issues. ML models will move to Auto ML without the need for programmers to create specific models. Furthermore, more jobs will make use of AI to complete everyday tasks which can help drive efficiencies and streamline businesses. Lin (2020) state that there had been a 270% growth in businesses turning to AI in the four years before 2019 and this number continues to increase in 2023 with the AI industry expected to be valued in excess of $267 Billion by 2027.

2.3 Machine Learning

The term "Machine Learning" was first introduced in 1959 by Arthur Samuel whilst working for IBM. He produced one of the worlds first successful computer based self-learning programs and developed a model that had the ability to play checkers. The 'Samuel Checkers-playing program' is seen as a key development in ML research, and introduced the concept of beta pruning of search trees (Samuel, 1959; Weiss, 2003).



1950: The Turing Test

The Turing Test is developed and can be used to measure the intelligence of machines.

1955: AI Phrase First Used

The phrase Artificial Intelligence was first used by John McCarthy when completing an application for funding.

1966: Eliza The Chat Bot

Eliza the first chat bot was developed and is considered an example of the Turing test.

1970-1980: AI Winter

No major breakthroughs of note in this time. Due to the lack of computational power it wasn't possible further the field in this time

1997: IBM Develop Deep Blue

Deep Blue is the first chess playing computer that managed to beat 6 chess champions at that time.

2011: Siri

Siri was introduced as a digital assistant that was found on iPods, iPhones and many other apple products. It allowed users to request making phone calls and send messages etc.

2011: Watson

Jeopardy was won by the Watson supercomputer against two human competitors.

2014: Alexa

Amazon launched their first digital assistant Alexa which was built in to their echo products.

2016: Tay

Microsoft's Tay was a chatbot launched on Twitter. It was removed 16 hours after launch due to the racist and sexual questions and answers it produced.

2017: AlphaGO

Go is a more complex game than chess and was beaten by an AI tool that was produced by Google.

2018: AI Ethical Guidelines

An EU team met with industry experts to develop the first set of ethical AI guidelines, in regards to questions on the most effective way to deal with AI

2022: Chat GPT-3

It is a conversational language model that is trained on 175 billion parameters. It is one of the most powerful AL language models to date.

Figure 2.1: A Timeline Charting Key Developments in Artificial Intelligence

In recent years, ML has advanced from the studies of computational learning theory and simple pattern recognition techniques. It is now a category of computer algorithms that can learn from historical data and accurately predict outcomes without the need of being programmed explicitly (Samuel, 1959) depending on the problem domain it is applied to. ML-based learning algorithms can discover hidden patterns and features embedded within the data. The analytical models that are produced allow both data scientists and computer analysts to make informed, valid, and reliable decisions and results (Sarker, 2021).

2.4 Current Approaches in Machine Learning

Predictive modelling and data mining are similar processes to those undertaken by ML. These three methods make use of vast datasets and search through them looking for patterns, adjusting the outputs in the process accordingly. As well as personalised marketing (as mentioned in the previous section), ML is commonly used in:

- Fraud detection
- Network security threat detection
- Building recommendation systems

All of these applications use the past behaviour of the user to build up a picture where patterns can be identified. They can then infer suggestions based on this past behaviour. An example of ML that is used frequently is a spam filter on an email inbox. Both spam and non-spammed emails are easily classified using ML techniques (Dada et al., 2019). Examples of both types of email are fed in to the ML algorithm, which will identify patterns that allow for the prediction of their type. This then leads to the creation of a rule that can be used with future emails. Future emails will be tested with the accurate prediction rule and classified accordingly based on the results of the algorithm (Angra and Ahuja, 2017).

Figure 2.2 presents an overview of the ML process. The primary sections are: Data Input, Feature Extraction, and Model selection. The Feature Extraction part of the ML process is one of the most important, as it aids in the production of producing an accurate model, by selecting attributes in the dataset that are most relevant to making good predictions. This process will identify and remove any attributes that are unneeded, irrelevant, or redundant as they do not contribute to the accuracy of the model and can in some case reduce the accuracy dramatically (Shaikh and Ali, 2019; Brownlee, 2015).

Unlike dimensionality reduction methods such as principal component analysis and Sammons mappings which try to combine attributes in the data to reduce their size, feature extraction methods include and exclude variables in the data without changing them (Shaikh and Ali, 2019).

ML has gone through a renaissance in recent years, with more companies looking to introduce some form of ML or AI into their business and business processes. With the improvement of Graphics Processing Units (GPU), improvements in computational data handling, and the accumulation of company records it is possible to produce accurate models to make predictions with very reliable outcomes. Applying ML in this way has trends and companies can adapt their business processes accordingly.

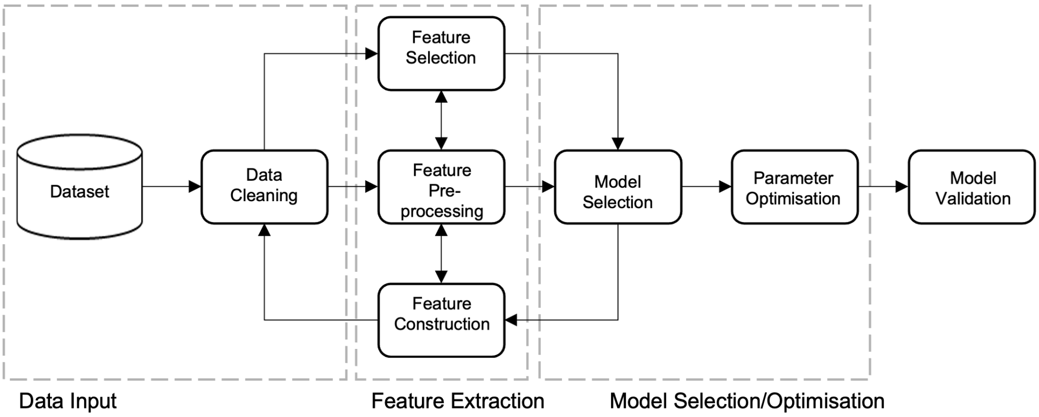


Figure 2.2: Structure Of Machine Learning Process

2.5 Taxonomy Of Machine Learning Methods

There are three main variations of ML algorithms: Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning. The following sections describe these variations and give basic examples. The main focus in this literature review will be on supervised and un-supervised techniques of ML, as they are the most relevant to the applications considered in this program of research. The principal difference between supervised and unsupervised learning is that supervised learning makes use of the ground truth labels – that is, we have existing knowledge of what the output variable of the given sample should be. As a direct result, the goal of supervised learning is to spot the pattern between the dependent variables and the true output by finding the best approximation for the relationship. Unlike supervised learning, in un-supervised the labels are not used to spot patterns in the data, so the algorithm has to infer the natural structure that is present in the data points (Wakefield, 2022).

2.6 Types of Learning Algorithm

As illustrated in figure 2.2, all ML design problems begin with a dataset. The main focus of the process, once the data has been standardised and missing data is dealt with, will be to select and refine a mathematical model that captures the dynamics of the problem. Performance bounds can be presented based on the optimised algorithm for the model (under the assumption that the dataset is a large enough distribution of the full ground truth data). "Which algorithm should I use?" is a common question asked by data scientists when looking at a new problem, especially with the variety of ML algorithms available. New algorithms and techniques are being developed at a rapid rate. The main factors that will affect the model selection are (Li, 2017):

1. Size, quality, and nature of data
2. Computational time that is available for training the model
3. What you want to achieve with the data in terms of metrics and outcomes

Selecting the correct ML algorithm in order to produce the best results is one of the more difficult challenges facing data scientists. Figure 2.3 shows the decision process to select which algorithm to use depending on the dataset, type of data, and desired output. Accuracy, ease of use, and required training time are aspects of ML that should always be considered when choosing an algorithm.

The selection of the most effective algorithm will be reliant on both the type of learning problem and also what is known about the data. Looking at

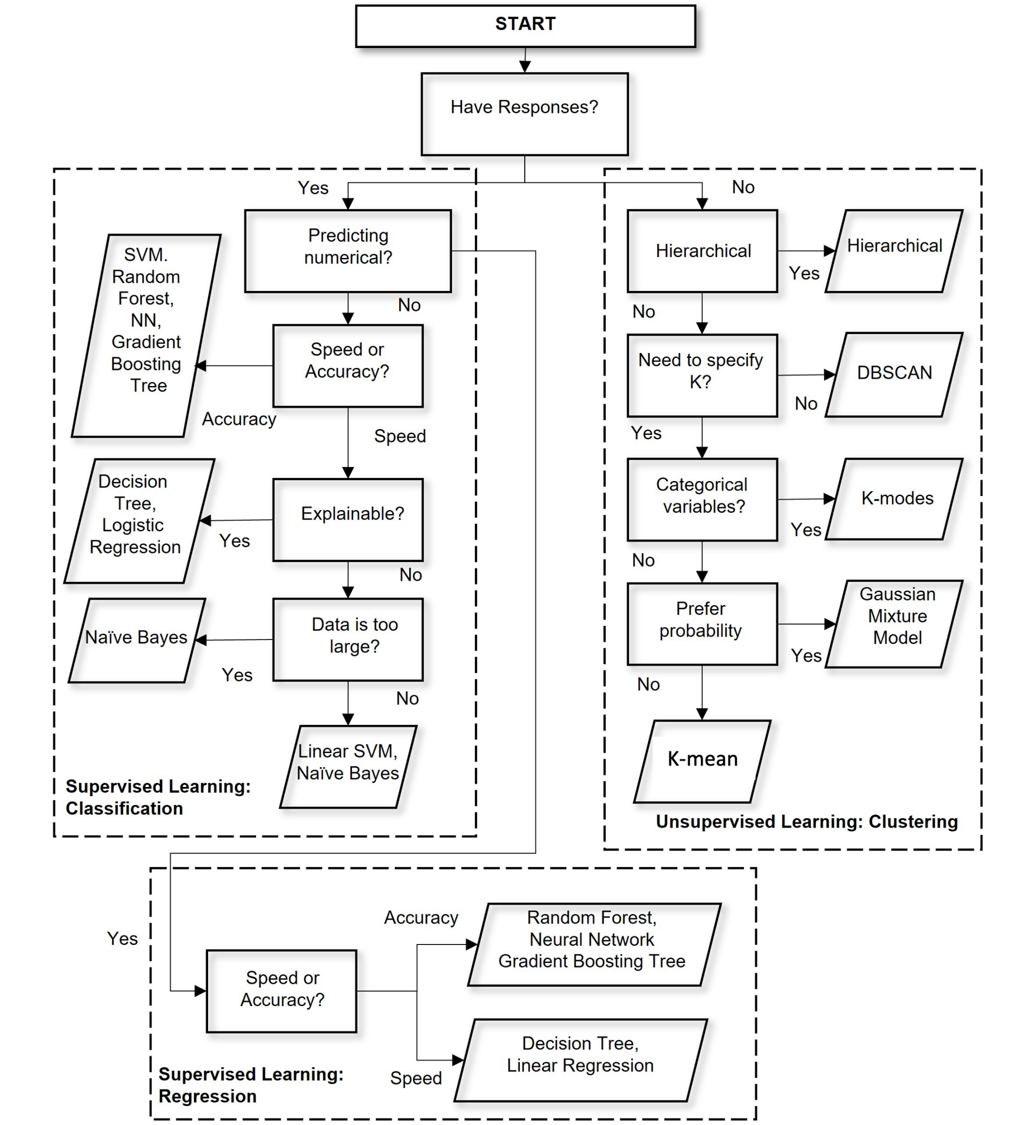


Figure 2.3: Which Machine Learning Algorithm Should Be Implemented Based On The Metrics And Taxonomy Of Data?

the individual algorithms makes it easier to understand what they provide and how they can be implemented (Dey, 2016). The following sections provide details on some common statistical modelling algorithms for both supervised and un-supervised problems. Further descriptions of any additional algorithms used later in this body of work will be described as used.

2.7 Supervised Learning

In Supervised learning a dataset will be a collection of labelled examples that can be denoted as:

$$(x_i,\ y_i)_{i=1}^N \tag{2.1}$$

Feature vectors are each individual element x_i amongst N. Features are described mathematically as x^j. A feature vector is a single vector in which the each dimension j=1,. . . , D will contain a value that will describe that example, for instance age of a patient or if they are a smoker or not. For all the examples in a given dataset j the feature vectors will always contain the same type of information. The finite number of classes that all of these labels belong to is referred to as y_i for each element.

Consider, as an example, the classification of a patient into the Body Mass Index (BMI) categories: underweight, normal, overweight, obese, extremely obese as per Table 2.1. A patients height in cm, age, and weight in Kilograms (kg) or Pounds (lb) would be provided as x^1, x^2, x^3, x^4 with the feature defined at position j being the same for all patients; for example $x_i^{(1)}$ will represent a patients weight in kg therefore $x_k^{(2)}$ will also contain weight information and so on for the whole series of data (Burkov, 2019).

Height			Weight									
			Underweight		Normal		Overweight		Obese		Extreme Obese	
in	ft-in	cm	KG	lbs	KG	lbs	KG	lbs	KG	lbs	KG	lbs
58”	4ft 10”	147.3	35 - 40.2	77 - 89	40.2 - 52	91 - 115	54.5 - 63.6	119 - 138	65.9 - 84.1	143 -185	88.6+	191+
5911	4ft 11”	149.9	36 - 41.5	79 - 94	41.5 - 54	94 -119	56.2 - 65.2	124 - 143	68.2- 88.6	148 - 193	90.9+	196+
60”	5ft	152.4	37.2 - 43	82 - 97	45.5 - 56.8	97 - 123	59.1 - 68.2	128 - 148	70.5 - 90.9	153 - 199	93.2 +	204+
61”	5ft 1”	154.9	38.5 - 45.5	85- 99	45.5 - 59.1	100 - 127	61.4 - 70.5	132 - 153	72.7- 93.2	158 - 206	96.5+	211+
62”	5ft 2”	157.4	39.6 - 45.5	87 - 104	47.7 - 61.4	104 - 131	63.6 - 72.7	136 - 158	75.0- 97.7	164 - 213	99.0+	218+
63”	5ft 3”	160	41.0 - 45.5	90 - 107	47.7 - 63.6	107 - 135	65.9 - 75.0	141 - 163	77.3 - 100	169 - 220	102.3+	225+
64”	5ft 4”	162.5	42.2 - 47.7	93 - 110	50.0 - 65.9	110 - 140	68.2 - 77.3	145 - 169	79.5 - 103	174 - 227	106.8+	232+
65”	5ft 5”	165.1	43.5 - 50.0	96 - 114	52.3 - 65.9	114 - 141	68.2 - 79.5	150 - 174	81.8 - 106.2	180 - 234	109.1+	240+
66”	5ft 6”	167.6	45.0- 50.0	99 -118	52.3 - 68.2	118 - 148	70.5 - 84.1	155 - 179	86.4 - 109.5	186 - 241	111.6+	247+
67’	5ft 7”	170.1	46.2 -52.3	102 - 121	54.5 - 70.5	121 - 153	72.7 - 86.4	159 - 185	88.6 - 112.9	191 - 249	115.7+	255+
68”	5ft 8”	172.7	47.7 - 54.5	105 - 125	56.8 - 72.7	125 - 158	75.0 - 88.6	164 - 190	90.9 - 116.2	197 - 256	119.2+	262+
69”	5ft 9”	175.2	49 - 56.8	108 - 128	59.1 - 75.0	128 - 162	77.3 - 90.9	169 - 196	93.2 - 119.6	201 - 263	122.7+	270+
70”	5ft 10”	177.8	50.5 - 56.8	112 - 132	59.1 - 77.3	132 - 167	79.5 - 93.2	174 - 202	96.5 - 123.2	209 - 271	126.3+	278+
71”	5ft 11”	180.3	52 - 59 1	115 - 136	61.4- 79.5	136- 172	81.8 -96.5	179 - 208	97.7 - 126.7	215 - 279	130+	286+
72”	6ft	182.8	53.3 - 61.4	118 - 140	63.6 - 81.8	140 - 177	84.1 - 100	184 - 213	102.3 - 130.2	221 - 287	133.5+	294+
73”	6ft 1	185.4	55 - 61.4	121- 144	63.6 - 84.1	144 - 182	86.4 - 102.3	189 - 219	104.5 - 134	227 - 295	137.5+	302+
74”	6ft 2”	187.9	56.5 - 63.6	125 - 148	65.9 - 86.4	148 - 186	88.6 - 104.5	194 - 225	106.8 - 137.6	233 - 303	141.1+	311+
75”	6ft 3”	190.5	58 - 65.9	128 - 152	68.2 - 88.6	152 - 192	90.9 - 106.8	200 - 232	109.1 - 141.4	240 - 311	145+	319+
76”	6ft 4”	193	59.5 - 69.5	131 - 154	70.5 - 90.9	156 - 197	93.2 - 107.9	205 -238	111.6 - 145.2	246 - 320	149+	328+
		BMI Ranges	16 - 18.5		18.5 -24 9		25 - 29.9		30 - 39.9		Over 40	

Table 2.1: BMI Height to Weight Age Chart in Kg and lbs

The main objective of Supervised learning is to find a mapping from the feature vector of labelled data to a target output. Supervised learning algorithms are trained by using labelled data (i.e where the output is already known). In supervised learning, each data sample corresponds to a target and the model is trained to find a set of "rules" to arrive at that target from the input features. The model "rules" will be constantly updated until it has been trained on all the data. Current ML techniques have achieved great success; however, it is important to note that in many real-world applications it is difficult to obtain reliable ground truth labels due to the high cost involved in fully labelling the data.

Supervised learning is used in many different areas of data science including:

- Bioinformatics
- Natural Language Processing (NLP)
- Computer vision applications

The most widely used supervised learning algorithms are: K-Nearest Neighbours, Neural Networks, Support Vector Machines, and Linear Regression (Burkov, 2019).

Supervised learning can be broadly divided into two tasks:

- **Classification:** When supervised learning is used to predict a categorical variable this is known as classification. If there are only two categorical labels, then it is described as binary classification.

- **Regression:** Unlike in classification, regression is used when predicting continuous values (Fumo, 2017).

2.7.1 Empirical Risk and Structural Risk Minimisation

Empirical risks - Empirical Risk Minimisation (ERM) is a method for finding a model that performs well on a given dataset. Given a set of training examples, the empirical risk of a model is defined as the average loss over all the examples in the dataset. The goal of ERM is to find the model that minimizes the empirical risk, or equivalently, maximizes the average performance on the training set. This is done by choosing the model that minimizes the following objective function:

$$\arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i; \theta)) \tag{2.2}$$

In this formula, θ represents the parameters of the model, $f(x; \theta)$ is the prediction made by the model for a given input x, and $L(y, f(x; \theta))$ is a loss function that measures the error between the prediction $f(x; \theta)$ and the true label y. The sum is over all N training examples in the dataset, and the average is taken by dividing by N. By minimizing this objective function, we find the model parameters θ that give the lowest average loss over all the training examples. This in turn gives us a model that is expected to perform well on the training set, and therefore, on unseen examples from the same distribution as the training set (Li, 2021).

Structural risks - In minimising the empirical risk, the model is susceptible to overfitting. Overfitting occurs because the supervised learning model has too much flexibility (e.g too many degrees of freedom). This can result in it being a bad candidate function for unknown data points, as the mapping focuses on noise in the dataset rather than the actual data. In figure 2.4, the solid and dotted-line both represent functions that reduce the empirical risk to 0; however, the dotted-line is a classifier as it generalises better and therefore might be more effective at predicting unknown data points.

Structural risk minimisation is used to prevent a supervised learning model from overfitting the data. An effective way of determining λ is to use cross validation techniques where the training data is divided into multiple sets and part of the data is used to train the model with the performance of the model tested on a validation set. For each iteration the penalty is adjusted to find the value for λ which minimises the risk (Zhang, 2010).

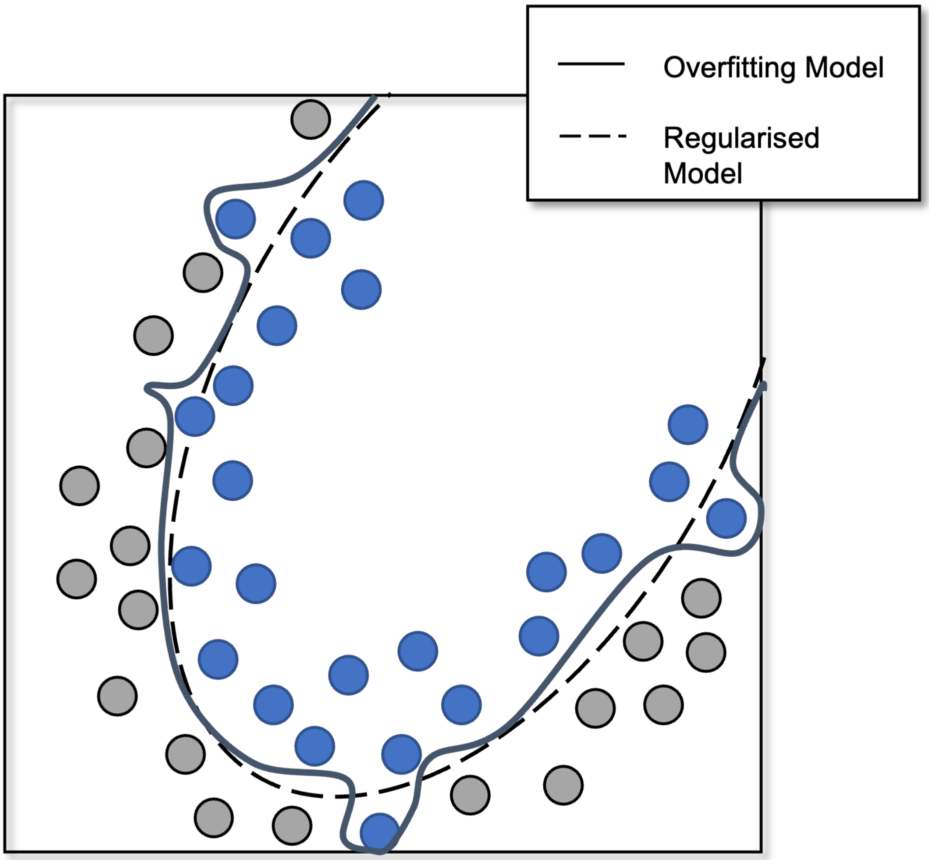


Figure 2.4: Diagram To Demonstrate Overfitting Of Datapoints

2.7.2 Linear Regression

Developed in the field of statistics, linear regression finds the relationship between input and output variables. It is frequently used and is probably the simplest ML algorithm.

Linear regression has been used to solve statistical problems for more than 200 years. It has been widely researched and many academic papers use it in one form or another. Fundamentally linear regression is a model that finds the linear relationship between the input variables (x) and output variable (y). If there is a single input variable then the model is known as simple linear regression, conversely a model dealing with multiple input variables is known as multiple linear regression.

The ordinary least squares algorithm is frequently used to estimate the linear regression model parameters from a dataset. Linear regression is frequently used in ML due to its simple representation and explain-ability.

The mathematical representation is a linear equation that combines:

- Input values (x)
- The output (y)

The general form of a linear regression model is:

$$Y_i = f(x_i \beta) + e_i \tag{2.3}$$

Where the linear equation assigns a scale factor to each of the input values (x), known as the coefficient it represented as β. The bias coefficient is represented as e_i which gives the line an additional degree of freedom.

In high dimensions [2] the linear regression model finds a hyperplane in high dimensional space. Regression models are sometimes described by defining their complexity (typically the number of coefficients or degrees of freedom in the model). More coefficients mean the model is more flexible, but more complex. If a coefficient is equal to 0 then that input variable has no effect/influence on the model. Regularisation methods are used in linear regression to reduce the complexity.

2.7.3 Logistic Regression

Investigating the relationship between the target and independent predictor variables using regression analysis is a version of a predictive modelling techniques. It is frequently used to look at:

1. forecasting
2. finding causal effect relationships
3. time series modelling

Logistic Regression is a supervised ML algorithm used for classification tasks. It is a type of regression analysis that is used to predict the probability of a binary outcome, such as whether an email is spam or not spam. In Logistic Regression, the input data is first transformed into a set of features, which are then used to make predictions about the probability of the binary outcome. The algorithm works by finding the best coefficients for the features that will maximize the probability of correctly predicting the outcome. The

[2] working with more than one input value (x)

prediction made by logistic regression is based on the logistic function, which maps the predicted probability to a value between 0 and 1. The logistic function is defined as:

$$f(x) = 1/(1 + e^{-x}) \tag{2.4}$$

where x is the predicted value based on the features and coefficients. To train a Logistic Regression model, the algorithm iteratively adjusts the coefficients to minimize the error between the predicted probabilities and the true outcomes. This is typically done using an optimization algorithm, such as gradient descent. Logistic Regression is a popular algorithm because it is relatively simple to implement and interpret, and it is widely used for binary classification tasks. However, it is not suitable for tasks with more than two classes, or for tasks where the relationship between the features and the outcome is more complex than a linear relationship.

There are many different types of Logistic Regression. These are defined as:

- **Binary Logistic Regression:** The categorical response has only two different values. For example detecting whether a patient has diabetes or not.
- **Multi-nominal Logistic Regression:** There are more than two different categorical variables such as detecting the weather based on meteorological and atmospheric input variables to classify if its going to rain, snow or have sun.
- **Ordinal Logistic Regression:** Ordinal Logistic Regression is used to

predict multiple categories where the categorical output will have some order. An example of this could be in predicting the degree classification of a student, where the possible outputs have a natural order.

2.7.4 Decision Trees & Regression Trees

A decision tree is a directed acyclic graph that can be used to make decisions. Decision trees can be used to build both classification and regression models. A key use is in data mining to discover existing patterns of information that are present within the dataset. Decision trees work by reducing the size of the dataset by breaking it down in to smaller subsets by producing a set of "rules".

Unlike other learning algorithms decisions trees can accept both categorical and numerical variables and do not require data normalisation which means in some cases, the time needed to set up the model is reduced. A completed decision tree comprises decision nodes and leaf nodes. Leaf nodes are representative of the final output and should be equal to the ground truth variable. A decision node must have two or more branches and represents a "rule" applied to the input variable. The top node of the decision tree is known as the root node, and is the strongest predictor. Figure 2.5 shows the example of a decision tree about prescribing antibiotics to children (Martignon, 2010).

ID3

The most commonly used algorithm for building decision trees is called ID3. It was developed by Quinlan (1986). The algorithm uses a top-down greedy

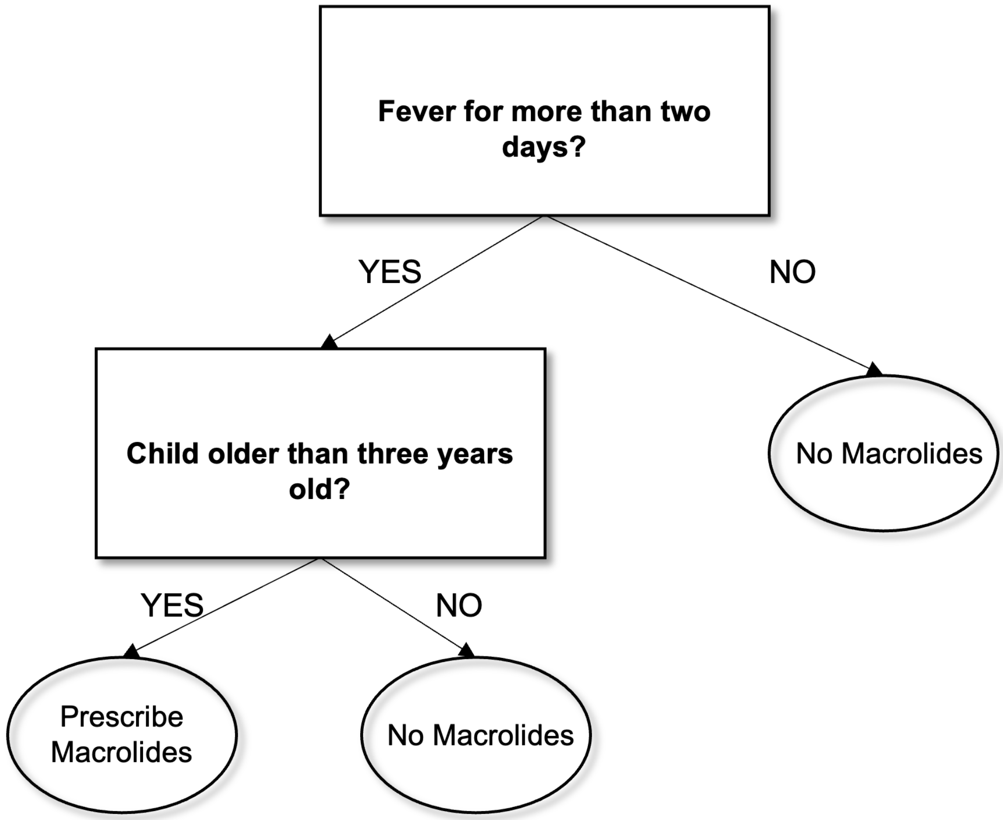

Fever for more than two days?
YES
NO
Child older than three years old?
No Macrolides
YES
NO
Prescribe Macrolides
No Macrolides

Figure 2.5: Decision tree to detail whether antibiotics should be prescribed or not (Martignon, 2010)

search to search through all of the space and consider all of the possible branches with no backtracking. ID3 was designed to build decision trees when there are multiple features; however, it is generally found to construct simple trees and there is no guarantee that better trees have not been overlooked. ID3 uses information gain and entropy to build each node of the decision tree.

Entropy

The ID3 algorithm uses entropy to compute how similar the sample is and, at each node, partitions the data into subsets where the resulting entropy of those subsets is minimised. This is commonly known as calculating the homogeneity of a dataset. If the sample is all the same, then the entropy value is 0, conversely, if a sample is equally divided then the entropy is 1. Entropy is calculated using the following formula 2.5, below.

$$H(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{2.5}$$

The steps of the ID3 algorithm are:

1. Calculate the entropy for each attribute of the dataset.
2. Split the dataset for each different attribute.
3. Calculate the entropy for each different potential branch.
4. Add all the parts to get the total entropy for the split.
5. The results from step 4 are subtracted from the entropy before the split (this is the information gain).
6. The attribute with the largest information gain is a decision node. Divide the dataset at this point and repeat the steps 1-5.
7. Branches that have an entropy of 0 are leaf nodes, anything with a value greater than 0 requires further splitting.

These steps are repeated until all of the data is classified.

Pruning

Unfortunately the ID3 algorithm when used on larger datasets with multiple targets can result in overfitting of the dataset. There are three methods used to reduce overfitting for decision trees and to improve its ability of predicting outputs: top-down pruning, bottom-up pruning, and error driven pruning.

2.7.5 Random Forests

Despite the pruning techniques described previously, decision trees do not perform well with noisy data. A common effect of this is that multiple runs of training a decision tree model can produce different trees and negatively effect the overall accuracy of the model. Breiman proposed a method to overcome these limitations called random forests (Breiman, 2001). The final work by Breiman combined their earlier approaches published between 1995 and 1998. Random forests build on the decision tree learning algorithm to improve how they deal with noisy data. Random forests aggregate and weight the results of multiple smaller decision trees (created from subsets of the dataset as well as subsets of the features). This reduces the impact bad data can have and is known as a form of ensemble learning.

Another benefit of using random forests over decision trees is that they can dramatically reduce the computation time needed to train the model. As datasets grow and evolve, models need retraining or redeveloping.

A key advantage of decision trees is that they are easy to interpret. It is possible to take a single sample and trace it through the tree to make a decision. Unfortunately, the ensemble learning within random forests means it is not possible to ascertain why the model has produced good results.

2.7.6 Support Vector Machines

Support Vector Machines (SVMs) are an alternative ML method that can be used for both classification and regression problems. The key ideas were written by Vapnik, Chevonenkis and co-workers where they described their method for creating a maximum-margin hyperplane to separate data classes. However, the research went largely unnoticed until 1992 when Boser, Guyon and Vapnik described them in COLT-92 (Boser et al., 1992). Original SVMs weren't particularly useful until the 1992 creation of the the kernel trick which dealt with linearly separated data. SVMs are aptly described as:

> *Support Vector machines can be defined as systems which use hypothesis space of a linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory (Huang et al., 2017).*

The main objective of a Support Vector Machine (SVM) is to find a hyperplane in N-dimensional space (where N is the number of features) that distinctly classifies all of the points. SVMs aim to find the hyperplane with the maximum distance between points of different classes. In the process of maximising the margin distance, the model is provided with reinforcement; therefore additional observations can be correctly classified with a higher degree of confidence.

These hyperplanes are the decision boundaries that are used to classify observations. Depending on which side of the hyperplane the observation sits, will determine which class it falls in to. When the number of input features is two, the hyperplane can be considered as a line where data points can fall above or below the intersecting line. If there are three input features, then the hyperplane is a two dimensional plane.

The position, orientation, and angle of the hyperplanes is determined by the support vectors. These are single observations that sit close to the hyperplane on the edge of the decision boundary. The more support vectors, the easier it is to maximise the margin of the classifier. Finding the maximum margin hyperplane enables SVMs to overcome the problems caused by overfitting and underfitting. To maximise the margin between the observations and the hyperplane a loss function known as Hinge Loss is often used. As well as the loss function, a regularisation function is also used. When the SVM mis-classifies a data point then the loss value and regularisation cost are used to update the hyperplane gradient (Cristianini and Ricci, 2008).

One key aspect that makes SVMs so popular and powerful is the kernal trick which makes them applicable to nonlinear classification tasks. The kernel trick works by transforming the input data into a higher-dimensional space using a kernel function, so that the classes can be separated by a hyperplane in that space. This allows the SVMs to perform nonlinear classification without actually having to compute the transformation of the data into the higher-dimensional space.

The kernel function used in the kernel trick can be any function that satisfies the Mercer condition, which basically states that the function should be continuous and positive-definite. Some common kernel functions used in SVMs include the linear kernel, the polynomial kernel, and the radial basis function Radial Basis Function (RBF) kernel.

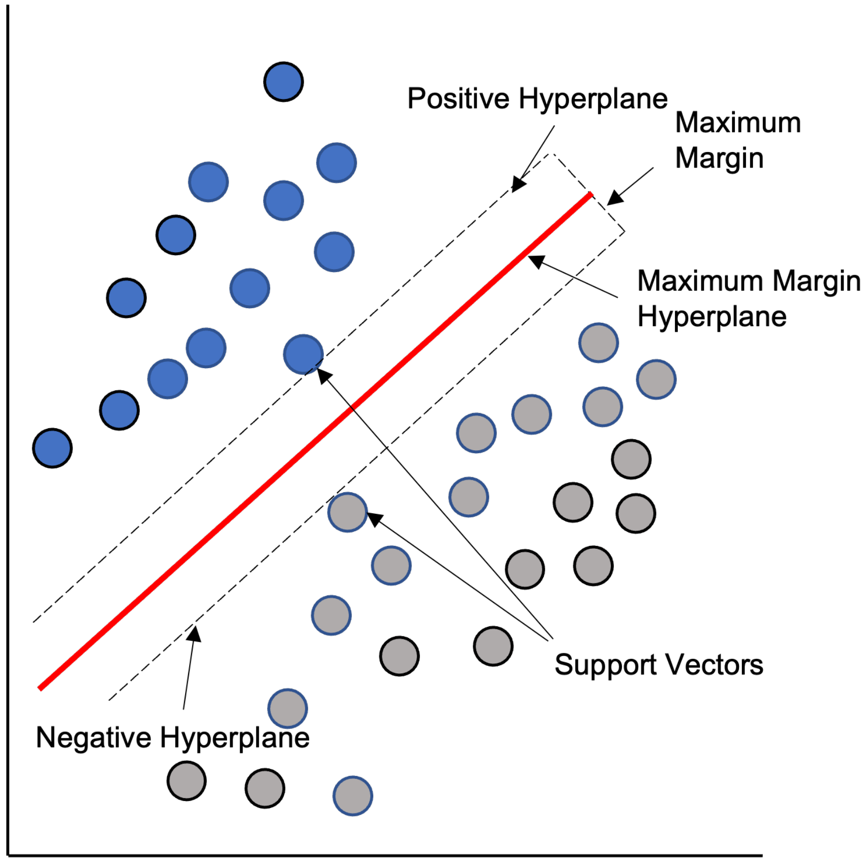


Figure 2.6: Support Vector Machine Graphical Explanation

The kernel trick is a useful technique because it allows SVMs to perform nonlinear classification without requiring the user to manually specify the transformation of the data into the higher-dimensional space. This can be particularly useful when the data is complex or when it is not clear what kind of transformation would be most effective.

2.7.7 Artificial Neural Networks

An Artificial Neural Network (ANN) is a type of ML model inspired by the structure and function of the biological brain described in 2.7.7. It consists of layers of interconnected “neurons”, which process and transmit information. Each neuron receives input from other neurons, and uses this input to compute and output a signal to other neurons in the next layer. The structure and function of Artificial Neural Networks (ANNs) allows them to learn and adapt to new data and tasks, without the need for explicit programming. This is done through a process of training, in which the neural network is presented with a large dataset and adjusts the strengths of the connections between neurons (called weights) to improve its performance on the task. ANNs have been successful in a variety of applications, including image and speech recognition, language translation, and even playing games. They are a powerful tool for solving problems that involve complex patterns and relationships in data.

A neural network is a widely used classification technique that makes use of some of concepts of the way the human brain learns (Gurney, 1997).

In 1943, Warren McCulloch and Walter Pitts published a paper entitled “A logical calculus of the ideas immanent in the nervous activity” (McCulloch and Pitts, 1943). The paper described how the authors used existing knowledge of brain cells and how they are tied together in order to learn complex patterns. These brain cells are commonly known as neurons. This collective research allowed the authors to introduce the McCulloch-Pitts Neuron (MCP) model which is the basis for all common neural networks used today. The MCP model makes use of features from biological neurons to make up each node in the network. The first MCP neuron developed had its limitations, and it wasn't until Frank Rosenblatt introduced the perceptron in the 1960s (Rosenblatt, 1960) that significant developments were made in the field of ANNs. In the perceptron, the Neural-Network (NN) neuron is passed through a "pre-processer" that will contain units that are associated with it. The perceptron will check if there is a specific feature in the data that can be used to help predict the output (Rosenblatt, 1960). ANNs are capable of finding patterns in data that are usually too complex for human beings to identify. An ANN becomes proficient at solving a specific problem because of the information it is trained on.

The Human Brain

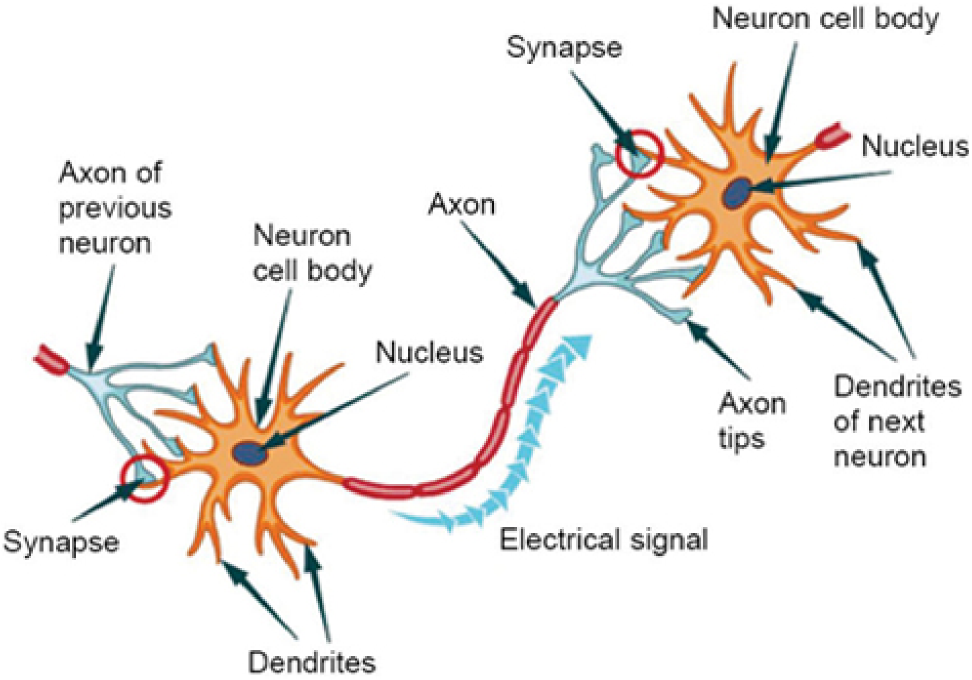


Figure 2.7: Schematic of a neuron within the human brain (Awan-Ur-Rahman, 2019).

The human brain is made up of cells called neurons. It is estimated that there are 100 billion neurons in the human brain with 10e^{15} connections with each other. A biological neuron consists of four parts:

1. Soma

2. Axon

3. Dendrite

4. Synapse

The Dendrite receives varying electro-chemical signals from other neurons into the cell body. The Soma sometimes known as the cell body, contains a nucleus and other chemical structures that are required to support the cell and lastly, performs the data processing. This is effectively done by triggering an output when the strength of the input signal exceeds a certain threshold. The Axon carries the output signal from the neuron to other neurons. The Synapse is the point of connection between the dendrites of two neurons.

Artificial Neurons

In its simplest form a biological neuron takes some inputs, carries out some calculations and produces an output - an ANN works in the same way Figure 2.8 illustrates what a 2-input neuron would look like.

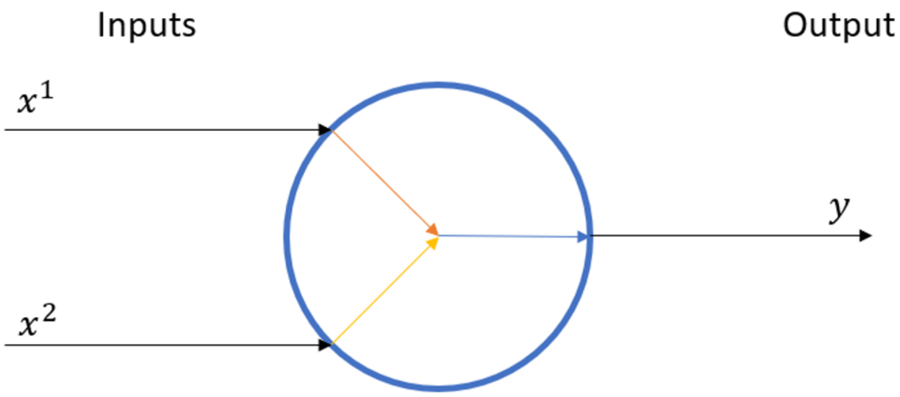


Figure 2.8: Two Input Neuron Diagram.

There are several things happening in figure 2.8.

1. Each input (x) is firstly multiplied by a weight (w):

$$x_1 \;\rightarrow\; x_1 * \; w_1 \tag{2.6}$$

$$x_2 \;\rightarrow\; x_2 * \; w_2 \tag{2.7}$$

2. The weighted inputs are added together with a bias (which is denoted by b).

$$(x_1 * \ w_1) + \ (x_2 * \ w_2) + \ b \tag{2.8}$$

3. The resulting calculation from equation 2.8 is passed through a defined activation function (f).

$$y = \ f\left((x_1 * \ w_1) + \ (x_2 * \ w_2) + \ b\right) \tag{2.9}$$

   This activation function is used to turn an input that is unbounded into an output that has a predictable form. Sigmoid, the hyperbolic tangent function, and the Rectified Linear Unit (ReLU) function are commonly used as activation functions in ANNs. A sigmoid function can be thought of as compressing data in the range $(-\infty, \ +\infty)$ into (0,1).

Feed Forward Networks

A feedforward neural network is a type of ANN in which the connections between the neurons do not form a cycle. This means that information flows through the network in only one direction, from the input layer to the output layer, without looping back.

In a feedforward neural network, the input data is passed through the input layer, which then passes it on to one or more hidden layers. Each hidden layer processes the data and passes it on to the next layer until it reaches the output layer. The output layer produces the final result or prediction based on the input data.

Feedforward neural networks are typically used for supervised learning tasks, such as classification and regression. They are called feedforward because the data flows through the network in a single direction, from the input layer to the output layer, without looping back.

The structure of a feedforward neural network can be represented as a series of interconnected layers, where each layer consists of a set of artificial neurons or nodes. The input layer receives the raw input data, and the output layer produces the final prediction. The hidden layers process the data and pass it on to the next layer. The number of hidden layers and the number of neurons in each layer can be adjusted to optimize the model's performance on a given task.

2.8 Unsupervised Learning

In contrast to supervised learning, unsupervised learning aims to train a system to represent particular input patterns in a way that does not require the ground truth output. An example of this is the way humans learn to recognise objects. For example, when a baby is introduced to the family dog. If weeks later, a family friend brings another dog round, the baby is able to recognise it as a dog even though she has not seen this particular dog previously. The features of the dog are recognised by the baby: 2 eyes, walking on 4 legs, tail, and a collar. The baby has created a mental model of a dog without necessarily having a mental label attached to it. This is an example of unsupervised learning where you can learn from the structure of the data provided.

> *"We expect unsupervised learning to become far more important in the longer term. Haman and animal learning is largely unsupervised: We discover the structure of the world by observing it, not by being told the name of every object" (LeCun et al., 2015).*

There are three main reasons to use unsupervised learning over supervised learning:

1. Unsupervised ML is able to find unknown patterns within data.

2. Features can be found in unsupervised learning that can be made use of for categorization.

3. It is quicker to produce datasets with unlabelled data, as it doesn't

require any manual intervention (unlike labelled datasets that are expensive to produce).

Clustering is the most important unsupervised problem and involves finding structure and patterns in collections of uncategorised data. Businesses that need to understand customer behaviour and purchase history may use clustering techniques to focus advertising - particularly on social media. Customers can be clustered on factors such as age, gender, purchase process, and payment type.

Three different versions of clustering that can be implemented are:

1. **K-means clustering:** where the data points are partitioned into K clusters based on minimising the variance between each cluster.

2. **Hierarchical clustering:** where data points are clustered into both parent and child clusters. Customers may be initially split by age and then further split by other identifying traits.

3. **Probabilistic clustering:** where a probabilistic scale is used to cluster the data points into different clusters. An example might be in developing categories for sport equipment: "Football boot", "Rugby boots", "Football ball" and "Rugby ball" can be clustered using two different properties they relate to e.g.: the sport "Rugby" and "Football" or the equipment type e.g. "boot" and "ball".

Disadvantages Of Unsupervised Learning

1. Results may not be as reliable as they are automatically generated.

2. The user needs to spend time interpreting the final outcome clusters

2.8.1 Clustering

Clustering is a type of unsupervised learning. Clustering is used to find meaningful structure and groupings that are inherent within the data without using labelled datasets. It involves dividing the population of data points into a number of different groups such that data points within the same group are similar to each other in some way. Figure 2.9 displays the data points for a given data set. From visual inspection it is obvious that three clusters can be formed from the data, as shown in Figure 2.10.

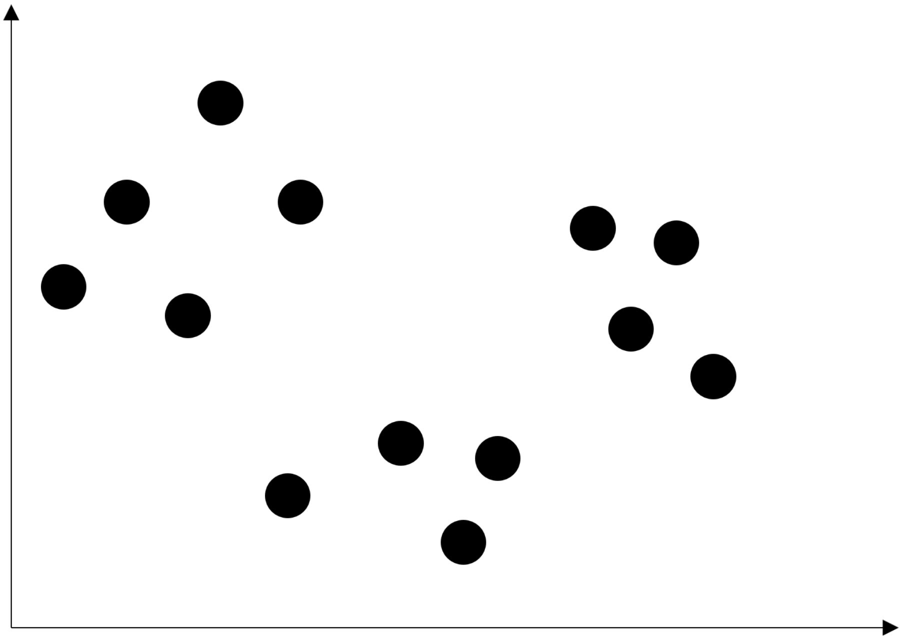


Figure 2.9: Randomly generated dataset which could be split in to 3 clusters

Clustering is widely used in various industries to segment data and to understand intrinsic groupings. The interpretation of whether they are good clusters or not depends on the use case. However, common evaluation metrics include Silhouette Scores (which measures the separation of clusters), and

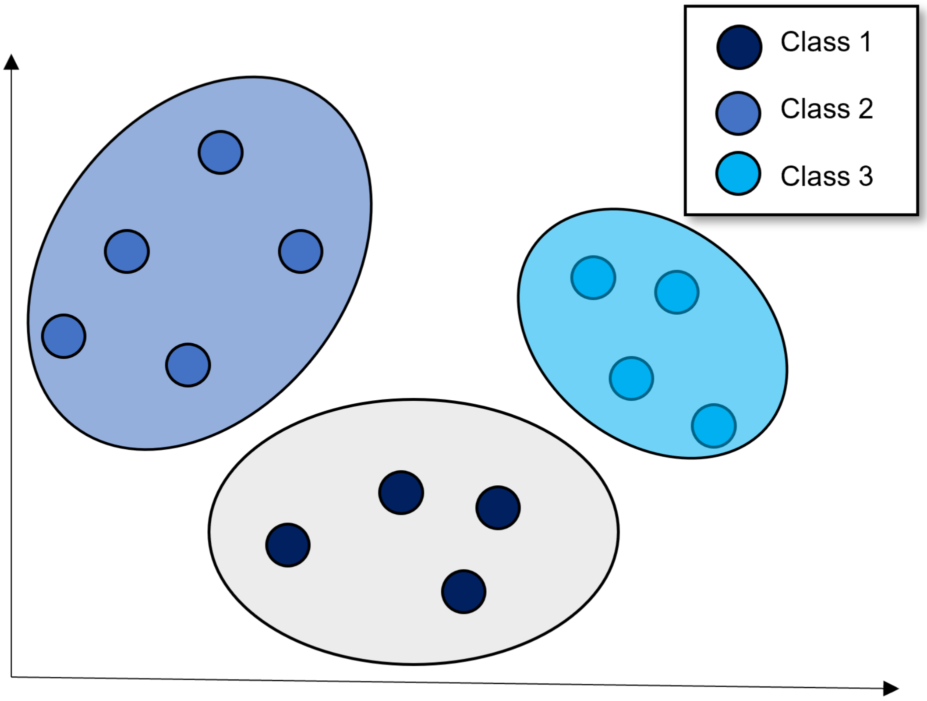


Figure 2.10: Clustering Example with cluster generated

the Calinski-Harabasz index (which is the ration between cluster distribution) as well as the Davies-Bouldin Index (which measures how well spread and dense the clusters are) (Wang and Xu, 2019; Xiao et al., 2017).

Clustering can be further split in to four main types of segmentation:

1. Centroid Models
2. Distribution Models
3. Density Models
4. Connectivity Models

Centroid models are iterative clustering algorithms which measures similarity by the closeness of a data point to the randomly placed centroid. K-Means clustering is commonly used in clustering and is a centroid model. When using these models, the number of required clusters needs to be defined before training the model. Therefore, it is important that some prior information is known about the dataset and what outcomes are to be expected. K-Means clustering is the most commonly used clustering algorithm. It is an iterative approach that aims to find the local maxima at each iteration. The K-Means algorithm is usually attributed to Stuart Lloyd who introduced the algorithm when working in the Bell Labs in 1957 (Lloyd, 1982). It was developed as a technique for pulse code modulation. The K-Means algorithm comprises of five steps:

1. Specify the number of clusters that the data observations will be segmented in to.
2. Randomly assign each observation to a cluster.
3. Randomly assign the starting centroids of the data. The number of centroids should match the number of clusters determined in step 1.
4. Re-assign each point to the closest centroid and re-compute the positions of the centroids, moving them around the data space as required.
5. Repeat step 4 until no more improvements are possible.

Distribution models try and understand how probable it is that a data point belongs to the same distribution as other data points (Normal, Gaussian). As a consequence to this, they are often prone to overfitting. Expectation-Maximization is an example of distribution models and it uses multivariate normal distributions to determine which cluster data points should fall into.

No prior knowledge or understanding of the dataset is required, which makes them a good technique to use in data exploration.

Density Models look at how dense a cluster of points is in the data space. If some points are grouped densely together, it is said they belong to the same cluster if they are also in the same region. DBSCAN and OPTICS are commonly used density models.

Connectivity models are based on the idea that data points that are closer in the data space will be similar to each other. Connectivity models use hierarchical approaches to group the data together. Hierarchical Clustering Analysis (HCA) is an unsupervised clustering algorithm which generates clusters that have prevalent ordering from the top to the bottom. Agglomerative Hierarchical Clustering (AHC) is one of the most commonly used models in HCA. It is a known as a "bottom up" method, as each observation starts as its own cluster and pairs of clusters are iteratively merged together. The distance between each cluster is measured using different linkage methods. These methods are:

- **Complete-Linkage:** Complete-Linkage describe when the distance between two clusters is defined as the longest distance.
- **Single-Linkage:**Single-Linkage is used when the distance between the two clusters is the shortest distance this is known as Single-Linkage. This linkage method is susceptible to outliers in the data.
- **Average-Linkage:** Average-Linkage is when the distances between each pair of observations in each cluster is summed up and divided by the total number of pairs This provides the average inter-cluster

distance.

- **Centroid-Linkage:** Centroid-Linkage is the distance between the centroids in two different clusters.

Different linkage methods are applicable for different data sets. There is no exact use case for each method as they all produce different clusters.

Unlike AHC, Divisive Hierarchical Clustering (DHC) is a "top down" approach. All of the data points are initially assigned to a single cluster. The observations are then partitioned to the two "least similar" clusters. This recursive partitioning continues iteratively until there are no more splits that can be done. For both DHC and AHC the user needs to specify the number of clusters required to understand when the termination should take place.

2.8.2 Dimensionality Reduction

With the computerisation of Electronic Health Records (EHRs), datasets within the medical domain are increasingly getting bigger. To make these datasets easier to interpret, dimensionality techniques are used to reduce the number of features whilst ensuring that most of the information within the dataset is preserved (Jolliffe and Cadima, 2016). Furthermore, explaining the final classification of a ML problem can be difficult when there are many features present. As the feature set grows beyond 2 and 3 dimensions, it is not easy to visualise the dataset. Many features are correlated[3] and hence redundant. For this scenario, where there are many correlated features, dimensionality reduction is utilised. Dimensionality reduction reduces the number of

[3] Correlation explains how one or more variables are related to each other

features under consideration by producing a set of principle variables. The set of principal variables is obtained by two different methods: feature selection and feature extraction (Uberoi, 2017).

1. **Feature Selection**: A subset of the original features is found. Features are selected using three different techniques:
    - Filter
    - Wrapper
    - Embedding
2. **Feature Extraction**: The data in a high-dimensional space is reduced to a lower-dimensional space.

Zhu et al. (2015) developed a novel dimensionality reduction technique called Niche Genetic Algorithm (NGA); their methodology enabled them to reduce the number of features in their sepsis dataset from 77 to 10 (many of the features overlapped and were heavily correlated). The resulting model produced an accuracy of 92% when predicting 28-day death in sepsis patients. To better understand the concepts of dimensionality reduction, consider a 3D classification. This can be difficult to visualise; however, a 2D problem can be mapped to a 2-dimensional space. A 1D problem can be mapped to a single line.

There are various methods that can be employed for dimensionality reduction. They include:

- **Principal Component Analysis (PCA)** is a dimensionality reduction technique that is commonly used in ML. It is a linear transformation method that reduces the number of dimensions in a dataset by projecting the data onto a lower-dimensional subspace.

PCA is based on the idea that the directions with the highest variance in the data are the most informative, and that the data can be projected onto a lower-dimensional subspace while preserving as much of the original variance as possible.

To perform PCA, the data is first centered by subtracting the mean from each feature. The covariance matrix of the centered data is then computed, and the eigenvectors of the covariance matrix are found. The eigenvectors are ranked by the corresponding eigenvalues, which indicate the amount of variance in the data explained by each eigenvector. The eigenvectors with the highest eigenvalues are selected as the principal components of the data.

The data is then projected onto the subspace defined by the principal components, resulting in a lower-dimensional representation of the data. The number of dimensions in the final representation can be controlled by selecting the number of principal components to keep.

- **Linear Discriminant Analysis (LDA)** works by projecting the data onto a lower-dimensional space that maximizes the separation between the different classes. It does this by finding a projection that maximizes the ratio of the between-class variance to the within-class variance. In other words, LDA tries to find a projection that maximizes the difference between the means of the different classes while minimizing the variance within each class.

To perform LDA, the mean vector and covariance matrix for each class are calculated. The mean vector for each class represents the center of mass of the data points belonging to that class, and the covariance matrix represents the spread of the data around the mean.

The projection is then found by solving a set of linear equations that maximize the ratio of the between-class variance to the within-class variance. The resulting projection is used to transform the data onto the lower-dimensional space.

2.9 Semi-Supervised Learning

Semi-supervised learning is a combination of supervised and unsupervised learning approaches. It is used for similar applications to that of supervised learning, and makes use of a combination of both labelled and unlabelled data for training. There is usually a small amount of labelled data and a greater amount of unlabelled data. Labelled data is usually much more expensive to generate as labelling the data is often labour intensive. Semi-supervised learning is used when a large amount of data is required, but the cost associated with labelling is too high for a fully labelled training process. Face identification, such as that implemented by Facebook and Google, is an example of semi-supervised learning (Liu et al., 2021).

A classic example of semi-supervised learning models is speech analysis. Applying semi-supervised learning techniques can reduce and minimise the effort required by human resources to greatly improve speech analytic models. Web content classification is another example of where semi-supervised learning is utilised. Similarly to the previous use case, human intervention is typically required to classify the content. Semi supervised learning techniques can be implemented to speed up this content classification process. However the key drawback is that it isn't currently possible to verify that the labels produced are accurate, therefore the resulting outcomes are not as trustworthy as fully supervised techniques.

Gu et al. (2020) used semi-supervised learning with a graph embedded Random Forest. A major challenge in the analysis of medical imaging is the lack of images with labels or annotations present. As previously discussed the process of labelling and annotating records can be a very costly process and in the medical domain it requires a level of expertise to correctly identify records. The results presented by Gu et al. (2020) demonstrated that using the information gain calculation in Random Forests reduced the accuracy of the results, however, utilising a graph-embedded entropy, they were able to produce results that were significantly improved whilst also maintaining the low computational burden and robustness to over-fitting that is a key deciding factor in Random Forests.

Table 2.2: Summary Of Machine Learning Techniques

Parameter	**Supervised Learning**	**Unsupervised Learning**	**Semi-Supervised Learning**
Definition	Supervised learning is a ML technique in which a model is trained on labelled training data to make predictions or decisions. The training data consists of a set of input examples and corresponding correct output labels, which are used to train the model to associate inputs with their correct outputs.	Unsupervised learning is a ML technique in which a model is trained on an unlabelled dataset to discover patterns or relationships in the data. Unlike supervised learning, the model is not given correct output labels for the training data. Instead, it must discover the inherent structure of the data	Semi-supervised learning is a ML technique that combines a small amount of labelled data with many unlabelled data points to train a model.
Input Data	Labelled	Unlabelled	Mixture of Labelled Unlabelled
When to use	You know what result you are looking for	When you don't know what you are looking for in the data	Used when it is expensive or time-consuming to label a large amount of data, or when there is a large amount of naturally occurring unlabelled data available.
Applicable In	Classification & Regression	Association & Clustering	Classification & Regression
Algorithms	Random Forest, Decision Trees, SVMs	K-Means, Gaussian Mixture Models	GANs, CoTraining, Multi-View Learning

2.10 Comparison Of Algorithms

Linear Regression

Pro's

- Simple to implement and efficient to train.
- Overfitting can be reduced by regularization.
- Performs well when the dataset is linearly separable.

Con's

- Assumes that the data is independent which is rare in real life.
- Prone to noise and overfitting.
- Sensitive to outliers

Logistic Regression

Pro's

- Less prone to over-fitting but it can overfit in high dimensional datasets.
- Efficient when the dataset has features that are linearly separable.
- Easy to implement and efficient to train.

Con's

- Should not be used when the number of observations are lesser than the number of features.
- Assumption of linearity which is rare in practise.

- Can only be used to predict discrete functions.

Decision Tree

Pro's

- Can solve non-linear problems.
- Can work on high-dimensional data with excellent accuracy.
- Easy to visualize and explain.

Con's

- Overfitting. Might be resolved by Random Forest.
- A small change in the data can lead to a large change in the structure of the optimal decision tree.
- Calculations can get very complex.

K Nearest Neighbour

Pro's

- Can make predictions without training.
- Time complexity is O(n).
- Can be used for both classification and regression.

Con's

- Does not work well with large dataset.
- Sensitive to noisy data, missing values and outliers.

- Need feature scaling.
- Choose the correct K value.

K Means Clustering

Pro's

- Simple to implement.
- Scales to large data sets.
- Guarantees convergence.
- Easily adapts to new examples.
- Generalizes to clusters of different shapes and sizes.

Con's

- Sensitive to the outliers.
- Choosing the K values manually is tough.
- Dependent on initial values.
- Scalability decreases when dimension increases.

Support Vector Machine

Pro's

- Good at high dimensional data.
- Can work on small dataset.
- Can solve non-linear problems.

Con's

- Inefficient on large data.
- Requires picking the right kernal.

Principal Component Analysis

Pro's

- Reduce correlated features.
- Improve performance.
- Reduce overfitting.

Con's

- Principal components are less interpretative.
- Information loss.
- Must standardize data before implementing PCA.

Naive Bayes

Pro's

- Training period is less.
- Better suited for categorical inputs.
- Easy to implement.

Con's

- Assumes that all features are independent which is rarely happening in real life.
- Zero Frequency.
- Estimations can be wrong in some cases.

Artificial Neural Network

Pro's

- Have fault tolerance.
- Have the ability to learn and model non-linear and complex relationships.
- Can generalize on unseen data.

Con's

- Long training time.
- Non-guaranteed convergence.
- Black box. Hard to explain solution.
- Hardware dependence.
- Requires user's ability to translate the problem.

Adaboost

Pro's

- Relatively robust to overfitting.
- High accuracy.

- Easy to understand and to visualize.

Con's

- Sensitive to noise data.
- Affected by outliers.
- Not optimized for speed.

2.11 Summary

Spotting patterns in data can often be improved and sped up by using AI to spot patterns in data quickly. Many real-world applications currently use AI techniques to assist in making decisions and offer support for different types of datasets. This Chapter has shown that the subset of AI, ML can provide robust solutions and provide an accurate prediction that could be applied to many medical tasks. The main drawback now is the complexity of developing such mathematical models and how to deal with missing data and unbalanced datasets effectively.

This Chapter has aimed to provide a detailed introduction to the history of AI and the different types of ML and introduce some core ML algorithms. In section 2.2 an overview of AI and some of the most note-able historical points to date were presented. Section 2.5 described the different taxonomy of ML models before supervised learning, and unsupervised learning is defined and described.

The different types of learning algorithms and the decision considerations of which algorithm to use are outlined in section 2.6. Figure 2.3 is a flowchart of the decision-making process.

Supervised learning is discussed in section 2.7 and the algorithms Linear Regression, Logistic Regression, Decision Trees, Random Forests, SVMs, and ANNs are reported between subsections 2.7.2 to 2.7.7.

Section 2.8 introduces the concept of unsupervised learning and semi-supervised algorithms. Clustering (in subsection 2.8.1) and Dimensionality reduction (in subsection(2.8.2), which are statistical techniques are also defined.

Section 2.10 outlines the pros and cons for different ML algorithms. The concepts, algorithms and decision-making processes are utilised throughout this work. Chapter 3 describes some areas of medicine that utilise AI and the algorithms that they have used. Decision Trees, Linear Regression, Neural Networks, Random Forests are used in Chapter 4. The Logistic Regression algorithm is selected for use in Chapter 5.

Chapter 3

Machine Learning In Medicine

Image Technology News estimates that the market for AI in healthcare will grow to more than \$31.3 billion by 2025. This is a growth of more than 40% since 2018 when the market was valued at \$22.4B (Inc, 2019). This chapter discusses some of the key areas of medical decision support that can be helped by ML techniques, whilst also providing some examples that are commonly described in literature. (Rajkomar et al., 2019; Inc, 2019)

3.1 Recordkeeping

As more health records in different countries are being moved to digital systems, health informatics are being used to streamline recordkeeping, improve patient care, reduce the need for large administrative costs, and ensuring that patients are not administered incorrect medicines due to allergies.

This shift to EHRs allows data about patients to be easily transferred between clinicians; however, the unstructured nature of these records makes them particularly difficult to process automatically. NLP - the application of computational techniques to analyse natural language or speech - is a crucial tool for making use of such data by extracting key information from EHRs.

3.2 Data Integrity

ML algorithms can only effectively use EHRs where the data is complete and contains minimal missing data. Gaps in healthcare information can result in ML algorithms providing inaccurate predictions, which can hamper decision making. It is important for healthcare professionals to maintain the integrity of records and make sure that they are as complete as possible. However, it is sometimes not possible to obtain all of the data due to time constraints, machine failure, or lack of funding to complete tests. There are several considerations that medical practitioners use to maintain the integrity of patient data:

- **Understand the process workflow and data life-cycle:** The flow of data should be well documented, continually reviewed and maintained. The mapping of data can help to consolidate workflows by highlighting potential risks and areas of improvements to the existing workflow.

- **Automate data workflows:** Manual data entry or transcription can lead to poor integrity behaviour. Furthermore, different data entry methods can lead to results that differ even though they are the same. Automating data workflows can reduce the need for clinicians making decisions when capturing data. Automated solutions that currently exist are Electronics Lab Notebook (ELN) and Lab Information Management Systems (LIMS), these systems can be put in place to capture data efficiently in real-time and add the metadata to patient data.

- **Review data for quality and completeness:** Critical data should be reviewed by experts with a knowledge of the subject area. The Medicine and Healthcare Product Regulatory Agency(s) (MHRA)'s provide further guidance for data integrity and how it should be reviewed.

3.3 Predictive Analysis

Combining ML, predictive analysis, EHRs, and health informatics enables the development of supporting tools that can be used to improve the healthcare processes and leads to more accurate diagnosis and more effective treatments. Furthermore, and most importantly, it can improve patient outcomes by suggesting alternatives to surgery or suggesting medicines that may not have been considered. ML can also provide information on areas that require closer inspection and can be used to target specific research areas that may be behind or where there is currently a gap in the area.

Jamin et al. (2021) demonstrated that utilising different ML algorithms, including SVMs and ANNs, can be trained on medical data to provide better results than medical support tools currently being used within industry.

Although predictive analysis demonstrates potential in medicine, protecting a patient's safety is essential. Regulatory and professional bodies ensure that advanced algorithms are scrutinised and adhere to strict standards of clinical benefit (Yu and Kohane, 2019). These standards are also applied to clinical therapeutics and predictive bio-markers. Independent and external validation and prospective testing of newly developed algorithms are clearly needed, although certain regulatory bodies have expressed concern about the standard of these validations. Parikh et al. (2019) have proposed five standards and guidelines to help regulate predictive analysis, which can be used to validate algorithms before implementation within the clinical domain. The Transparent Reporting of a multi-variable prediction model for Individual Prognosis or Diagnosis (TRIPOD) checklist is an example of an existing standard used to validate multi-variable prediction models within medicine (Collins et al., 2015).

3.4 Applications of ML in Healthcare

ML is being considered for many different projects across the world as a key tool for solving a range of issues (Obermeyer and Emanuel, 2016). Some areas in medicine, where there has been significant research undertaken in ML and where ML techniques are currently being used, are described in the following sections.

3.4.1 Disease Identification & Diagnosis

The detection of patterns in data is a core function of ML. In medicine this can be used to detect patterns in diseases and specific health conditions by training models on electronic health records and additional patient data. As well as improving patient outcomes, many ML models in medicine can provide additional benefits such as:

- Shorter time to train than a human
- Improved diagnosis
- Improved consistency

Alzheimer's Disease (AD) is the leading cause of dementia in Western countries (Terry, 1994). Alzheimer's is usually characterised by the loss of memory and the impairment of at least one cognitive function. There is no specific test to determine if someone has AD, with a definitive diagnosis only available on autopsy or biopsy (Jameson et al., 2020).

The current test for AD diagnosis is based upon:

- Clinical history
- Neuropsychological
- Laboratory tests
- Neuroimaging
- Electroencephalography (EEG)

To enable more accurate diagnosis faster Trambaiolli et al. (2011) described an effective way of improving the diagnosis of Alzheimer's. Their results can produce more accurate diagnoses and follow-treatment results. Their study utilised SVMs to search for patterns in EEG windows to spot the difference between those patients that are within the control group or are showing signs of having AD. Their experiments result in a quantitative Electroencephalography (qEEG) processing method that can automatically determine patients that have AD from normal individuals. The study was undertaken by looking at EEGs from 19 normal subjects. 14 were female and 5 were male with a mean age of 71.6 years. 16 AD patients showing mild to moderate symptoms were considered within the study (14 females/2 males). The analysis of EEG epochs found that the accuracy was 79.9% and a sensitivity value of 83.2%.

3.4.2 Medical Imaging Diagnosis

High resolution imaging technologies such as X-rays, Computerized Tomography (CAT) scan, and Magnetic Resonance Imaging (MRI) provide so much detail that it can be hard to spot cancerous cells by eye. Using these medical images, ML techniques have been able to look at the images on a pixel level to detect problems such as cardiovascular abnormalities and cancers. There is a lot of research in the area of medical imaging diagnosis.

One such example is Soenksen et al. (2021), who have trained an algorithm at the Massachusetts Institute of Technology that is more accurate at diagnosing skin cancer than *"board-certified dermatologists"*.

Global cases of melanoma skin cancer will reach nearly half a million (466,914) by 2040, an increase of 62% on 2018 figures (Team, 2022, 2020). Melanoma accounts for only about 1% of skin cancers but causes a large majority of skin cancer deaths. Melanoma are a type of malignant tumour that can be found on the skin. For years physicians and medical practitioners have utilised visual inspections to identify Suspicious pigmented lesions (SPL). These SPLs can be an early indicator of skin cancer. Typically, at this stage, a tumour biopsy is removed and tested for cancerous cells (further treatment is determined based on the results of these tests). Earlier identification of SPLs in primary care can dramatically reduce patient costs and improve the patient experience.

Soenksen et al. (2021) describe the utilisation of Deep Convolutional Neural Networks (DCNN) to classify and cluster images (these algorithms are within the subset of ML called deep learning). The DCNNs are used to develop an SPL analysis system. The system enables the identification of skin lesions that could be dangerous if missed by primary caregivers or left untreated by the patients themselves. 20,388 wide-field images from 133 patients located at Madrid's hospital Gregorio Marañón were publicly available and utilised by Soenksen et al. (2021). Each image was visually inspected by dermatologists who determined the legion, allowing the researchers to compare their results. The system demonstrated 90.3% sensitivity in distinguishing SPLs from nonsuspicious lesions, skin, and complex backgrounds whilst eradicating the need for cumbersome and time-consuming individual lesion imaging.

3.4.3 Robotic Surgery

Making use of data from previous surgeries that have been successful, ML-based robots can be trained to carry out complex surgeries. Human surgeons are susceptible to making mistakes - potentially causing a patient harm. Building on existing learning strategies for surgeons, ML-assisted robots have been developed to focus on two key areas: feature detection and computer-assisted intervention. These use cases are applied to both pre-operative planning (determining the most effective area to make an incision) and intra-operative guidance (utilising image systems to understand how the body will react to specific procedures). Training an ML-based surgical robot model to assist in performing these tasks can reduce human error and aid medical professionals during complex procedures. Furthermore, more operations can be completed via keyhole surgery (Zhou et al., 2020).

The advancements in these fields have led to an increase in Minimally Invasive Surgery (MIS), and the combination of computer-aided intra-operative guidance with the skills of surgeons has resulted in a reduction in surgical trauma. The four key areas where different AI techniques are applied to computer-aided intra-operative guidance are:

- **Shape instantiation:** This assists the surgeon to determine what shape and size an incision should be to be most effective for the operation.

- **Endoscope navigation:** During an intra-operative procedure directing an endoscope through the body can be a dangerous task, using computer vision, navigation is supported.

- **Tissue tracking:** Tracking biopsies that have been removed or monitored can be assisted by AI.

- **Augmented Reality (AR):** This enables the surgeon to see inside the patient and explore the body effectively.

3.4.4 Personalised Medicine

Reducing the time burden on any health practitioner is a tangible benefit of using ML for personalised medicine. This will also empower practitioners to treat more patients quickly and correctly. This advanced care is developed using EHRs, genetic data, and other patient information to train models. This leveraging of big data and predictive analysis techniques has created many opportunities for researchers to tackle and solve issues surrounding diseases, cancers, and depression (Dutta, 2021).

Throughout the Coronavirus Disease (COVID-19) pandemic, deciding upon the most effective line of treatment for medical practitioners and clinicians was a monumental challenge for them to answer quickly. As the world looked to medicine for the answers to open the world again, there was confusion amongst clinicians about the efficiency of using remdesivir or corticosteroid on patients with COVID-19 and if it leads to better survival rates. A ML algorithm was developed to assist with this.

Lam et al. (2021) answered this concern by utilising a gradient-boosted decision tree model for training and testing on adult patient data (aged $\geq$ 18 years) from 10 hospitals in the United States (US). They wanted to test the performance of both drugs on patients with longer survival times. Their findings were significant and were based on the Fine and Gray proportional-hazards models. The sample size for the experiment was 2364 patients. 893 patients had been treated with remdesivir and the remaining 1471 were treated with a corticosteroid. Their results were hazard ratios of 0.56 and 0.40, respectively (both, $P = 0.04$). This demonstrated that both groups of patients were less likely to have increased survival rates using either drug. This resulted in patients not being administered either drug.

There are many limitations to using AI in the development of personalised medicine. One such limitation is that many argue that big data analysis, that combines information on individual patients to reflect population-level relationships between data points, does not provide important individual-level relations. The lack of ergodicity within these results can mean that results are not beneficial for making treatment decisions for individuals (Fisher et al., 2018).

A second limitation is that there is a requirement to vet or test the utility of healthcare products that are developed using AI. This limitation is motivated by previous results that have been inconsistent when developed utilising AI – inconsistent results have been demonstrated in many AI-rooted health products, including IBM's Watson treatment decision system. Some existing healthcare tools have been tested via traditional randomized clinical trials, and some AI-based decision support tools have been accepted via these clinical trials (Schmidt, 2017; Abràmoff et al., 2018; Zhou et al., 2019).

It might be effective to implement AI-based learning systems with an ongoing review of their algorithms, parameters, and features to ensure the systems are always fit for purpose. This rolling training can require a large amount of retraining and computational resource to provide effective results (Schork, 2018; Ioannidis and Khoury, 2018; Frieden, 2017; Abernethy and Khozin, 2017; Nature, 2018).

Lastly, many AI-based decision support systems leverage algorithms that can be very difficult to interoperate. These systems rely heavily on deep learning and complex neural networks. Although the results for these trained models can be reliable (if a large enough training set is used), it can be very difficult to understand the interlinks between the inputs and outputs.

3.5 Machine Learning & Pharmaceuticals

Many monotonous customer service industry tasks have been early adopters of the new methods in the field of ML, whereas uptake in the medical and pharmaceutical industry has lagged behind. However, due to the low success rate of drug development (defined as phase I clinical trials to drug approvals) across the globe, there is a growing need for pharmaceutical companies to lower the costs in finding successful drugs. Quris recently released the "patient-on-a-chip" system which can be used to reduce the need for animal testing and speed up drug development (Coldewey, 2021; Taylor, 2022; Bein et al., 2022).

Many stages of the drug manufacturing and development process have been reviewed and areas of improvement utilising ML algorithms have been considered by major pharmaceutical companies. Some examples of where ML algorithms have been applied are:

- **Targeting disease associations**: Disease associations are the relationships between two or more diseases. These relations can be lifestyle-related, genetic, or environmental. There are many different published conference papers and academic journal papers which relate to targeting disease associations with machine learning. The work undertaken in 2012 by Iordanescu et al. (2012) aimed to use SVMs to identify new drug targets for Alzheimer's disease. The algorithm was trained on a dataset of 10,000 genes that are known to be associated with Alzheimer's disease. The algorithm produced a number of the potential drug targets that have not been previously considered for Alzheimer's. Some of these targets include proteins that are used in the production of the amyloid beta plaques – which are a pathological *hallmark* of Alzheimer's disease. The work showed real promise and demonstrated that ML can be a powerful tool for identifying new drug targets especially for Alzheimer's.

Lind and Anderson (2019) made use of the random forest algorithm to develop new drugs that are effective against all types of cancer. Due to the heterogeneous nature of the disease, there is a wide variety of different types of cancer. Cancer has many different symptoms and causes which lead to many different treatment types. The work undertaken was trained on a data set that contains 1001 cancer cell lines and 225 drugs, including experimental and approved anticancer drugs. The trained random forest classifier was able to predict the response of patients to new drugs with an >80% accuracy. The paper highlighted that additional validation would be needed to confirm the results of the classifier.

- **Improve the design and optimisation of small-molecule compounds**: Poly(ADP-ribose) polymerases (PARPs) are a class of enzymes that are critical in repairing DNA. Parp indicators are used as a new type of cancer drug. In 2019 Ai et al. (2022) used machine learning to design new PARP inhibitors which had an improved target affinity and selectivity. A dataset of 10,000 molecules was used with the random forest classifier to correctly develop the inhibitors.

- **Further understand the disease mechanisms**: Identifying diseases and some of the causes of diseases has been researched for many years. ML has shown promise in supporting this field of research. One such example is the work undertaken by Konovalov et al. (2021), the authors utilised a dataset with information on 20,000 patients. 10,000 of the patients had cancer and 10,000 did not. Their trained random forest algorithm was able to identify many bio-markers that were associated with cancer. These bio-markers were not identified making use of traditional methods. Furthermore, the model was able to identify bio-markers that were associated with specific types of cancer. The work demonstrated throughout this paper shows that machine learning has the potential to revolutionize the identification of bio-markers for cancer whilst also highlighting how this could be applicable to other diseases as well.

Due to the promising results in certain areas of pharmaceuticals many companies have continued to invest in ML or purchase start-up companies that specialise in ML for medicine. IBM, Google, and Amazon are utilising their cloud-based computation services to support the health care industry by working with partners such as GE healthcare (Vermeer and Thomas, 2020; Breant et al., 2018; J.D, 2022).

3.6 The Ethics of AI within Medicine

As more decision making processes are being supported by ML in healthcare, it is important that all of the ethical concerns that usually arise from an ML problem are considered. This is of paramount importance when considering medical problems, as the data provided must also be subject to legislation already in place for dealing with medical data. There are three core areas that consideration should be given to:

1. **Sharing Patient Information:** Naturally there are restrictions in place around sharing patient information. It is important to ensure that data is not shared that could be traced back to an individual patient. The core of effective ML is effective, organised, and clean data. As part of the cleaning process, typically hospital staff will make sure the data is General Data Protection Regulation (GDPR) compliant by removing any identifiable information. In a medical setting, data can be shared for medical reasons. For example a doctor may share patient information with a surgeon or another doctor to get feedback or for a second opinion on a decision that could effect a patient's health.

2. **Patient & Clinician Autonomy:** Throughout the health-care industry there are different types of patients. A vulnerable group of patients are those who are incapable of making health care decisions themselves. It is possible to use ML in conjunction with electronic health records to assist in making these clinical decisions. However, there is a strong argument that ML should not replace patient or clinician autonomy, but instead should support (rather than replace) clinical decision making.

Tools that are developed using ML techniques must be used as support tools to help inform the decision but not to make the final decision.

3. **Patient Safety:** *"Garbage in, Garbage out"* is a cliché often used in ML. The general meaning is that if you pass flawed information into a model, you will get flawed predictions. The systems reliability can be undermined when using erroneous data. As a result, models should be used with caution until the quality of data used to produce the model is verified.

   Similarly, cultural bias encoded in datasets can mean results can be biased against certain ethnic backgrounds or cultures. This is very important for medical decisions as it can result in over-diagnosis or under-diagnosis which can mean some patients will be treated when not needed and conversely not treated when required. These mistakes can be life changing.

   Patient safety and the resulting outcomes should always be considered when looking at the predictions from ML models that could be life-changing. As a direct result, all of the results presented in this body of research should be taken as indicative. Additional required work and the limitations of this work are discussed in Chapter 7 (Yoon et al., 2021).

All patient data that is used as part of this research is GDPR compliant and anonymised to ensure that the patient can not be traced. The interpretability and explainability of the models developed is an important ethical step that has been considered throughout the research. Some additional questions that are answered in this body of research are:

1. Does the dataset contain any sensitive data?
2. Does the training dataset accurately represent the source population?
3. Can developers examine the logic behind the code base?
4. Are the patients made aware that their data will be used in the study?

3.7 Summary

As described previously, the AI market share within medicine has risen by nearly 40% in recent years and will soon eclipse $31B USD. Medicine is one of the areas where AI techniques are starting to be utilised as support tools for clinicians to improve the speed of diagnosis, perform minor surgeries and for drug discovery. The emerging paradigm of AI in medicine is described in this Chapter and where the areas for research and improvements lie.

This Chapter has aimed to provide an overview of the current state of the industry and define some common use cases to date. In section 3.1 the changes to record keeping from paper to electronic health records is described. The benefits of this development are defined and how it can lead to the ethical sharing of patients records for research purposes is also explained. Moving to EHRs has also seen an improvement in the integrity of the data collection, section 3.2 talks about this and some of the dangers of having missing data within records. Predictive analysis and what these advances can mean to the industry is also described in section 3.3 with an emphasis on how it can be used to spot areas that need additional research.

There are many different applications of ML in industry, section 3.4 focuses on these applications and details some existing academic research or use cases. Disease identification and diagnosis and the work undertaken withing research as well as in a clinical setting are outlined in subsection 3.4.1. A brief overview of robotic surgery and how it could revolutionise routine surgeries are introduced in subsection 3.4.3. In addition, subsection 3.4.4 introduces the concept of personalised medicine and what it can mean for improved medical diagnosis by combining EHRs, genetics and other patient information to produce effective ML models by leveraging all the datasets available.

Pharmaceutical companies and leading technology companies are working together to improve the drug discovery process and improve the time taken to find solutions to existing medical conditions. The steps involved in this and some of the exact use cases are described in section 3.5.

Due to the sensitive nature of the work undertaken in this research, it is important to make sure that all the data is utilised ethically. Ethical approval was provided at the start of this research. Section 3.6 outlines some of the ethics around ML with an emphasis on the medical domain. Lastly how the data in this research is ethically compliant is described at the end of the Chapter.

Chapter 4

Mortality Prediction in Intensive Care Units

4.1 Introduction

With National Health Service (NHS) waiting times failing to meet targets for over 16 months now and government cutbacks to nursing and hospital staff, innovative ways of diagnosing and assessing patients will soon be introduced (Campbell and editor, 2017). State of the art technology such as bespoke medicines and computational models will be made use of. Computational models that use ML algorithms have already been tested using real-world hospital data, with promising results in predicting mortality rates in patients at the ICU at North Middlesex hospital (Shenfield et al., 2017).

Predicting the survival of a critically ill patient is a difficult task. There have been many different scoring systems designed that have been used to grade the severity of a patient's illness. These systems include:

- **Acute Physiology and Chronic Evaluation II APACHE II:** The APACHE II score is currently the most commonly used system for classifying the severity of disease of patients admitted to critical care units. It is usually applied within the first 24 hours of admission to ICU, and uses a combination of physiological variables, the patients age, and the patients chronic health status to determine mortality rate. Although later versions of the Acute Physiology and Chronic Evaluation (APACHE) score exist, the most commonly used version is APACHE II due to later versions requiring more diagnostic tests (Wagner and Draper, 1984).

- **2$^{\text{nd}}$ Simplified Acute Physiology Score (SAPS II):** The SAPS II score was introduced in 1993 as an alternative to the APACHE II score. The features required for the SAPS II score should be collected within the first 24 hours of admission in to the ICU. The Area-Under the Receiver Operating Characteristic Curve (AUROC) generated for the SAPS II score was 0.86, this was higher than the original SAPS score that was 0.80. The score can be used to estimate mortality risk of a group of patients however it is not intended to describe the chances of survival of a patient (Le Gall et al., 1993).

In this Chapter, the application of different ML techniques for accurately predicting mortality in ICU is discussed and compared. The key contributions of this research are:

1. Evaluation of existing methods used for mortality prediction in ICU is compared.

2. Development of an effective ML pipeline provides an accurate prediction of mortality that could be used as a support tool by medical practitioners.

3. Comparison of the proposed ML pipeline with the existing state-of-the-art research.

4. Investigate the effects of retraining ML models varying ages and date of submission for different patient cohorts.

The remainder of the chapter is structured as follows. Section 4.2 evaluates the existing state-of-the-art research, section 4.3 describes the proposed novel approach to developing a ML pipeline and validating the results of the proposed system as well as describing how the online approach to training will be utilised and tested. Section 4.4 describes the dataset utilised throughout this Chapter. The results themselves are discussed in section 4.4.2 and compared to existing state-of-the-art research methodologies using confusion matrices and Receiver Operating Characteristic (ROC) curves. Section 4.5 presents the methodology and results for online learning while utilising different age and date of submission patient cohorts to understand the effects and accuracy of results. The conclusions section 4.8 also examines the limitations of the proposed approaches and outlines further work.

4.2 Review of Existing Predictive Risk Mortality Research

Even though there are multiple new methods for determining mortality within ICU, the APACHE II and SAPS II scores continue to be the most used point-based schemes worldwide (Keuning et al., 2020). Similarly, the Sequential Organ Failure Assessment (SOFA) is used in some parts of the world as a mortality risk assessment tool, even though it was developed to assess sepsis risk (Arts et al., 2005). Some common limitations that are associated with these tools that have been detailed in the literature are:

1. There has been a decrease in performance over time. Kramer (2005) indicated that SAPS II was not within calibration tolerance by 2005.

2. There have been some calibration issues with both APACHE II and SOFA scores (particularly when applying them to new patient cohorts (Sakr et al., 2008)).

3. Sakr et al. (2008) and Lew et al. (2019) noted that the tools were not very reliable for patients within Europe or Singapore, as they were not developed with data from these patient cohorts.

4. Some variables which are required to provide a score are difficult to obtain, especially when patients are admitted into critical care situations. For many cases, the data might not be available because it requires expensive pathological laboratory tests and full patient medical history.

These limitations have led to researchers exploring alternative approaches for mortality prediction. The resurgence of ML techniques has provided some promising preliminary results in this problem domain. Furthermore, online ML models are comparatively easy to update, retrain, and re-calibrate for different patient cohorts and as patient cohorts evolve over time (Lew et al., 2019). Traditional approaches to mortality prediction often only capture a single time period; this approach misses out on valuable insights and data that could improve the models accuracy, precision or recall as things change over time. Online learning can learn from new examples in real-time, ensuring that the model constantly generalises well to the populations it is applied to, even as environmental factors, operations, and medicines change over time.

A standard metric used in medicine to determine the performance of diagnostic tools is the AUROC. AUROC is the “Area under the curve” for the Receiver Operating Characteristic curve. The AUROC score is a way of measuring how successful a binary classifier is at distinguishing between classes (a detailed description of AUROC curves is in section 4.4.2).

For practical application, a mortality risk prediction model should only use vital signs that can be continually monitored and should allow the doctor to see how the risk will change. Deliberato et al. (2009) have developed a model using purely vital signs. However, the models AUROC of 0.65 showed that it is a poor discriminator between mortality and non-mortality cases. They also used a combination of vital signs and additional features culminating in a higher AUROC of 0.85 when the data was combined with SAPS II score and patient demographic.

Throughout the literature, there are many ML techniques used to consider the prediction of mortality. However, there has been little focus on predicting mortality at admission (first 24 hours) to the ICU. One of the fundamental disadvantages of this methodology is that it does not consider complications that occur after admission. Neural networks (introduced in Chapter 2.7.7) have also started to be considered for mortality prediction. Some works have focused on using simple feed-forward networks that are able to produce results comparable to APACHE II. Shenfield et al. (2017) used ANNs and the JADE optimisation algorithm to obtain an accuracy of over 90% when at decision criteria between 30-80%, with an AUROC score of 0.932.

An AUROC of 0.836 was achieved by Alves et al. (2018), who used Convolutional Neural Networks (CNN) layers before Long-Short Term Memory (LSTM) layers, significantly improving accuracy over purely LSTM layers. CNN have been proven to be valuable tools for solving medical problems. Samir et al. (2021) used CNN to predict heart anomalies accurately. Similarly, Bukhari et al. (2020) predicted gait detection correctly making use of CNN.

Karabulut et al. (2012) outlined that the selection of features is an important step in developing all ML models. To develop a model that can be automatically updated throughout a patient's stay, the features must be easy to obtain, measure, and repeat (preferably with no manual intervention required from clinicians).

The Artificial Intelligence Mortality Score (AIMS) (Baker et al., 2020) scheme uses a hybrid CNN-LSTM network with a combination of age, gender, and a selection of statistical parameters obtained within the first 24 hours of admission into the ICU. AIMS achieved an AUROC score of 0.884 – 0.858, depending on the length of stay within the ICU. In the AIMS system, scores are generated over 3-day, 7-day and 14-day windows. Yu et al. (2020) used

LSTM techniques to determine mortality and to take complications in to account. Forty-eight hours of feature recording is needed to predict mortality effectively using this method. They obtained an AUROC score of 0.885 using a bi-directional LSTM.

The research described above relies on features containing complex diagnostic results, details about exiting health conditions, and previous patient histories. The previously mentioned studies that look at mortality risk prediction use a diverse feature set mainly made up of laboratory results that include blood tests, urine samples, breath monitoring, and other complex measurements that can take time to obtain. As described in Chapter 7, one of the main factors affecting the uptake of ML in the medical domain and the success of ML models within the industry is the lack of transparency and interpretability of models. Using common features (e.g clinical laboratory tests and vital signs) helps overcome these problems by allowing results to be interpreted by domain experts.

4.2.1 Mortality Prediction in Real-Time

Existing academic research that aims to investigate and improve on existing support tools for medical practitioners dealing with patients admitted in to ICU look at a single snapshot in time and do not demonstrate how their models perform overtime. These approaches can result in models that are not effective against new medicines. The medical landscape with vast amounts of new treatments, prescribed antibiotics, and medical recommendations is changing rapidly. Biomedical research has resulted in breakthrough accomplishments which has seen the eradication of many life threatening diseases and viruses such as polio and improved life saving options for Acquired Immune Deficiency Syndrome (AIDS), cancer, and COVID-19. Due to the exponential rise of treatments and solutions the United States of America (USA) has seen the number of drugs which are Food and Drug Administration (FDA) approved rapidly increase (Craven, 2022).

The continued development of medical advancements has seen a large growth in medical understanding, increased complexity of medical practice and more experts with medical specialism. Ideally as all fields of medicine improve other areas should look to keep up and improve. The fact that the APACHE II score (even though it is flawed) is still utilised today demonstrates that improvements are required in this area.

4.3 Machine Learning Pipeline Development

To test the hypothesis that ML techniques can develop a mortality risk prediction tool that provides similar or better accuracy than existing support methods currently being used within ICU. The ML pipeline development framework shown in Figure 4.1 is used. The framework is made up of the following steps:

1. The complete dataset is split into 3 sections. 70% is used as the training dataset, 20% is used as the testing dataset and 10% is held back as the unseen data.

2. The training data is used to train multiple different types of classification

models using standard hyperparameters. The 20% test data is used to identify top 3 classifiers based on varying performance metrics.

3. The top three trained models (based on accuracy) then use different hyperparameter optimisation and data re-balancing techniques to try and find the most effective ML model according to the performance metric of interest.

4. The resulting model is then tested using the 10% unseen data to generate final scores and to check that the model generalises to unseen data and doesn't show signs of overfitting.

In this Chapter, K Nearest Neighbour, Linear SVMs, Radial Basis Function SVMs, Gaussian Process Models, Decision Trees, Random Forests, ANNs, AdaBoost, Naïve Bayes, and Quadratic Discriminant Analysis (QDA) are investigated. Each classifier is trained and tested using repeated 10-fold cross-validation, with the average accuracy calculated for each classifier. Stratification techniques are used to ensure that the training and testing sets reflect the overall class imbalance of the data (see Section 5 for more details).

Once the base classifiers are trained and the results are generated for each classifier, the top three classifiers are selected based on the accuracy of the model - accuracy is utilised as it indicates as a percentage how many results the model got correct (Walker et al., 2020). If the target classes are imbalanced, synthetic data is introduced to rebalance the dataset using the Synthetic Minority Oversampling Techniques (SMOTE) (Chawla et al., 2002). The combination of synthetic data and real data is then used to retrain the top three performing models. The test runs are repeated multiple

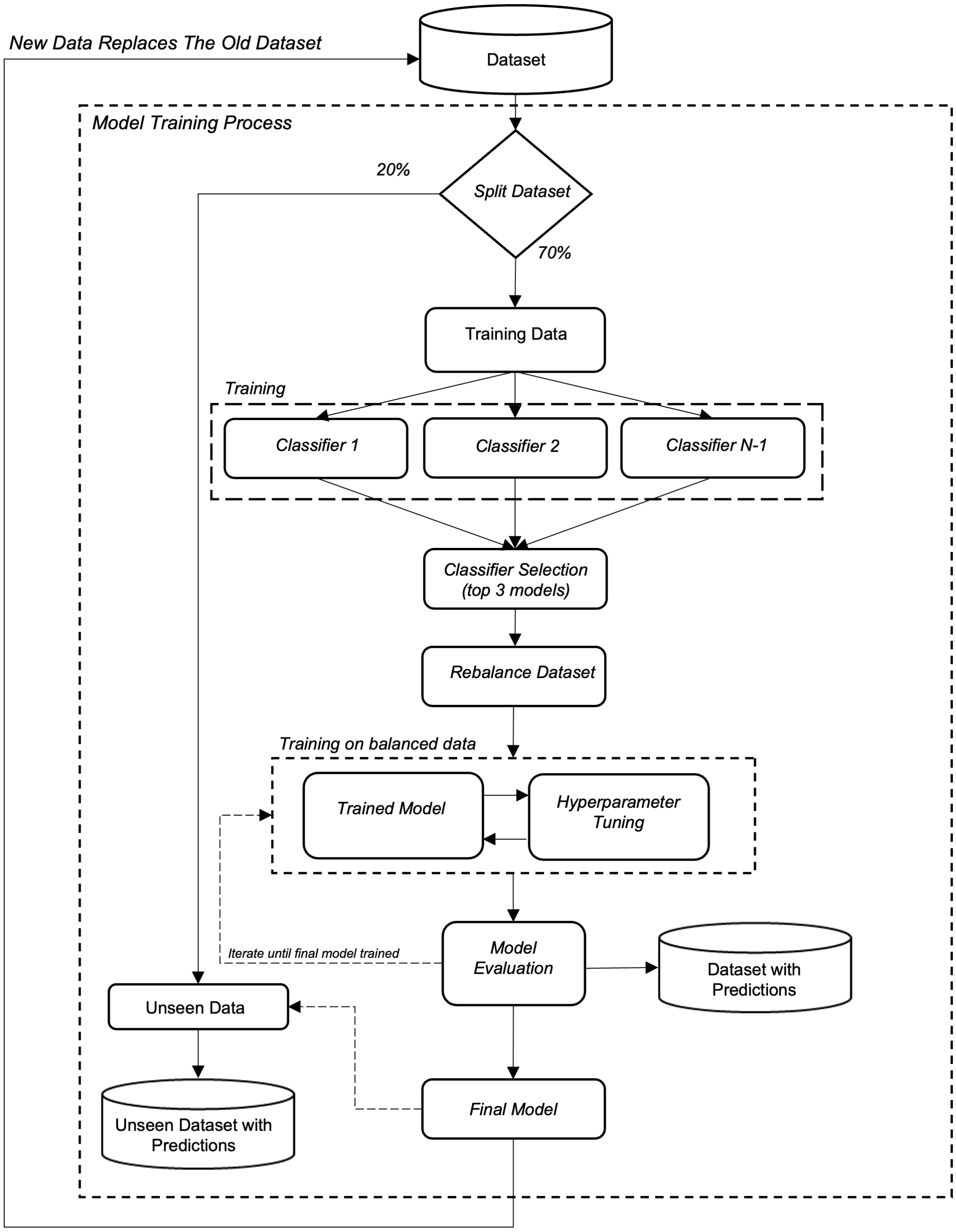


Figure 4.1: Machine Learning Pipeline Methodology for developing a mortality risk prediction tool

times to check the results obtained are consistent and the averages are generated, furthermore, it enables the spotting of trends and patterns forming within the results. It also reduces the variance in the overall results which is presented by showing the standard deviation ranges. Each model then has the hyperparameters optimised using both random search and grid search optimisation methodologies (see section 4.3.4). The performance metrics used in this Chapter are described in section 4.3.2.

The final models are again compared using both the accuracy score and the AUROC score. The most effective model is then tested on the unseen dataset to make sure that the model is not overfitting on the training/validation dataset. The following sections describe the dataset and different techniques used in the development of the ML pipeline.

4.3.1 Feature Importances

Feature importances can provide a deeper understanding of a dataset. The scores can demonstrate which features are most relevant to the target variable and, conversely, which are least relevant. This can then be interpreted by a domain expert to either remove unnecessary features or to collect more useful data. The scores can also provide further insights into the model. For example, inspecting the different scores that result from using different ML algorithms can show how different features have different effects depending on the variant of model used. Lastly, feature importances can improve the predictive model by empowering the developer to remove unnecessary features, resulting in a model that can be trained quicker. This deletion of features is commonly referred to as dimensionality reduction.

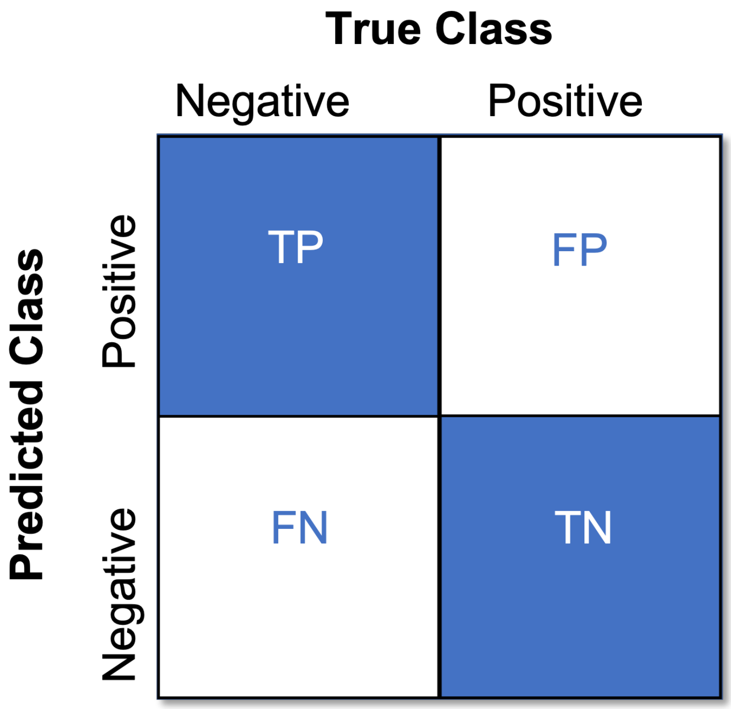
4.3.2 Performance Metrics

Confusion matrices sometimes known as error matrices, are often used to summarise the prediction results of classification problems. They are commonly used in ML projects due to how easy they are to interpret. A typical confusion matrix for a binary classification problem is shown in Figure 4.2 and shows:

- **True Positive (TP):** That is the number of correctly classified examples that are positive.
- **True Negative (TN):** That is the number of correctly classified examples that are negative.
- **False Positive (FP):** That is the number of negative examples that are mis-classified as positive.
- **False Negative (FN):** That is the number of positive examples that are mis-classified as negative.

Confusion matrices are used to know how many mistakes a classifier makes, and what those mistakes are. Correct predictions are shown in the diagonal entries (blue squares with white text in Figure 4.2). A classifier performing well should contain minimal examples in cells that are not on the diagonal. The Error Rate (ERR) and accuracy can be calculated using the equations 4.1 and 4.2.

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{4.1}$$



		True Class	
		Negative	Positive
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 4.2: An Example of a 2x2 Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

The True-Positive Rate (TPR) and False-Positive Rate (FPR) are specified as:

$$TPR = \frac{TP}{TP + FN} \tag{4.3}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.4}$$

When a classification task has significantly imbalanced target output classes, accuracy should be used with caution. If 95% of the dataset consists of the positive class, simply always predicting a sample as positive yields an accuracy of 95%, which is misleading. By making use of the TPR, it is possible to see how well the classifier is performing even if the classes are unbalanced.

Precision (PRE) means “how many of the predictions made are correct?” whereas Recall (REC) means “how many positive points in the output are successfully identified as being positive?” PRE and REC are very useful in the medical domain, as it is essential to understand the performance of the optimistic predictions.

PRE and REC metrics are calculated using:

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

The combination of PRE and REC into a single score is known as the F1 score, and is defined as:

$$F1 = \frac{(2 * TP)}{((2 * TP) + FP + FN)} \tag{4.7}$$

4.3.3 Cross-Validation

In K-fold cross-validation, the original dataset is partitioned randomly into K equal sized partitions. A single K partition of the data is retained and kept unseen (this is used as the validation data for testing the model) and the remaining K-1 subsets are used as the primary training data. This cross-validation process can then be repeated K number of times, with each different subset used once, as the validation data. This has many advantages over other validation techniques as all of the observations are used for both the training and validation data. Furthermore, each validation set is used precisely once meaning that the models aren't just trained on the same test and train data sets. Figure 4.3 shows how the data can be partitioned for each fold. Stratified K-fold cross-validation was introduced to address datasets that are not evenly balanced between the different classes. The data for each fold is selected so that each fold contains a similar proportion of class labels to that within the whole dataset. Repeated cross-validation repeats the cross-validation a given number of times. These results are then averaged out to produce a better estimation of the model performance.

4.3.4 Rebalancing Datasets

The process of re-balancing a dataset is often used in real-world dataset classification tasks where the majority of results fall within a single class (known as the majority class). A dataset can be said to have a ratio of 4:1 if, of the 100 records, 80 belong to the majority class and 20 belong to the minority class. There are many techniques that can be used to rebalance datasets. Chapter 6 focuses on this problem but, in this Chapter, the SMOTE technique will be used.

SMOTE is a method used to auto-generate new synthetic instances of data from the minority class within an unbalanced dataset. SMOTE works by adding points around the minority classes instances. New instances are created by combining existing instances, therefore minimising (but not eliminating)

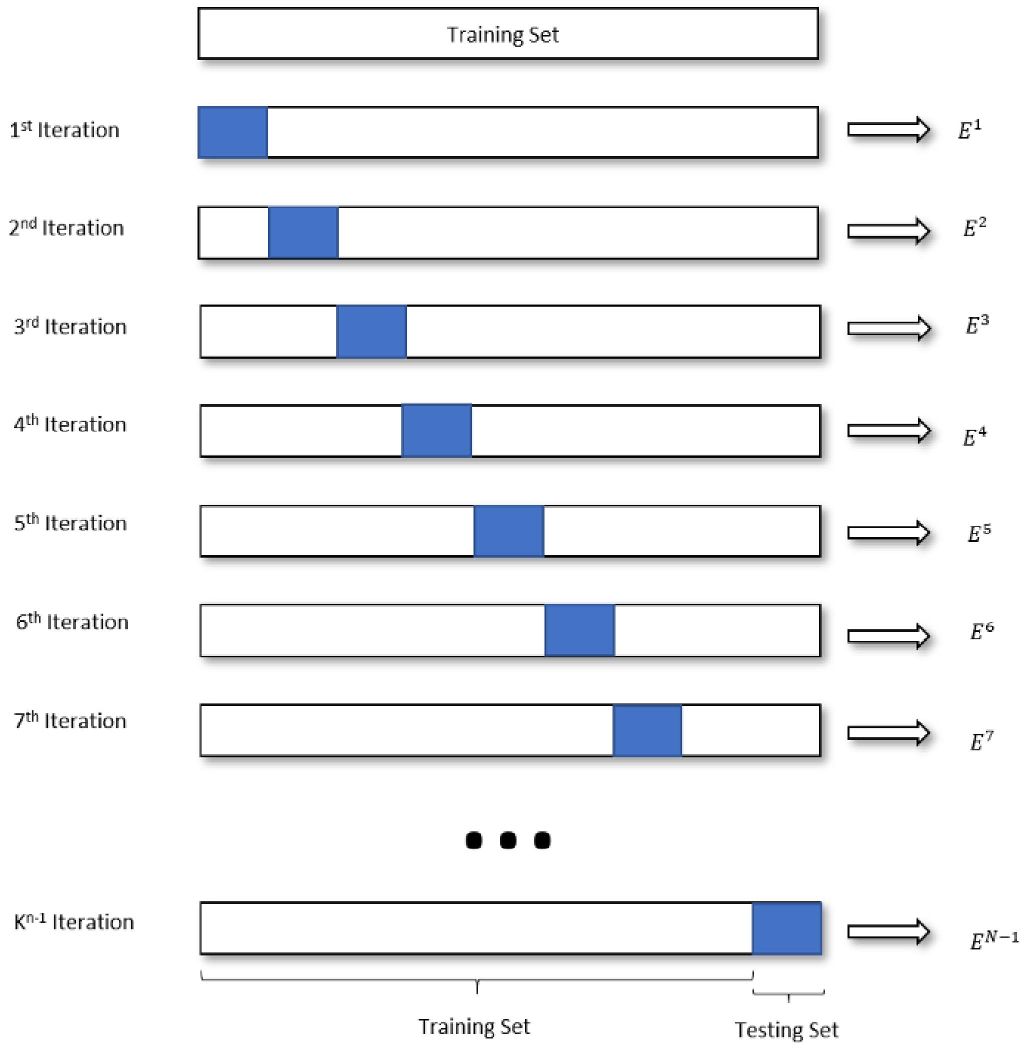


Figure 4.3: An illustration of K-fold cross-validation

the disadvantage of overfitting. Points within the minority class are selected, and then synthetic data with similar attributes is imputed within the feature plane. Figure 4.4 shows an example of SMOTE, and how it can be applied to data (Chawla et al., 2002).

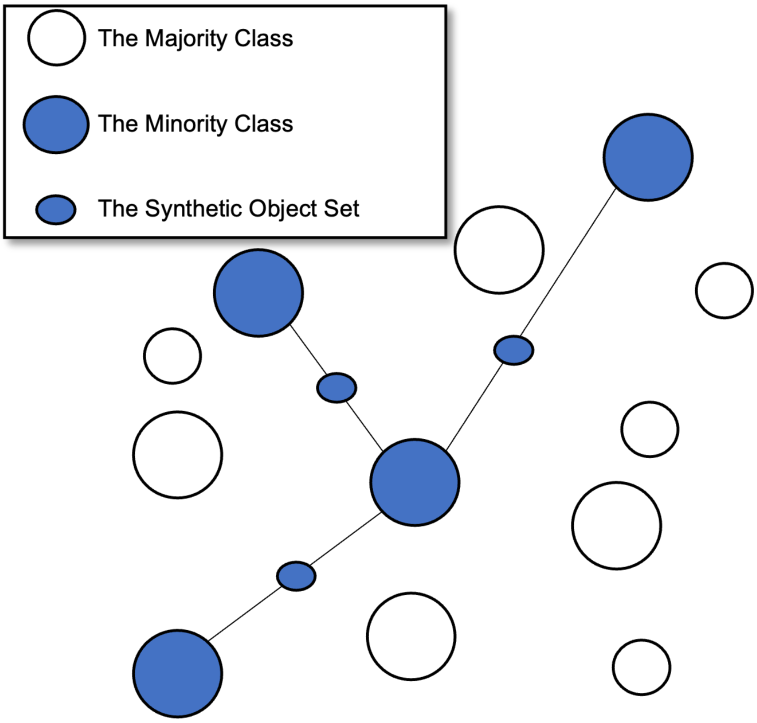


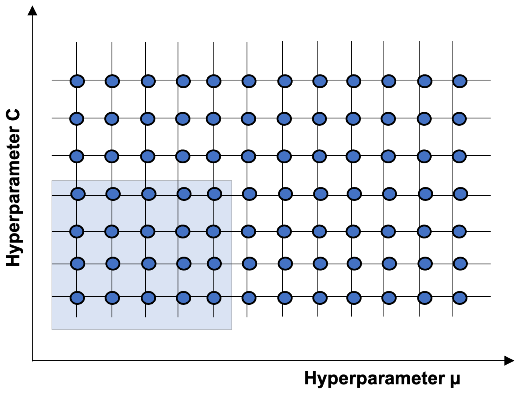
Figure 4.4: An overview of the synthetic minority oversampling technique

4.3.5 Hyperparameter Optimisation

ML techniques have multiple parameters that are critical in the training of the model and controlling the accuracy of the resulting trained model. As a result, the tuning of hyperparameters is an important step within any predictive model development. The learning rate of a Neural Network is an example of a hyperparameter, and is defined before the model is trained. Conversely, the weights of the Neural Network are optimised during the training stage of the ML model development using the specified hyperparameters. There are several common methods used to find a good set of hyperparameters and the following subsections (4.3.5 and 4.3.5) will describe two of these methods - grid search hyperparameter optimisation and random search optimisation.

Grid Search

Grid search is the most commonly used technique to optimise hyperparameters in conventional ML. This brute-force approach iterates over every defined combination of a specified set of hyperparameter values, kernels, or training methods to find the result that provides the best performance. Figure 4.5 shows a set of combinations of different values that will be tested for hyperparameters 1 and 2. It is computationally expensive to try all the various combinations of hyperparameters for many real-world problems, particularly as the problem space becomes more complex. An alternative technique is random search.



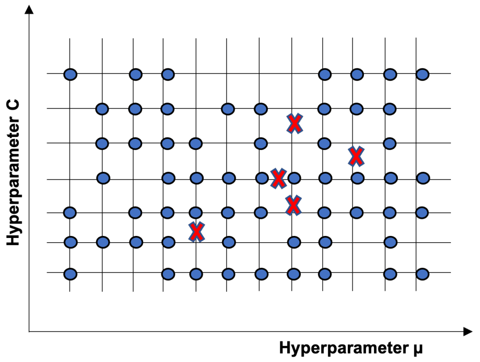
Parameter Ranges	Possible Test Sets		
C = (10, 100, 1000)	0.1, 10	0.2, 10	0.5, 10
μ = (0.1, 0.2, 0.5)	0.1, 100	0.2, 100	0.5, 100
	0.1, 1000	0.2, 1000	0.5, 1000

Figure 4.5: An overview of the grid search algorithm and showing all of the possible combinations for the two hyperparameters

Random Search

Using random combinations of the hyperparameters to find the optimal set for the constructed model is known as random search. A common drawback of using random search is the variance that is introduced during computing. Figure 4.6 shows a visual description of how random search can be applied to a dataset.

Random values within a set of bounds are selected for the hyperparameters at each iteration of the testing. The model is then trained and evaluated with that set of hyperparameters (often using cross validation techniques) and then a new set of hyperparameters are selected at random and the process starts again. This iterative approach, combined with the randomness, typically means that a large amount of the search space is considered. The random search will continue to run for a finite number of iterations at which point the training will stop.



Parameter Ranges

C = (10, 1000)

μ = (0.1, 0.5)

Possible Test Sets

0.1, 500

0.5, 600

0.25, 700

0.2, 780

0.15, 800

0.114, 820

Figure 4.6: An overview of the random search algorithm showing all of the selected combinations for the two hyperparameters considered here.

4.4 Mortality Prediction using Machine Learning

4.4.1 ICNARC Dataset

The research in this section was undertaken using the ICNARC dataset that was collected at North Middlesex University Hospital cluster between January 1st 2012 and April 30th 2014. The dataset consists of 13,494 patient records, where each row corresponds to a patient admitted into the ICU. There is no missing data in the dataset.

The dataset is comprised of 41 features. As well as the physiological features, there is some additional patient information collected; including patient age at the time of admission into the ICU, whether the patient had Cardiopulmonary Resuscitation (CPR) within 24 hours of admission, the location of the patient before the admission (which is often referred to as the source), and whether the patient was intubated during the first 24 hours. All of the features are defined in Table 4.1.

Table 4.1: Features of the ICNARC Dataset

	Used Variable	Utilised
1	Anonymised Unit Identifier	
2	Age in years at last birthday	
3	Gender	
4	Residence Prior to admission	
5	Prior Dependency	
6	Severe Liver Disease	
7	Haematological Malignancy	
8	Metastatic Disease	
9	Severe Respiratory Disease and Home Ventilation	
10	Immunocompromise	
11	Cardiovascular Disease	
12	Renal disease	
13	CPR within 24 hours prior	
14	Primary reason for admission	
15	ICNARC Diagnostic Category	
16	Condition Description	
17	Type of Admission	
18	Mechanically Ventilated at admission	
19	Highest level of care received in unit within 24 hours	
20	Basic respiratory support	
21	Advanced respiratory support	
22	Basic cardiovascular support	
23	Advanced cardiovascular support	
24	Renal support whilst in unit	
25	Neurological support whilst in unit	
26	Gastrointestinal support whilst in unit	
27	Dermatological support whilst in unit	
28	Liver support whilst in unit	
29	APACHE II score	Removed
30	ICNARC model physiology score	Removed
31	Your unit survival	Removed
32	Your hospital survival	Removed
33	Expected dependency post-discharge from your hospital	Removed
34	Date of admission to your hospital	Removed
35	Date of discharge from your hospital	Removed
36	Date of admission to your unit	Removed
37	Date of discharge from your unit	Removed
38	Date of death	Removed
39	Date of declaration of brain stem death	Removed
40	Readmission within same hospital stay	Removed
41	**Died or Survived**	**Target Variable**

Not all features of the dataset were utilised throughout the study, as with many ML models, the dataset is explored to removed or encode features to make sure they can be used by the ML classifiers. For this body of work all date-time type features were removed from the study as they many of them are only available if a patient has died and helps to mitigate overfitting. 4.1 Further to this point, the APACHE and ICNARC scores were removed from the model. There was no missing data present in the dataset so no features were removed because of that.

The mean age of the patients within the given ICNARC dataset is 60 years old. The minimum age of patients is 10 years old whilst the maximum value is 103. Figure 4.7 shows the count of different ages present within the dataset split by gender. There are more male records (55.05%) with only 44.95% identified as female. The average length of stay for those admitted into the ICU is 17 days.

There are 675 different condition descriptions described within the dataset in the *condition1desc* column. The most common description is those who have been admitted into the ICU with Pneumonia (6.75%). Figure 4.8 is a word cloud of all of the descriptions.

The total number of patients present in the dataset who passed away in the ICU is 1668.

4.4.2 Model Development & Evaluation

Section 4.3 discusses the steps taken to produce an effective ML pipeline. As part of the exploratory data analysis stage, it is important to understand if the class labels are imbalanced. Figure 4.9 shows the distribution of mortality in the dataset, with 11,838 patients surviving and 1,675 dying. As a result,

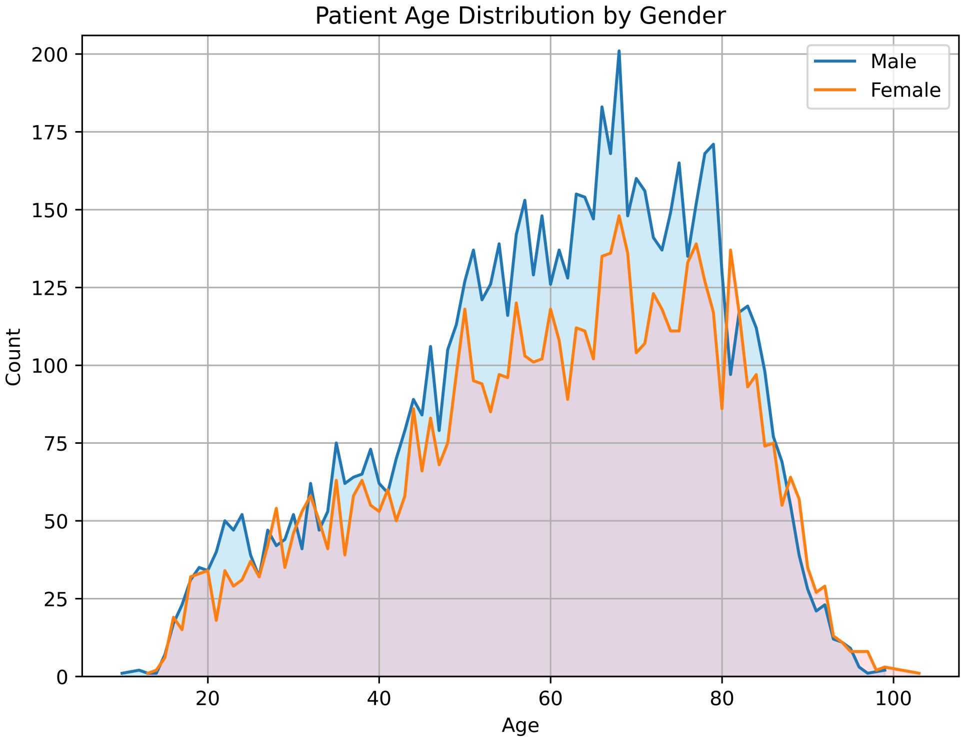


Figure 4.7: The gender distribution within the ICNARC dataset.

following the classifier comparison stage and hyperparameter optimisation techniques, SMOTE is used to rebalance the minority class (see section 4.3.4 for more information).

The ICNARC dataset (outlined in section 4.4.1) was used to train the ML classifiers described in section 4.1 using a variety of different performance metrics to determine the suitability of the classifiers. The results of the preliminary stages of training are presented in Table 4.2. The results obtained show the scores for PRE, REC, F1-score, and accuracy. Comparing the results from Table 4.2 to those described in section 4.2, it can be seen that the Decision Tree and AdaBoost classifiers obtain a comparable overall classification accuracy (91% and 90% compared to the greatest classification accuracy

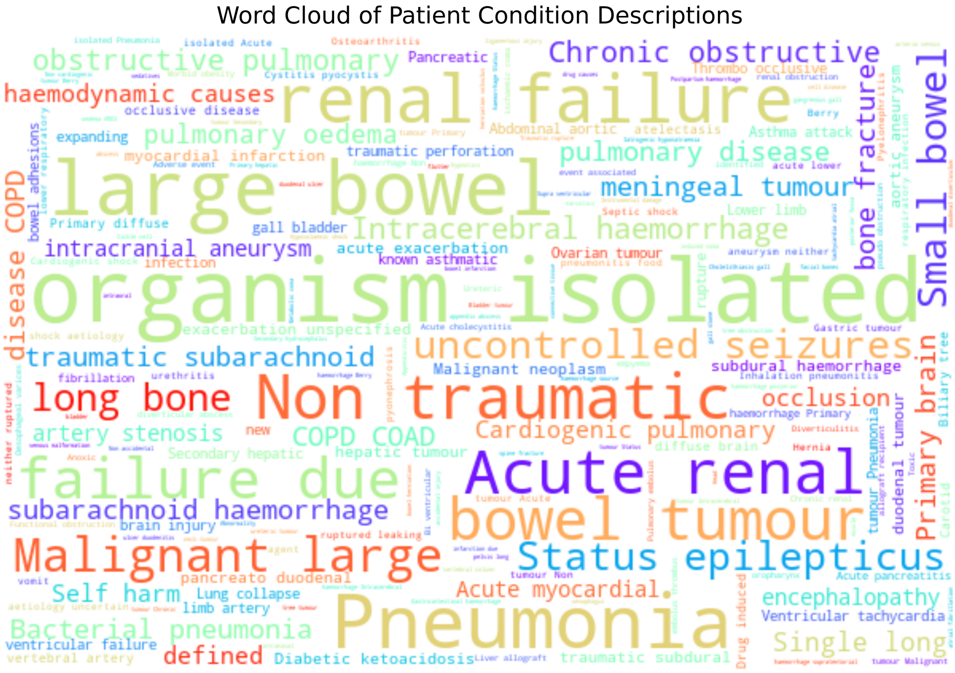


Figure 4.8: A word cloud of the most popular patient conditions within the ICNARC dataset.

described in literature that utilised a similar dataset as 90% (Shenfield et al., 2017)).

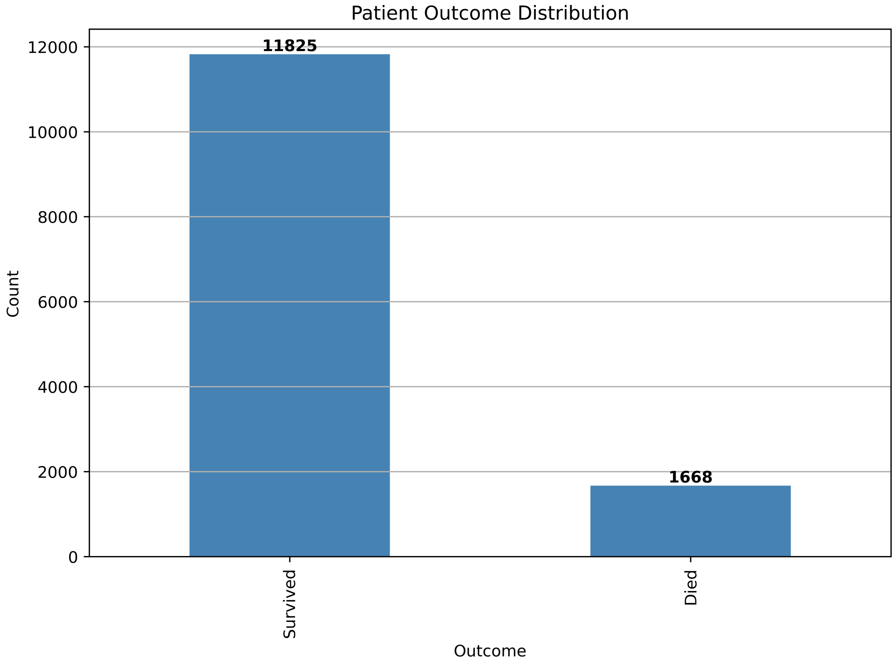


Figure 4.9: Histogram to show the distribution of patient outcomes in critical care units as part of the ICNARC dataset

Table 4.2: Performance of different base classifiers (with the top 3 results presented in bold)

Classifier	**Precision**	**Recall**	**F1-Score**	**Accuracy**
K-Nearest Neighbour	0.84 (0.14)	0.85 (0.10)	0.84 (0.10)	0.85 (0.12)
Linear SVM	0.89 (0.08)	0.90 (0.07)	0.88 (0.08)	0.89 (0.04)
RBF SVM	0.75 (0.03)	0.87 (0.04)	0.81 (0.02)	0.87 (0.04)
Gaussian Process	**0.90 (0.01)**	**0.90 (0.01)**	**0.90 (0.01)**	**0.90 (0.02)**
Decision Tree	**0.90 (0.09)**	**0.91 (0.04)**	**0.89 (0.02)**	**0.91 (0.03)**
Random Forest	0.85 (0.07)	0.87 (0.07)	0.88 (0.03)	0.87 (0.10)
Neural Network	0.88 (0.02)	0.89 (0.03)	0.88 (0.02)	0.87 (0.03)
AdaBoost	**0.89 (0.05)**	**0.90 (0.03)**	**0.89 (0.04)**	**0.90 (0.02)**
Naive Bayes	0.87 (0.03)	0.82 (0.02)	0.84 (0.02)	0.82 (0.18)
QDA	0.87 (0.04)	0.83 (0.01)	0.85 (0.02)	0.83 (0.08)

Confusion matrices for the top three performing classifiers (i.e Decision Trees, Gaussian Process Models, and AdaBoost) are show in Figures 4.10, 4.11, and 4.12 respectively. Additional information and a complete set of confusion matrices produced for the classifier comparison stage is available in Appendix A.

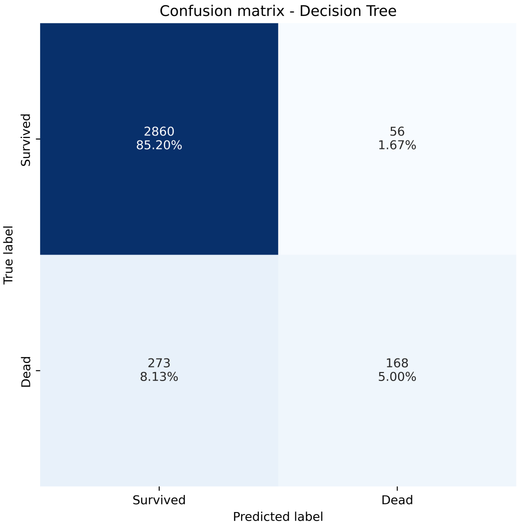


Figure 4.10: Resulting confusion matrix produced from preliminary testing using a Decision Tree to infer mortality in critical care units

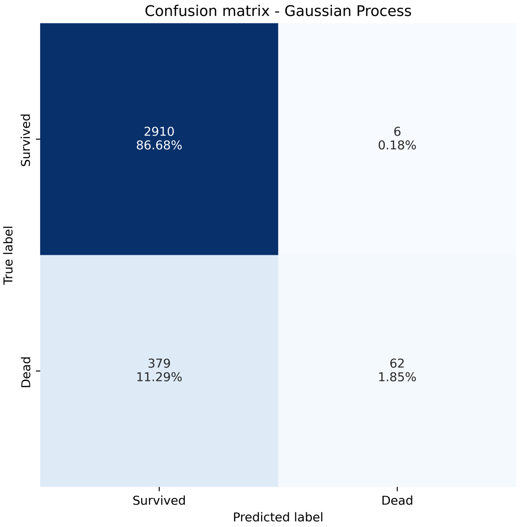


Figure 4.11: Resulting confusion matrix produced from preliminary testing using a Gaussian Process to infer mortality in critical care units

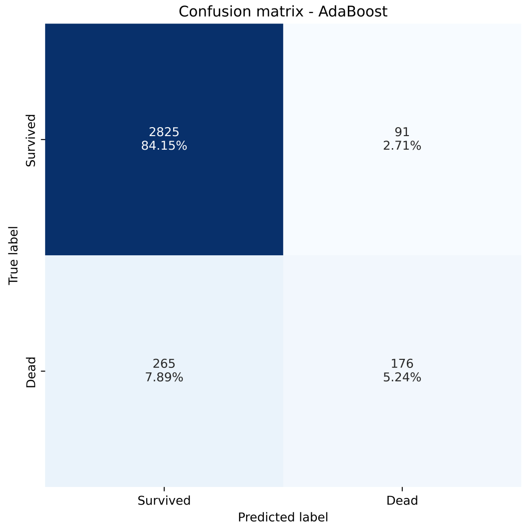


Figure 4.12: Resulting confusion matrix produced from preliminary testing using AdaBoost to infer mortality in critical care units

4.4.3 Classifier & Hyperparameter Optimisation

Each of the top three preforming classifiers (from section 4.4.2) were retrained using both Grid Search hyperparameter optimisation and Random Search hyperparameter optimisation techniques, to determine the most effective hyperparameter set for each model. For each set of parameters, AUROC score was produced and tests were repeated multiple times to reduce the impact of stochasticity.

For the Decision Tree classifier the maximum leaf nodes is tuned in the training process, where as in the a AdaBoost classifier the n_estimator (number of estimators) and learning rate are the hyperparameters that were modified in the tuning phase. Lastly, the Gaussian process algorithm that is typically used for regression problems, has the kernal and optimiser selection modified throughout the tuning process. The list of parameters that were modified remained consistent with the same thresholds (limits) maintained throughout the experimental process.

Tables 4.3 and 4.4 show the mean AUROC scores and the standard deviation over five runs using different hyperparameter optimisation techniques. The Gaussian Process Model produced an average AUROC of 0.77 using grid search. This demonstrates that the Grid Search hyperparameter optimisation technique was not capable of finding a good set of parameters from the specified search space. The best AUROC values using grid search were produced by the Decision Tree classifier (0.92) and the ADAboost algorithm (0.91).

The average AUROC scores produced when using the random search hyperparameter optimisation were typically higher than those obtained when using the grid search methodology. This is often seen as the grid search methodology can miss local optima due to the parameters not included within the defined parameters to search. The highest AUROC was using Decision Trees (with an AUROC of 0.93).

For both runs of training the model with random search and grid search, the results and standard deviation of the AUROC for the Gaussian process varied quite a lot. This demonstrates that the hyperparameter selection stage is very important for producing satisfactory and repeatable results.

It is also apparent that the results when using hyperparameter optimisation and SMOTE techniques are a lot higher than the initial classification results presented in section 4.2 with each classifier outperforming the corresponding classification results from earlier tests. SMOTE was applied to the training dataset to introduce additional records in the minority class, the sampling strategy was used to make sure there was an equal amount of both target classes.

Table 4.3: Results to show AUROC for mortality prediction making use of Grid Search Hyperparameter optimisation

Classifier	**AUROC**
Gaussian Process	0.77 (0.40)
Decision Tree	**0.92** (0.08)
Adaboost	0.91 (0.03)

Table 4.4: Results to show AUROC for mortality prediction making use of Random Search Hyperparameter optimisation

Classifier	**AUROC**
Gaussian Process	0.87 (0.32)
Decision Tree	**0.93** (0.06)
Adaboost	0.89 (0.05)

Random grid search hyperparameter optimisation, with the decision tree algorithm proved to be the most successful algorithm. The final parameter set is shown in table 4.5

Parameter	**Tuned Value**
n_estimator	200
min_samples_leaf	4
max_features	auto
max_depth	10
min_sample_splits	2

Table 4.5: Random Search Optimised Hyperparameters for Decision Tree Classifier

4.4.4 Final Performance Assessment

To further improve the discriminative capabilities of the trained model, synthetic data was introduced to the minority class to rebalance the data frame. This synthetic data made use of the SMOTE technique to rebalance the dataset ensuring there are equal numbers of both classes. Figure 4.13 shows a ROC curve for the Decision Tree classifier. Used to demonstrate the trade off between sensitivity and specificity of ML classifiers across different classification thresholds, ROC curves are commonly used in the medical domain to determine the overall discrimination of a trained classifier. The AUROC of the the decision tree model is labelled on Figure 4.13 and is also shown in Table 4.6.

The dashed red line in Figure 4.13 is defined as the baseline and is often known as the "worst case" for a ROC curve. The blue line indicates the average ROC curve across the repeated 10-fold cross validation, using random search hyperparameters optimisation and SMOTE oversampling techniques.

Table 4.6 shows the results obtained from repeating the final tests multiple times. The maximum AUROC was 0.95, with a mean AUROC of 0.93. The standard deviation of the AUROC over all runs 0.02 showed that there was little variance in the results. Figure 4.14 is the ROC and AUROC score generated for the APACHE II score on the same dataset. The final optimised algorithm has produced better results than that and has improved on the AUROC score for SAPS II introduced in section 4.2 however this is not using the same set of features or dataset so further work would be required to fairly compare the two scrobing systems.

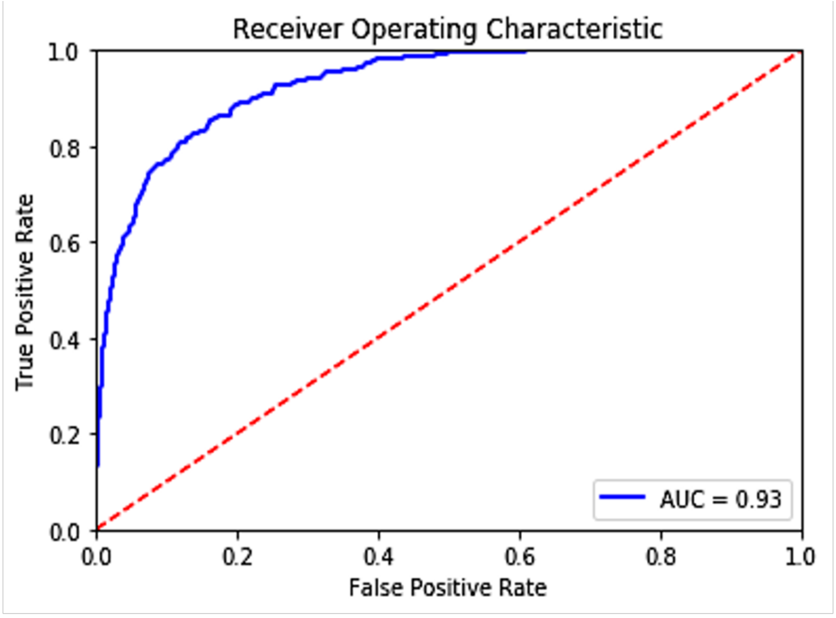


Figure 4.13: Receiver Operating Characteristic Curve for final decision tree model

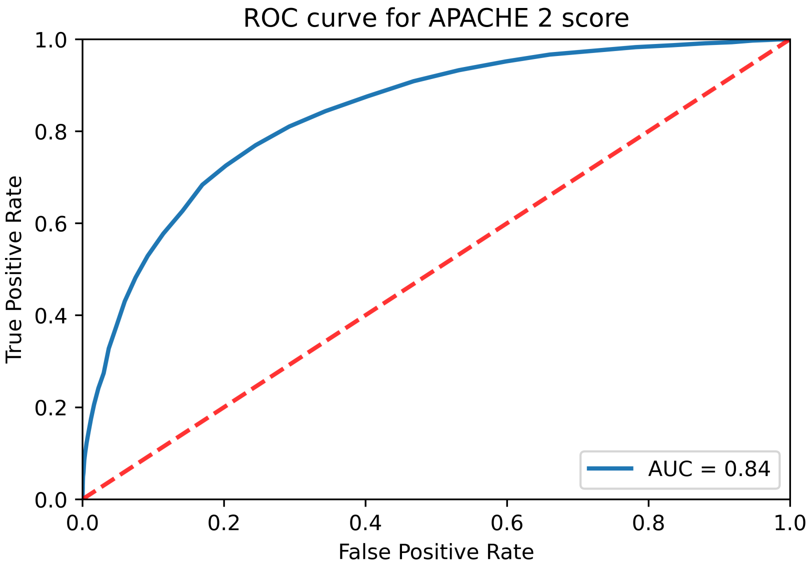


Figure 4.14: Results to show ROC curve for determining mortality using the APACHE-II score

Table 4.6: Results to show AUROC scores for mortality prediction making use of Random Search Hyperparameter optimisation and SMOTE

Metric	**Result**
Average AUROC	0.93
Standard Deviation AUROC	0.02
Maximum AUROC	0.95
Minimum AUROC	0.89

Table 4.7: Results for mortality prediction utilising decision tree classifier with Random search optimisation on unseen data

Metric	**Result**
Precision	0.91
Recall	0.93
Accuracy	0.92

The results in Table 4.7 showed that the addition of synthetic data increased overall accuracy by 1%. The resulting model performs well on unseen data and does not overfit on the original dataset, even with the addition of synthetic data.

4.5 Online Learning

4.5.1 Introduction

As discussed in section 4.2 a single mortality prediction model may start to perform poorly as patient cohorts and treatments change. A more effective solution might be to adapt the predictive model at a local level to deal with evolving population demographics and available medical treatments. To this point, an efficient solution would be that each hospital has their own trained model that can be maintained and updated in real time.

Online learning (also known as real-time ML) is the process of training a model in real time. As new data enters the system the trained parameters of the ML algorithm are updated to try and find the best result for a predefined metric. This section examines the effects of applying online learning to the ICNARC dataset using the pre-trained ML algorithm and parameters from section 4.4.4.

Event driven architectures are common ways of deploying ML models in a production setting. The continuous flow of data through a data stream is given to the model, and the model training pipeline will handle any data issues, transformations, or enrichment's to ensure that the data is consistent and ready to be utilised to retrain the model.

4.5.2 Mapping the System

Production ready online learning models require a detailed ML data pipeline that deals with the dataset at varying stages of the model training process.

Figure 4.15 shows a typical data pipeline for online learning. Due to the extensive model development introduced earlier in this Chapter, for this research I have focused on the resulting output, as the data fed in to the model over time changes.

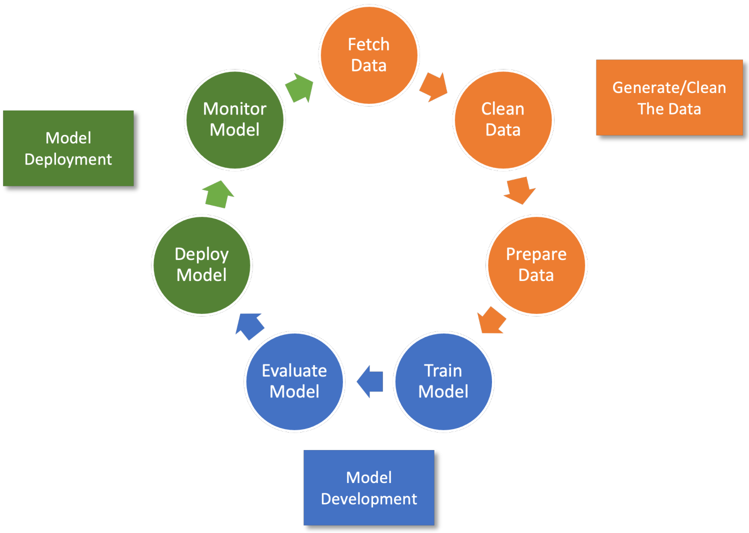


Figure 4.15: An example of an online machine learning data pipeline

Furthermore, many online models use all of the data provided, however, as the available treatments and medicines are changing rapidly within hospitals, the last three months of the data are used in the model at each stage. The dataset is split into different patient cohorts based on a feature, each different cohort is used to train the model with a final split being used to test the results.

4.5.3 Experimental Setup

To demonstrate the hypothesis that using different patient cohorts will improve the overall performance of a trained classifier a selection of tests were undertaken. To demonstrate that as medicines and treatments improve over time the model is constantly improving, the dataset was split based on different features: date of admission to ICU (training on the previous month, testing on the current) and the age of the patient (testing on the old and training on the young).

Experimental Setup: Date Of Admission

To demonstrate that retraining models, in real-time, can be used as an effective way to develop mortality risk prediction tools that can handle new patient cohorts. Two methods of splitting the dataset were used. One such method was training on a previous month of data and testing on the next, with the test data becoming the training data in the following month. Figure 4.16 presents this concept and how it would be applied across the year. The "date of submission" into the ICU is used to determine the data.

The model used to test this hypothesis is the optimised decision tree classifier introduced in section 4.4.4. Each run was undertaken 10 times with the AUROC and accuracy scores presented for each month.

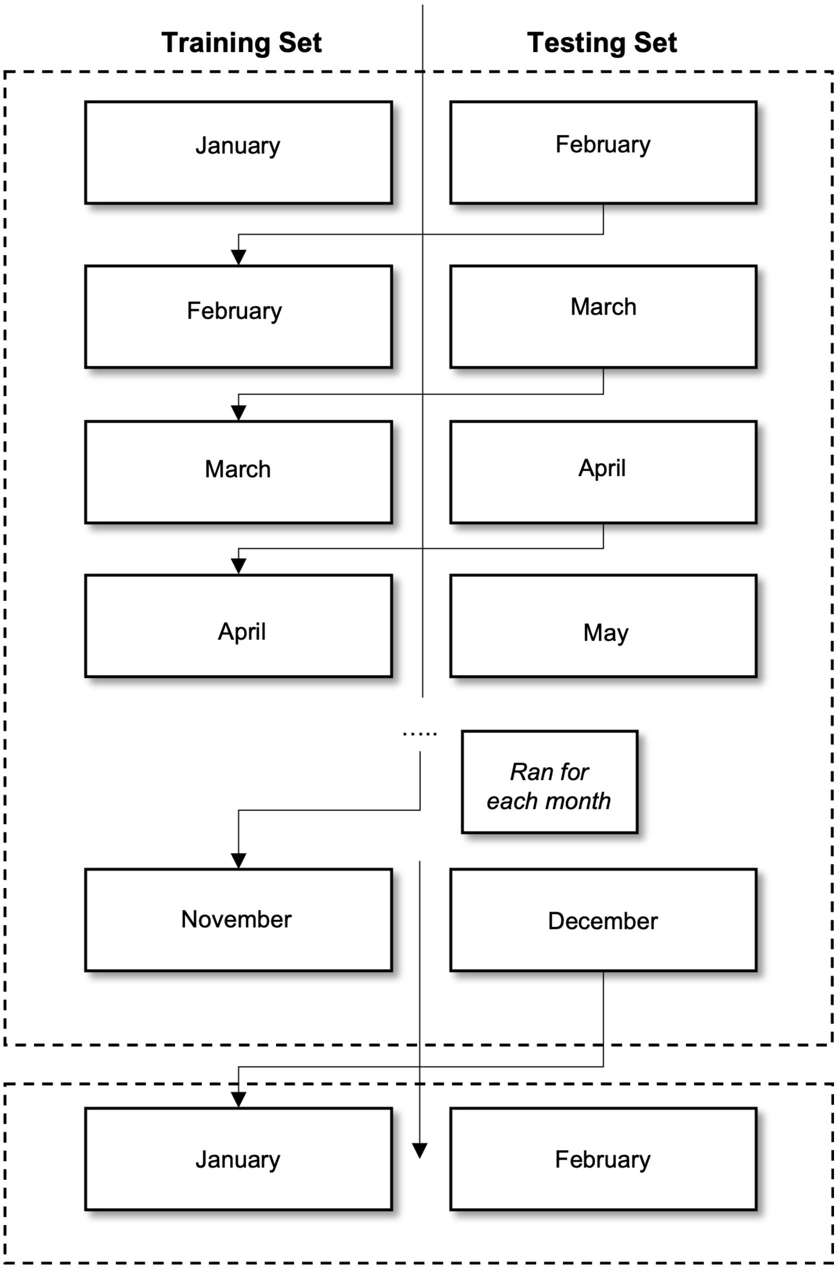


Figure 4.16: Experimental setup for training and testing using online learning with retraining triggered at the start of the month.

Experimental Setup: Age

Another alternative approach to using the date of submission into ICU as the feature to split up the dataset, is using different age cohorts. The dataset contains records of varying ages from 10 to 103. To prove the hypothesis that training the young and testing the old can be a practical approach to keeping mortality systems up to date, increments of 5 years will be considered. Training for the first round of experiments is undertaken on those under 40, and those older than 40 become the test set. 40 is selected as the cut off age as the number of patients within the dataset under the age of 40 is only 2119 (15% of the dataset) - younger ages were considered but there is only 1 patient under the age of 10 and 226 under the age of 20 (1.6%). For the second iteration, all the patients under 45 are considered, and those over 45 are used in the testing set. The AUROC and accuracy are recorded at each increment to monitor how it changes over each experiment. The same decision tree classifier is used to ensure the results are comparable to those produced in sections 4.4.4 and 4.6.1. Figure 4.17 demonstrates how the training and testing data changes over time.

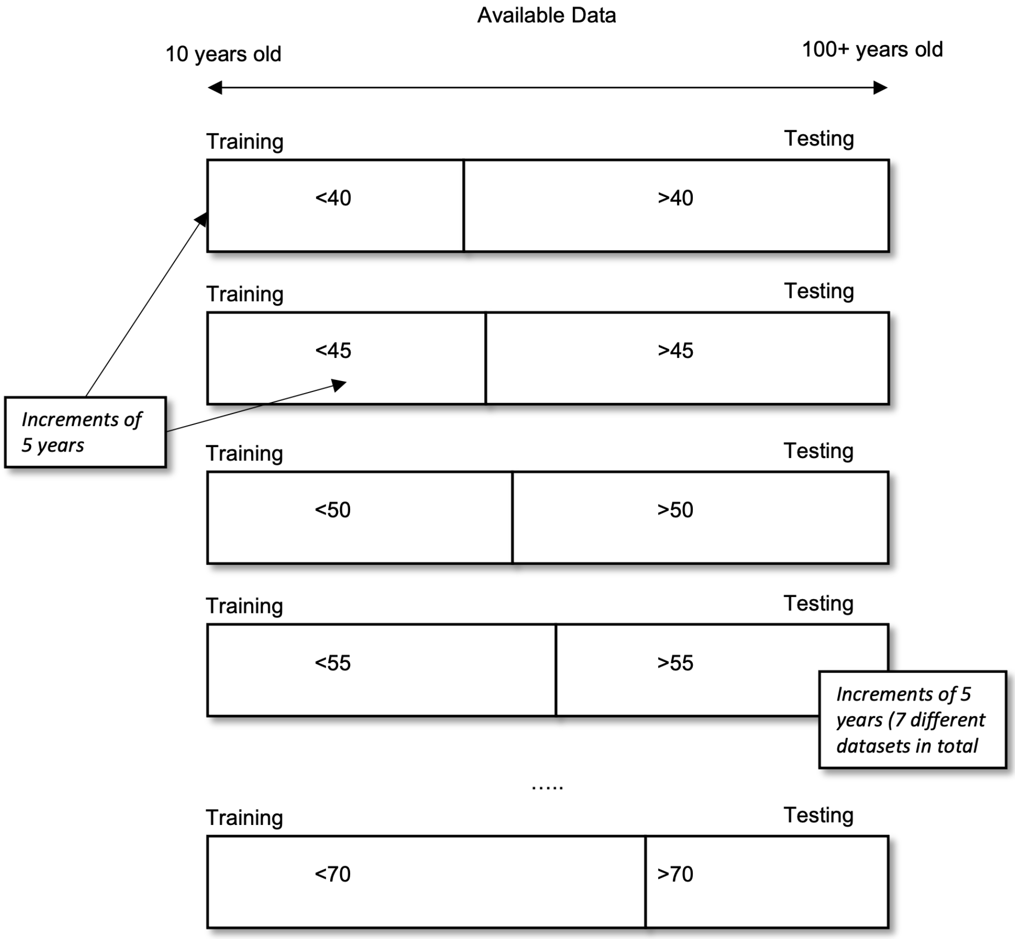


Figure 4.17: Experimental setup for training and testing using online learning with retraining triggered based on age.

4.6 Experimental Results

4.6.1 Experimental Results: Date of Admission

In total 11 different training and testing scenarios were considered throughout the experiments. No test results were provided for January as this was the first month used as training data. Accuracy and AUROC scores are provided from February to December. Each test was run ten times to remove as much variance as possible. Figure 4.18 and Figure 4.19 present the results for accuracy and AUROC results for each of the ten runs.

When considering accuracy, the months of November and December provided the worst results with average accuracy of 0.838 and 0.821 respectively. These results indicate that there may be new illnesses, or ailments in these months that are not generally seen throughout the rest of the year, making it hard to discriminate between those who will survive and those who will die. The trend over the year is consistent, with the accuracy scores not fluctuating much between 0.82 and 0.88. Additional data would be required to consider if this is consistent each year. The trend at the end of the month is an increase in accuracy; with additional data, it is easier to distinguish if this is continuing in previous or future years.

July saw 1068 records tested by the decision tree algorithm. It provided the highest accuracy with an average over the ten runs of 0.88. The number of people admitted to ICU did fluctuate through the training and testing, with 1226 patients admitted into the ICU in December. The accuracy between the spring months of April and May combined with the summer months of June and July showed that the accuracy scores plateaued. However, May provided the highest AUROC score at 0.732. This high AUROC score demonstrates that in the month of May, the model can be considered a good classifier and discriminate well.

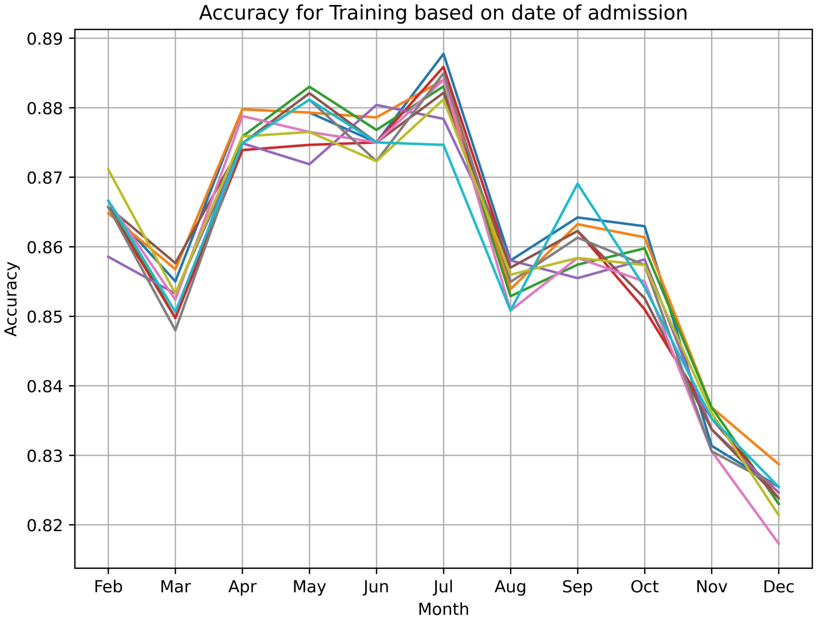


Figure 4.18: The accuracy scores for each run when splitting the dataset based on date of submission.

November AUROC was the lowest with a score of 0.63. This indicates that the model is only marginally better than the worst-case scenario. Perhaps maintaining the decision tree classifier and the hyperparameters is not an effective way of retraining a model each month with just a finite amount of data. Further to this point, in November of the 1269 patients admitted to ICU only 163 died. This heavily imbalanced dataset makes it difficult for the model to determine a patient's outcome accurately.

The results for the AUROC appear to run in three-month cycles. Typically AUROC scores peak after three months and then drop down again. This could be linked to seasonality. Runs for the months of March, April and May (Spring) where the AUROC peaks at the end of the season. Similarly, for the runs which are for the months of June, July and August; the AUROC peaks at the end of the cycle. This cycle indicates that there may be some seasonality associated with how successful a mortality prediction tool is when trained each month.

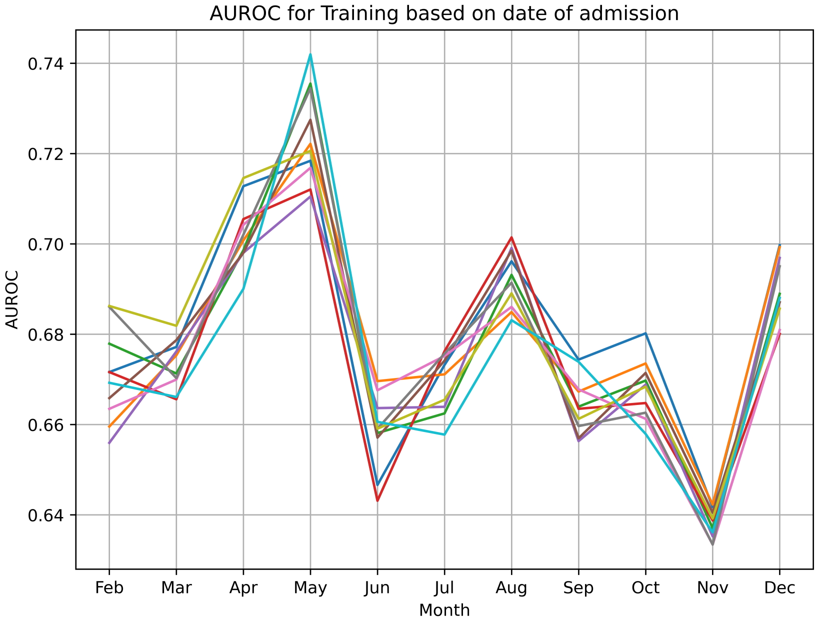


Figure 4.19: The AUROC scores for each run when splitting the dataset based on date of submission

Further tests could be undertaken on a more extended dataset that can demonstrate the results over a longer period than one year. Three-month periods could be considered for the training data and then tested on the current month. This would account for the seasonality changes presented in Figure 4.20.

The combination of the previous year's season and last month's data may provide the best results as it allows for changes in medicine, treatments and operations not previously used; but also provides information on any of the ailments or reasons for submission that might not be seen in the previous months/season.

4.6.2 Experimental Results: Age

The first round of experiments was split into seven training and testing scenarios when using age as the trigger to split datasets. Starting at 40 the dataset was split, training on those less than 40 and testing on those older. For each experiment, 5 years were added until 70 years of age. Figure 4.21 and Figure 4.20 present the average results for accuracy and AUROC results for each of the ten runs.

Unlike in section 4.6.1 the trend for AUROC is consistently rising throughout the seven different dataset combinations. The maximum AUROC is 0.82 with the lowest achieved at the start of 0.67. The best AUROC score, as expected, was achieved when training took place on those less than 70 and testing on the rest. The AUROC scores plateaued for experiments of ages ¡50 to ¡60 with an average of 0.78 AUROC score. These runs are for patients with cut-off ages varying between 50 and 60 years of age. This may indicate that for these ages, the information within the dataset remains relatively consistent with many patients experiencing the same issues.

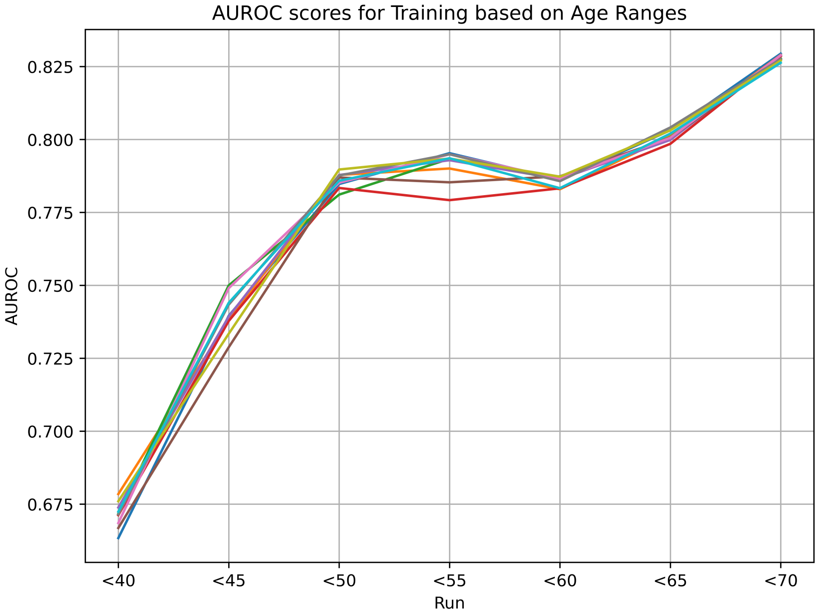


Figure 4.20: The average AUROC score when splitting the dataset based on age.

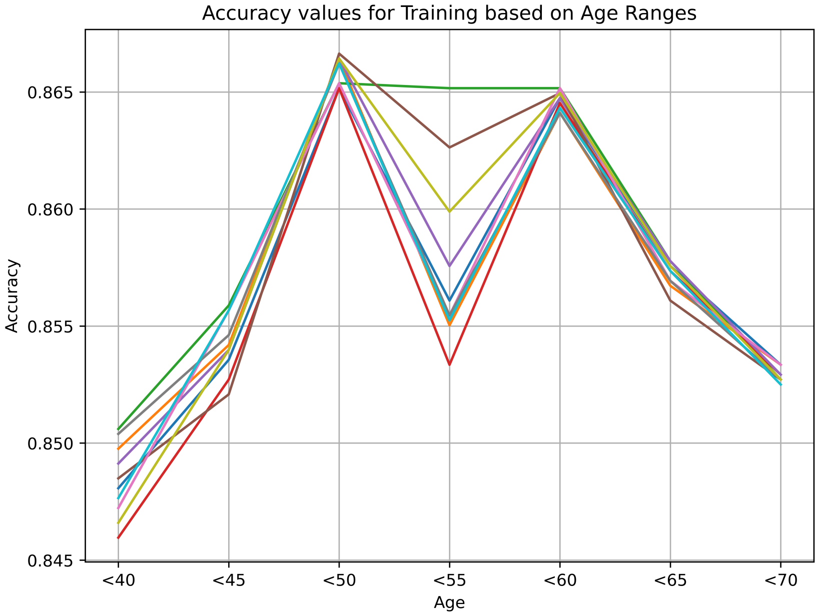


Figure 4.21: The average accuracy score when splitting the dataset based on age.

Figure 4.21 presents the accuracy over each different experiments. The maximum accuracy was over 86.5% when testing patients over 50 years of age. The accuracy did drop when testing with patients 55 and over but rose again to a second peak of 86.3%. The accuracy doesn't change dramatically throughout the experiments, as it only varies between 84.7% and 86.5% for all of the different patient cohorts.

To better understand the results and to see if age can be used as a good discriminator, further tests were undertaken, however, the age was not split by 5 years but rather by 2. Figure 4.23 is the average accuracy over the 25 different scenarios. Testing split on patients less than 64 years old showed the lowest average accuracy of 84%, but the overall trend continues to rise. Similar results were demonstrated on the AUROC scores with the overall score dropping. However, it is not the lowest AUROC achieved as that is still the first experiment and with the smallest training set when splitting on patients aged 40.

It is clear that more data shows it is possible to generate better overall results in accuracy and AUROC, the more patients included in the training, the larger the number of illnesses and issues are seen. This demonstrates that the work undertaken to date should be treated with care unless updated often using new data.

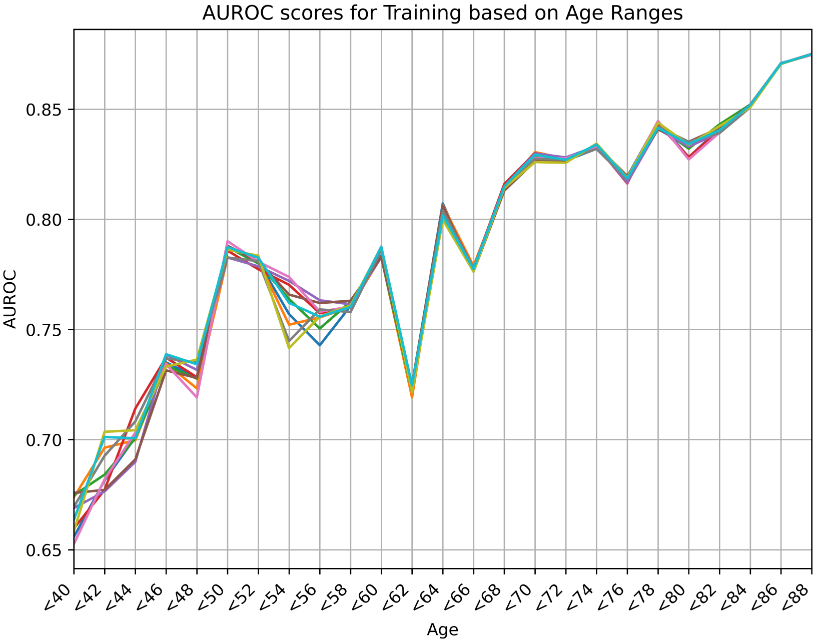


Figure 4.22: The average accuracy score generated for online learning when splitting the dataset 25 times.

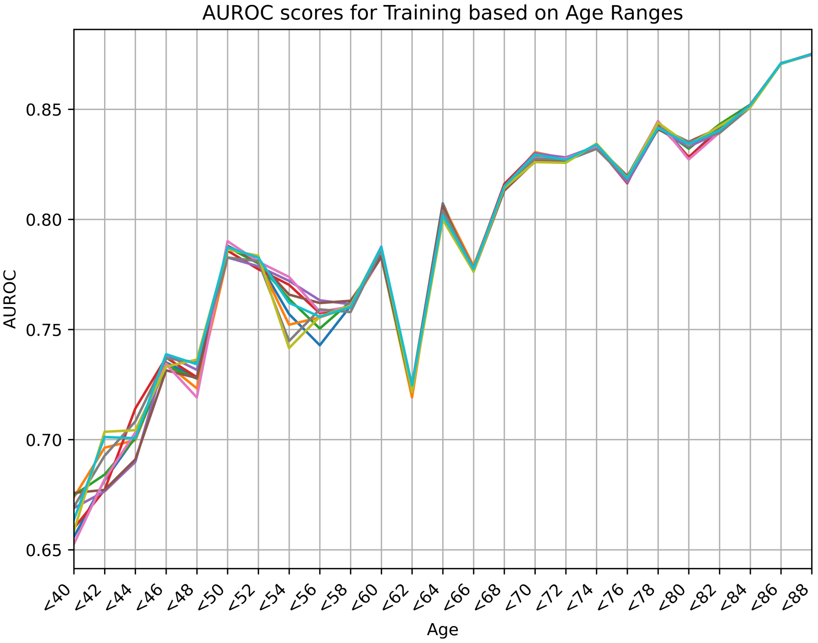


Figure 4.23: The average AUROC score generated for online learning when splitting the dataset 25 times.

4.7 Discussion

The results in this Chapter demonstrate that it is possible to apply ML techniques to mortality prediction cases within ICU. It has been shown that using simple ML algorithms with optimisation techniques for re-balancing the dataset and tuning hyper parameters can correctly identify 92% of patients admitted into the ICU who had their patient data recorded and distributed as part of the ICNARC dataset. Some limitations of this work is the finite amount of data made available to train these models. Theoretically, they have produced good results but would need to be reviewed over time to ensure that they are reliable for different patient cohorts.

The data suggested that using a greedy search based algorithm such as Decision Trees or Random Forests produced the most promising results. Further work could be undertaken to investigate this more by using the results from an ensemble of classifiers. The results produced by Neural Networks would also require additional research. Neural Networks require large datasets to correctly identify and train the weightings within the network.

The results challenge the existing support tools used in ICU. The results for the optimised decision tree classifier produced a higher AUROC score than APACHE II, 93% where as APACHE II only achieved an AUROC of 83% for the given dataset.

The novel approaches of retraining the model for different patient cohorts based on age/date of submission, demonstrate results that are comparable for most windows with those in state-of-the-art research. As procedures and medicines change it is important that support tools account for these changes.

Existing support tools such as SAPS II do not do this and have become quickly outdated. The approach introduced is quick and can be applied to any different patient dataset (with the same feature set).

There are some limitations in the work that needs to be considered. One of the key limitations of this work are the finite number of records available within the completed ICNARC dataset. A true representative sample with equal age and genders splits needs too be used to understand how well the model can discriminate for different age ranges and gender. Further to this point, the data available within the ICNARC dataset does not have information on any complications that have occurred whilst a patient has been in the ICU. Introducing this as a new feature would allow the model to account for those changes and show how a patients chance of survival may or may not deteriorate over time.

The models presented act as a support tool for medical practitioners and allow them to make more informed decisions on patients. The models presented also support medical practitioners in understanding the chances of survival without influencing whether they should intervene on a patient or not.

Some avenues for further research include:

- Investigate the effects of retraining the model for those who have entered the ICU via different wards or different reasons for admission.

- How does the results of the optimised ML algorithm compare to other support tools such as the SOFA score.

- Test different Neural Network approaches on a larger dataset.

Stakeholder Engagement

Throughout the research ongoing conversations took place with many experts within the field of medicine, they found the work fascinating and engaged throughout the work. They felt as though the research built on the existing research in this field of study but believe that there needs to be more attention given to complications that can arise throughout the duration of stay within ICU. Current mortality scoring systems provide a score at the time of admission and don't account for changes throughout the ICU stay.

4.8 Conclusion

The current study has indicated that ML can be used to accurately predict mortality in intensive care units. In this Chapter, the ICNARC dataset was used, which contains 29 features acquired within the first 24 hours of admission in to ICU. The most effective models made use of the Decision Tree algorithm and achieved a classification accuracy of 92%, a PRE of 91% and a REC of 93%. These results have indicated that the developed ML model is more effective than current state of the art techniques at predicting mortality on the patient cohort considered in this study.

Medical practitioners often use outdated methods to quantify mortality risks to patients in intensive care units. The existing methods do not account for changes in medicine, patients reactions to intervention, and can not be calculated at admission to the ICU. Many risk mortality scores exist and are specialised to different diseases or populations. This Chapter has presented the development of a ML pipeline that can be used to correctly identify and quantify the risk of mortality for patients admitted in to intensive care units. The methodology used to develop the pipeline can be applied to many real-world classification problems. The mortality risk prediction tool developed in this chapter uses 29 features to determine the score (unlike the previous work described in section 4.2). Section 4.5 introduces the techniques of online learning and how it can be used for changing cohorts and developments in medical treatments. The results show that it is possible to train the model at different intervals for varying cohorts to improve the model accuracy.

It is expected that further research and development into ML for mortality risk prediction will result in an online ML support tool that can be trained continuously with new data. This would make it suitable for application across many different patient cohorts, and allow the adaption of the model and methodology to be applied globally, as patient demographics evolve over time.

Chapter 5

Using Ensemble techniques for Data Imputation

5.1 Introduction

In statistical analysis and ML, missing data in datasets is a common occurrence and can have a significant impact on the validity of any conclusions that can be inferred from that data. This includes reducing the confidence levels of any statistical or ML model developed using that dataset, and thus its usefulness. There are a variety of reasons that missing data occurs including: a non-response of a variable in an observation, faulty equipment, or improper data collection. This means most statisticians and data science researchers have to deal with missing data at some point in their analyses. In many environmental science datasets, missing observations are a frequent occurrence because of instrument failure or data collection quality control procedures (Sabay et al., 2018).

An example of reported missing data is the US centers for disease control morbidity and mortality report published, which states 12,928,749 people had received at least one dose of the COVID-19 vaccination throughout the US in February 2021. It was noted that around 48.13% of those who had received the vaccine had not provided their race or ethnicity despite being explicitly told to provide this. This information is essential to ensure vaccine roll-out standards are met and to maintain a consistent roll-out across all ethnic groups (Painter, 2021).

Missing data causes issues with the feasibility of creating statistical and ML models, with most multivariate analysis algorithms expecting complete data (i.e. data for all observed variables) for each observation (Sabay et al., 2018). Complete case deletion (where the observation with missing data values is removed from the dataset) is commonly used to address issues as it is the simplest and most intuitive way of dealing with missing data (Afifi and Elashoff, 1966). The statistical model can then be created using the subset of complete observations. However, as the number of cases with missing data increases, the resulting impact of deleting cases becomes more severe due to the fact that statistical models are only really reliable and effective if they can make inferences about the entire population and not just the cases with complete data. These deleted observations may also contain vital information affecting the quality of the model output (Khan et al., 2018), so deleting an entire observation because some of its values are missing is often undesirable.

Prior to removing data, one of the most important decisions that will be under-taken by the researcher is deciding whether there is any pattern, in the missing data. Frequently understanding the way the data is missing helps to understand the reason why it is missing and therefore how best to address the missing data. For example, if data is missing at random then the remaining data sample (after cases have been removed) is likely to still be representative of the population. However, if data is missing according to a systematic pattern then excluding cases with missing data from the modelling process will bias the results of any model created from that dataset. In this case, methods to deal with this missing data without excluding the entire case are needed.

In this Chapter a novel approach to dealing with missing data by using ensemble ML methods is proposed, and this method is applied to the binary classification problem of detecting heart disease. The key contributions of this Chapter are as follows:

1. A novel ML framework for missing data imputation is proposed.

2. Results for the base classifier are compared with state-of-the art ML models described in literature.

3. The ensemble imputation methodology is applied to both data that is missing at random and data that is systematically missing.

4. Results of the EIM are compared with state-of-the-art ML approaches.

The rest of the Chapter is organised as follows: section 5.2 considers related work in identifying patterns in missing data and the application of ensemble methods in ML problems, section 5.3 describes the proposed novel approach to ensemble multiple imputation methods. Section 5.3 will then describe the experimental methodology in detail: including the dataset to be used in the experiments, the process for removing data from the dataset, and the baseline classifier to be used in the experiments. Section 5.4 shows the results of the proposed ensemble method for data imputation and compares those results both to a strong baseline method and to state-of-the-art results reported in the literature. Finally, section 5.6 will present conclusions and some ideas for further work.

5.2 Background and Related Work

5.2.1 Missing Data Patterns

Before deciding on the most appropriate way of dealing with the missing values in a dataset the type of *"missingness"* and features should be considered. Data that is missing can be uni-variate, monotonic, or in arbitrary patterns. It is also important to understand the mechanisms that have led to there being missing data. Little and Rubin (2002) introduced three different types of missing data categories that are formed in relation to randomness:

- Missing Completely At Random (MCAR)
- Missing Not At Random (MNAR)
- Missing At Random (MAR)

Missing data can be described as MCAR if the probability of data missing is equal for all the different cases, implying that the cause of the missing data is unrelated to the data itself. As a result, it is possible to ignore many of the difficulties that commonly arise with other types of missing data, besides the loss of information. An example of MCAR data is that of missing blood oxygen recordings in a clinical record due to sensor malfunction. This means each sample has the same chance of missing that data. Data that is MAR occurs when the probability of the data being missing is accounted for by other measured variables. For example, men are often less likely to respond to questions about mental health in surveys. Modern missing data inference methods generally start with the assumption that the missing data is MAR. If the missing data is related to data that is not present within the current dataset then the data is MNAR. It is difficult to impute missing values of this type, as none of the data that is currently available in the dataset is related to the data that is missing. For example, if a public survey was undertaken asking for people's opinions and those with the weakest opinions responded less often, then it would be referred to as MNAR data. MNAR data is the most complex to work with. Strategies to deal with MNAR data include performing 'what-if' analysis in order to understand how sensitive the results are to different scenarios (Hughes et al., 2019; Penny and Atkinson, 2012; Wells et al., 2013).

5.2.2 Ensemble Methods

Ensemble learning, also known as meta-learning, is the process of combining the predictions of multiple classifiers to solve a particular classification or function approximation problem. It is often successful at improving the overall performance of the model. Ensemble methods were first introduced as early as 1979 in the research undertaken by Dasarathy and Sheela (1979). They proposed using an ensemble system in a divide and conquer setup where the multiple classifiers were used to partition the feature space. Ensemble methods have two main benefits, variance reduction and bias reduction (Smolyakov, 2019). The two most common forms of ensemble methods are bagging (using sub-samples of the training dataset to develop different models) and boosting (where the model attempts to create a strong classifier by generating a model and then attempting to solve errors for each iteration by generating a new model). Simple ensemble methods often construct a set of base classifiers and then classify new data points by taking a vote on the predictions that are made. The learning procedure for these algorithms can be divided into three separate parts:

1. **Data Pre-processing:** Data pre-processing is the initial stage of solving any classification problem. The data must be properly formatted to allow the training of the base classifiers to take place. This is done by editing and adapting the original training data to suit. Reshaping the data, normalisation techniques such as PCA (see section 2.8.2), and conversion of categorical variables using one-hot encoding are examples of data pre-processing techniques that are commonly used in the literature.

2. **Constructing base models:** This is the creation of the base classifier with a specified learning algorithm as the base learner. In this Chapter, the base classifier used is Logistic Regression (see section 2.7.3 for a detailed overview), but the proposed techniques can be generalised to any different classification technique.

3. **Voting:** The final stage of developing an ensembled pipeline is to combine the predictions of the base models that were developed in the previous step into a final output prediction. There are several types of voting systems that can be utilised:

    - *Averaging*: Multiple predictions are made for each different data point with an average of the predictions taken from all of the models. Averaging can be used when making predictions in regression problems or while calculating probabilities for classification problems.
    - *Weighted Average*: Similarly to averaging, using a weighted average takes an average of all of the predictions. However, before averaging, each model is assigned a different weight that defines the overall importance of each model for prediction.
    - *Max Voting*: Generally used for classification problems, max voting is used when each model makes a prediction of what it thinks is the resulting class. Each prediction is a "vote", and the prediction with the most votes is used as the final predicted value (Tran et al., 2016).

5.2.3 Imputation Methods

One of the key methodologies used by researchers to populate and fill missing data is to utilise imputation. Imputated data is typically used in place of records where there is missing data (often known as incomplete data). By using various statistical calculations a suitable value can be found for what the missing data could be. The methods used for imputation can have a drastic impact on the overall reliability of a ML model. Where possible ML models should be trained with a complete dataset with no need for imputation.

There are three main categories that can be used for imputation:

1. **Listwise Deletion:** Listwise deletion is used when records are deleted from a dataset if they are missing data on any of the variables present within that data record. Although simple to apply and commonly seen in literature, this is only sometimes the most efficient way of dealing with missing data. When data is MCAR, listwise deletion should be used as there is no difference from the complete cases within the dataset. If a selection of results is missing due to the failure of a medical instrument (or the data has not been collected for an undetermined amount of time) the dataset could be biased therefore listwise deletion shouldn't be used. With any imputation or deletion methodology, caveats should be presented for transparency and imputation/deletion methods should be carefully selected depending on the sample size of the dataset and how much could be added or removed. Listwise deletion can be used when the sample size is overly large, and the number of cases for deletion is minimal (Grace-Martin, 2014; Glas, 2010).

2. **Single Imputation:** These procedures allow a single value to fill in the missing data element within a dataset by a defined single rule. Mean imputations is a commonly used method where the mean value is imputed in place of missing elements (Jamshidian and Mata, 2007; Glas, 2010; Jakobsen et al., 2007). The following methods are examples

of single imputation:

- *Hot Deck Imputation*: Missing values are replaced with values observed earlier from other respondents with a *"similar"* pattern in their other features. Despite being commonly used in practice, the theory determining what data is similar is not as transparent as other more straightforward imputation methods. Andridge and Little (2010) reviewed the literature to explain how the data should be selected.
- *Cold Deck Imputation*: A different approach for imputation is cold-deck imputation; this utilises results from a source unrelated to a dataset being considered. An example of cold-deck imputation is taking responses from an old patient's questionnaire to fill in the missing answers within a current questionnaire. There is a requirement to have expertise in the field to select the correct records.
- *Substitution*: Substitution imputes data for missing records by taking the results provided by others not initially included in the sample. Consider a dataset containing a selection of recorded features of women who have both had a miscarriage or have not had a miscarriage from 2019-2021. Any missing data could be filled by looking at patients from 2022 with no missing data where the outcome is the same in relation to miscarrying.

3. **Multiple Imputation:** Multiple imputation procedures were developed to combat the uncertainty and bias introduced into datasets by

using single imputation methods or listwise deletion. Several plausible imputed datasets are generated, and their results are combined to find the most relevant result.

Multiple copies of the dataset are synthesised with all the missing data replaced by imputed values. These values are samples from their predictive distribution based on existing values within the dataset. To account for the uncertainty within the missing data, the imputation procedure must use appropriate variability in the imputed values (Rubin, 2004; Sterne et al., 2009).

5.3 Materials & Methods

5.3.1 System Design

To test the hypothesis that using combinations of different single imputation methods in an ensemble ML pipeline improves overall classification performance in the presence of missing data, the experimental framework shown in Figure 5.1 is proposed. This framework consists of the following steps:

1. Split the complete data set into training, test and unseen datasets.

2. Use the training/test set to develop and train an optimised base classifier.

3. Remove some of the data from the unseen data set.

4. Use multiple different imputation methods to create multiple unseen testing data sets.

5. Make predictions from those multiple testing sets using the base classifier.

6. Combine those predictions into a final output using ensemble voting methods.

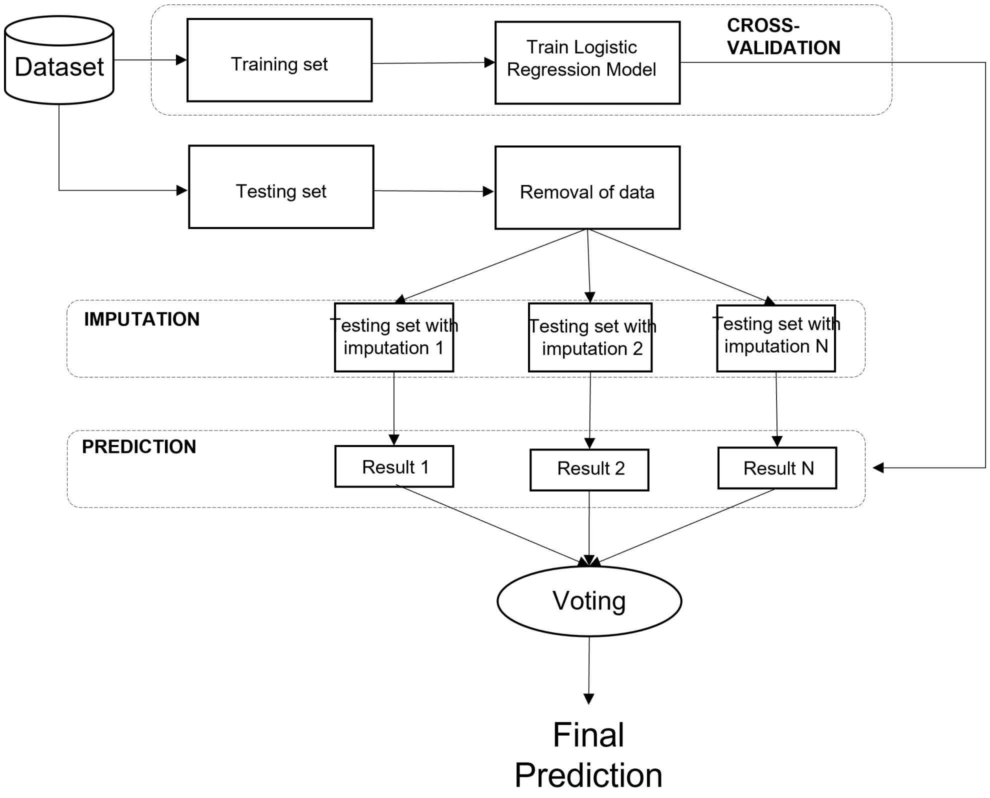


Figure 5.1: Experimental set up for ensemble-based Multiple Imputation pipeline

The first step in this experimental framework is to split the complete data set 70:20:10 into a training set, testing set and unseen dataset using a stratified approach based on target classes (to ensure that the held out test set is representative of the entire data set (Kumar, 2022)). The training set is then used to develop and train the base classifier using a stratified 10-fold cross-validation strategy to tune the model hyperparameters (again, as with the splitting of the data into training and testing sets, stratification is necessary to ensure that each fold is representative of the balance of the full dataset). Once the optimal model parameters were selected via this cross-validation process, the base classifier was then retrained on the complete training dataset using those parameters, before being used for inference.

In this Chapter, a Logistic Regression classifier (see section 2.7.3) is used as the base classifier primarily due to its simplicity and wide-spread usage. The cross-validation process is used to select the penalty function, regularisation strength, solver, and solver parameters to use. However, the approach proposed in this Chapter is generally applicable to any classification algorithm (e.g. ANNs, SVMs, Random Forests, etc.).

Once the base classifier is trained, a proportion of the data points are then removed from the unseen test set to mimic real-world missing data. In this Chapter, two missing data edge cases are considered: data that is MNAR, and data that is MCAR (see section 5.4).

5.3.2 Simulating Missing Data

As discussed in section 5.2, data that is MNAR represents significant difficulties for the development and application of predictive models – with the systematic nature of the *"missingness"* often resulting in a biased estimate of effect (Dong and Peng, 2013). In this Chapter the effect of MNAR data across dataset features of varying feature importance is evaluated, to see how the proposed approach performs in a range of MNAR data situations.

Firstly, to remove data in a systematic approach we use different combinations of up to three features with 10% of the samples in the data set removed.

In contrast to MNAR data, data that is MCAR typically does not introduce bias into a predictive model. This means that, although MCAR data reduces the number of samples we can analyse, the remaining data is representative of the full data set of interest. To evaluate the ensemble imputation approach proposed in this Chapter against MCAR data, data is simply randomly removed from 10% of the samples across all features (rather than systematically across specific features as done to simulate MNAR data). The total number of data points that is removed remains the same in both cases.

5.3.3 Imputing Missing Data and Ensembling Predictive Models

The data removed in step 3 (as described in section 5.3), is then imputed using a range of different simple imputation methods to create multiple testing data sets. Throughout this Chapter the effectiveness of the proposed method is evaluated using zero fill, one fill, mean fill, and mode fill imputation methods (however, as discussed in Chapter 7, other more complex imputation techniques can also be used, and this is an area for further work). The base classifier developed in step 1 is then used to make predictions using each of these testing sets, and the predictions are combined using ensemble voting techniques (see section 5.3) to produce a final prediction. All models have been implemented in the Sci-Kit learn ML framework (Pedregosa et al., 2011).

5.3.4 Cleveland Heart Disease Dataset

26 million adults have been diagnosed with Coronary Heart Disease (CHD) worldwide with an estimated 3.6 million adults being newly diagnosed every year. The cost for heart failure management and care is approximately 1-2% of the global expenditure of healthcare, with most cases being linked to recurrent hospital admissions. The increased widespread presence, increasing numbers of recurrent hospital admissions, and surge in hospital costs has highlighted the need for prompt diagnosis and estimation of severity of heart disease to allow for the most effective treatment to take place (NHS, 2015). Its estimated that in the United Kingdom (UK) alone, 2.5 million people are living with CHD, and CHD is responsible for 73,000 deaths each year (Townsend, 2014). Usually a coronary angiogram is used to accurately diagnose both the presence and severity of heart disease. This procedure is not suitable for large scale screening as it is an expensive and invasive procedure. A possible solution to these drawbacks is to use computational methods of predicting and estimating the presence of CHD.

303 cases of heart disease data from V.A medical centre, Cleveland Clinic foundation, and Long Beach were collected by Detrano et al. (1989), to develop a discriminate function model for estimation probabilities of CHD. The data collected has been used frequently in classification research literature to benchmark base classifiers and pipelines. The majority of studies that used the dataset consider 14 (13 input and 1 target feature) of the 76 problem attributes (which are detailed in Das et al. (2009) and Nguyen et al. (2015)). In this Chapter a subset of the classes were used. The data set categorises the

severity of CHD from 0 (No heart disease present) to 1 (mild heart disease) through 4 (severe heart disease). Although this data has been used extensively in research for classification, most researchers have reshaped the problem by combining groups 1-4 as a single class and distinguishing the presence of heart disease from no heart disease detected (Nguyen et al., 2015). Table 5.1 describes all of the features within the dataset and a description of each feature is also provided as well as the datatype. Figure 5.2 shows the spread of classes in the unaltered dataset. Conversely, Figure 5.3 presents the spread of classes used for this body of work once the classes were reassigned to make it a binary classification problem. This work also helped to rebalance the dataset.

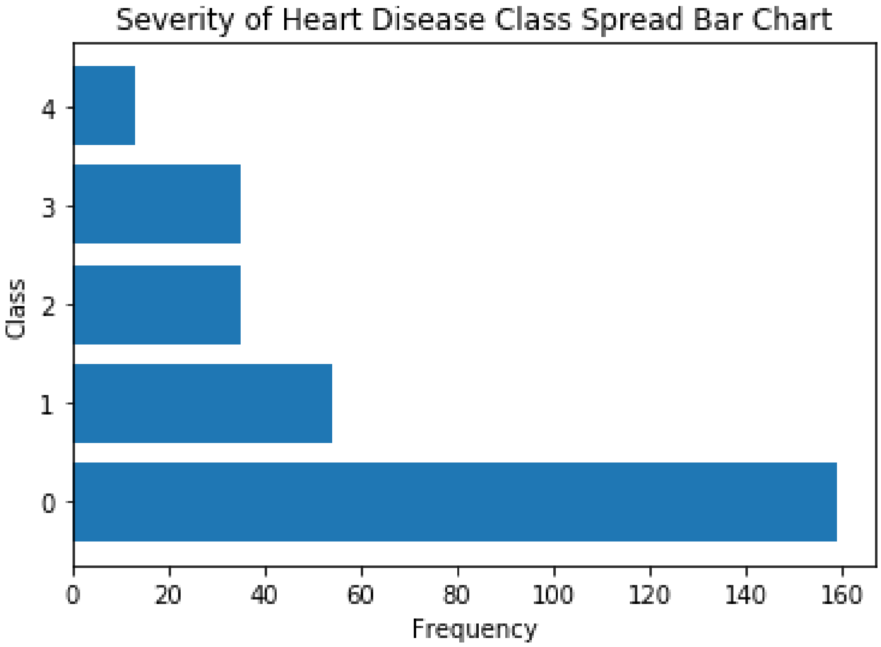


Figure 5.2: Graph to show the original classes spread

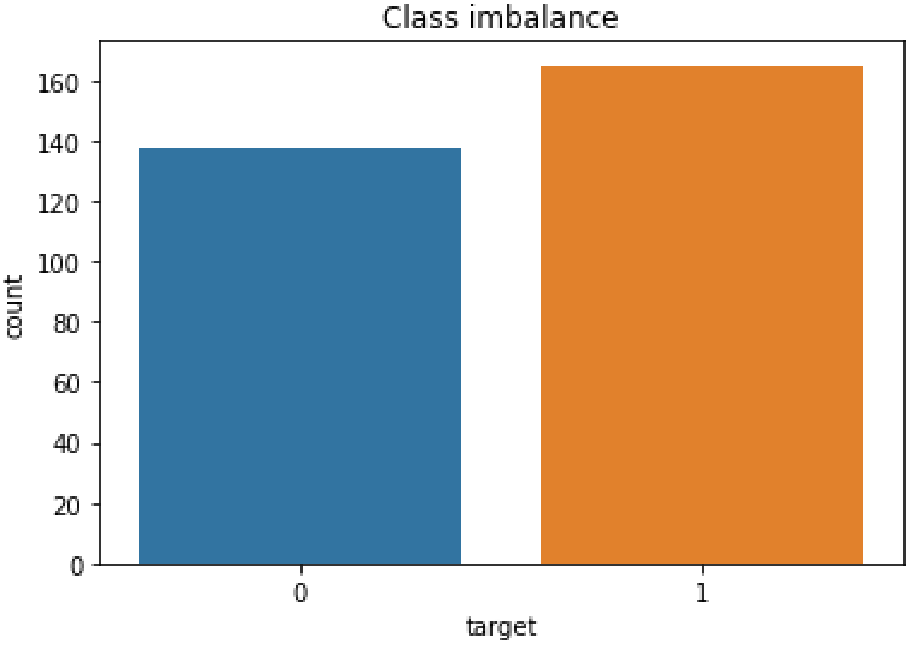


Figure 5.3: Comparison of how the dataset can be considered as a binary or multi-class classification problem

Table 5.1: General description of features present within the Cleveland heart disease dataset. (Detrano et al., 1989)

Feature	Type	Feature Description
age	Integer	Age of subject (25 – 80)
sex	Categorical	Sex of subject (0 = female, 1 = male)
cp	Categorical	Type of chest pain experienced (1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic)
trestbps	Integer	Resting blood pressure in mm Hg on admission to the hospital (94 – 200)
chol	Integer	Serum cholesterol measure in mg/dl (125 – 564)
fbs	Categorical	Fasting blood sugar level greater than 120 mg/dl (0 = false, 1 = true)
restecg	Categorical	Resting electrocardiogram results (0 = normal, 1 = ST-T wave abnormality, 2 = left ventricular hypertrophy)
thalach	Integer	Maximum heart rate achieved in bpm (71 – 202)
exang	Categorical	Exercise induced angina (0 = no, 1 = yes)
oldpeak	Real	ST wave depression induced by exercise relative to rest (0.00 – 62.00)
slope	Categorical	The slope of the peak exercise ST wave segment (1 = upsloping, 2 = flat, 3 = downsloping)
ca	Integer	The number of major vessels coloured by fluoroscopy (0 – 3)
thal	Integer	Thalassemia (3 = normal, 6 = fixed defect, 7 = reversable defect)
target	Categorical	Presence of coronary heart disease (0 = absent, 1 = present)

5.4 Experimental Results

This section introduces the 10-fold cross validation results for training and testing of the heart disease detection binary classification problem. The results from a collection of classification algorithms are presented with a focus on the Logistic Regression classifier. The ML models and EIM techniques introduced in this Chapter were trained on a M1 Pro Macbook Pro with 16Gb RAM. The training time for each classifier was negligible but is provided in 5.2 deonted with the column TT with results provided in seconds.

Baseline results and how they compare with existing literature are disseminated in section 5.4.1. The baseline results are then compared with the results from different testing scenarios in section 5.4.2.

5.4.1 State-of-the-Art Heart Disease Classification

The Logistic Regression classifier outlined in section 2.7.3 as well as some other common ML algorithms were applied to the Cleveland heart disease classification dataset containing those with or without heart disease; table 5.2 shows the Accuracy, REC, PRE, and F1 scores for the repeated 10-fold cross-validated results. Logistic Regression provided the highest overall accuracy of 0.86% and was therefore selected as the classifier that would be used to test the process of imputing different single imputation methods and using ensemble approaches on the results (described in sections 5.2.3 and 5.2.2). The optimised Logistic Regression classifier also produced the best results for REC and F1 score which are 0.90 and 0.88 respectively. The Naïve Bayes has the highest PRE score with a score of 0.87.

Table 5.2: Classifier Comparison results for the Cleveland Heart Disease Dataset

Model	Acc(%)	Recall	Prec	F1	TT(S)
Logistic Regression	0.86	0.90	0.86	0.88	0.263
Random Forest Classifier	0.82	0.86	0.82	0.84	0.041
Decision Tree Classifier	0.71	0.72	0.77	0.74	0.003
SVM - Linear Kernel	0.69	0.73	0.74	0.70	0.003
K Neighbours Classifier	0.69	0.82	0.69	0.74	0.005
Naive Bayes	0.67	0.50	0.87	0.61	0.002
Quadratic Discriminant A	0.59	0.54	0.64	0.56	0.003

Figure 5.4 shows the AUROC curve when using the optimised Logistic Regression algorithm for the unseen validation set; it demonstrates that the model can accurately discriminate between those with heart disease and those without it. The solid green line indicates the results of the ROC curve for those in Class 1 (with heart disease present). Class 0 (the solid blue line) indicates those who do not have heart disease. The dashed black line suggests a classifier which randomly chooses a patient's outcome class – this can be considered a worst-case scenario. Table 5.3 presents the overall results for the ROC curve as well as the maximum and minimum AUROC scores from the test dataset. The standard deviation of the AUROC over 10 runs is also provided.

Table 5.3: AUROC scores for the optimised Logistic Regression model

Metric	Results
Average AUROC	0.92
Standard Deviations AUROC	0.05
Maximum AUROC	1.00
Minimum AUROC	0.87

Several researchers have used different approaches to predict the presence of CHD in a patient from the Cleveland heart disease dataset. Table 5.4 summarises the outcome of these research projects. Many research teams

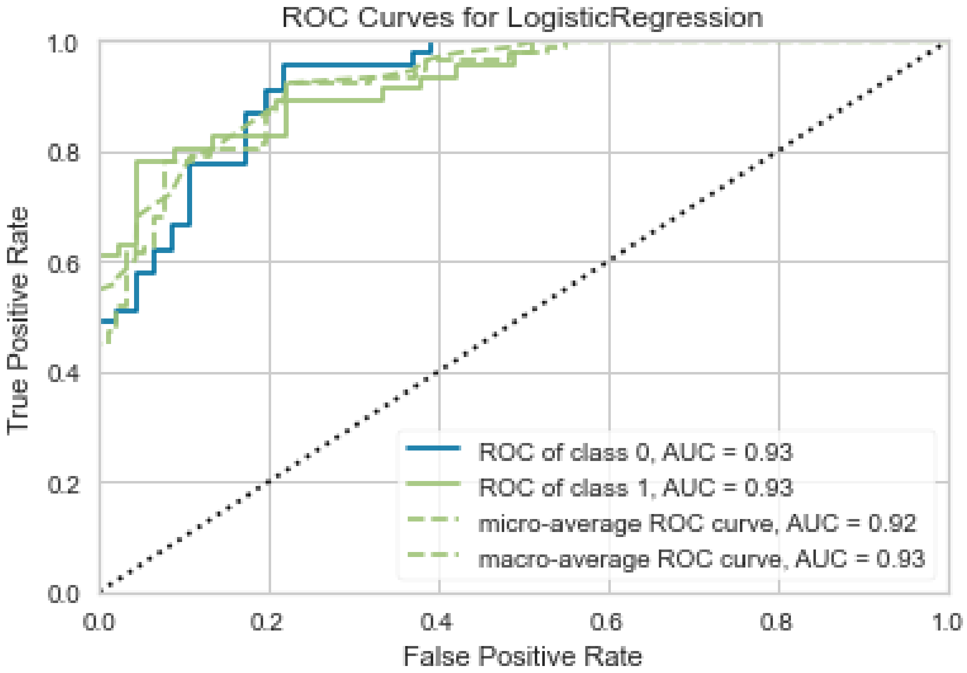


Figure 5.4: Receiving operator curves for baseline Logistic Regression model

have turned the dataset into a binary classification project, and only these results are compared with the results generated in this work. The proposed classifier for use in the EIM tests is a Logistic Regression classifier; the accuracy generated for the optimised model shows that it can determine heart disease at a comparable level to those described in the current state-of-the-art literature.

The baseline model trained on the complete dataset introduced in this section is used throughout the following sections. The unseen test dataset is maintained throughout the experiments, but different data removal methodologies are used with synthetic data imputed.

Table 5.4: Comparison of Accuracy scores for state-of-the-art heart disease research

Author	**Classifier**	**Features**	**Acc(%)**
Shorewala (2021)	Ensemble	13	75.1
Ali et al. (2020a)	Ensemble	27	83.5
Bharti et al. (2021)	SVM	13	84.26
Latha and Jeeva (2019)	Ensemble	13	85.48
Ali et al. (2020b)	ANN	13	86.20
Haq et al. (2018)	SVM	13	88.00
Vijayashree and Sultana (2018)	SVM	13	88.22
Aliyar Vellameeran and Brindha (2022)	DBN	13	88.8
Tuli et al. (2020)	DL	13	89
Sarra et al. (2022)	SVM	13	89.47
Proposed Baseline	**Log-Reg**	**13**	**86.32**

5.4.2 Ensemble Imputation Methods

Section 5.3 outlines the system design for testing the hypothesis that using different single imputation methods to fill missing data in different copies of a test dataset and then using different voting methods is an excellent approach to deal with missing data. The Cleveland heart disease dataset contains no missing data thus leading to a need to remove data from the unseen test dataset. Two approaches were used to remove data:

1. Systematic Removal (SR) of Data: 10% of the data is removed from 3 features (30% in total). Each combination of 3 features is run five times, and the average is presented. There are 13 features in the dataset which means that there are 286 different combinations of 3 features that can be used (1430 were undertaken due to the repeated experiments). For each of these combinations 10% of the data is randomly removed accross the dataset.

2. Random Removal (RR) of Data: 30% of the data is removed randomly

across the full unseen dataset from any feature. This is run 100 times. No data is removed from the target variable.

Once the unseen dataset with data removed was created, four copies of the dataset were created; each dataset had a different method for imputation, Mean, Mode, 0 fill and 1 fill.

The optimised Logistic Regression algorithm developed in section 5.4.1 provided predictions for each run with a different imputation method. Each resulting prediction had two different voting methods to determine the final prediction (Max Voting & Averaging).

Table 5.5 shows the PRE and REC obtained for the three most successful runs of the SR and RR methodology. For Systematic removal in many instances, the PRE score is increased and the REC results are comparable to those obtained from utilising the complete unseen dataset.

The REC range for all the experiments was between 0.61 – 0.79. The range is comparable to those that have utilised the dataset to generate optimised algorithms. Table 5.6 shows the PRE ranges presented in literature for two different research projects.

Table 5.5: Precision, Recall & F1 scores of the models considered in this work (as a percentage) for both methods of data removal and the scores compared to the baseline results.

Data Removal	Voting Method	Recall	Prec.	F1 Score
Baseline	-	0.87	0.77	0.82
SR	Max Voting	0.92	0.79	0.85
SR	Max Voting	0.71	0.68	0.69
SR	Averaging	0.80	0.61	0.69
RR	Max Voting	0.79	0.73	0.76
RR	Averaging	0.81	0.73	0.77
RR	Max Voting	0.75	0.67	0.71

Table 5.6: Precision range scores for the proposed EIM and different research projects

Base Classifier	Precision Range Scores
Decision Tree (A.Sabay et al., 2018)	69-77%
Decision Tree (Tu et al., 2009)	72%
Proposed Method	**61-79%**

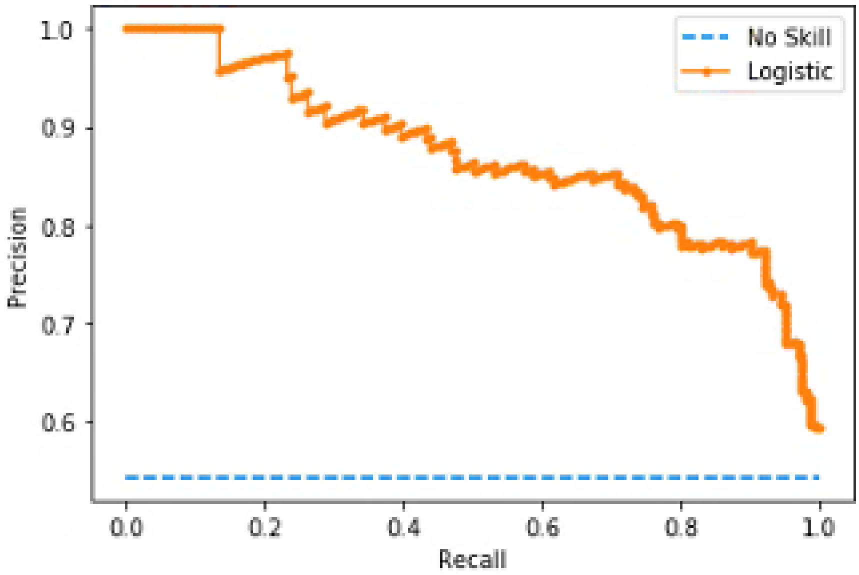


Figure 5.5: Precision-Recall curve for the developed pipeline ensemble model with max voting as the ensemble voting method.

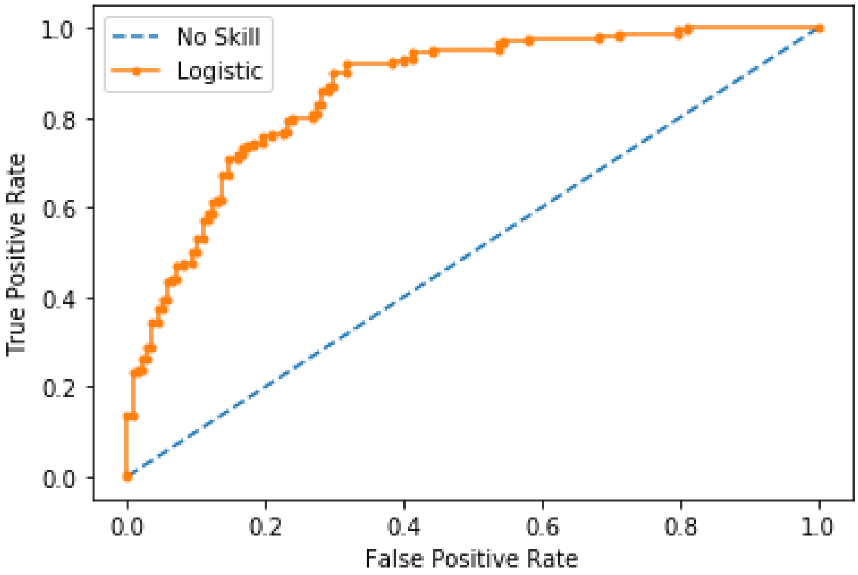


Figure 5.6: Receiver-operating characteristics curve for the developed pipeline ensemble model with max voting as the ensemble voting method and systematic removal of three features

5.5 Conclusion and Further Work

The key contribution of this Chapter is the development of a new method of dealing with missing data which makes use of ensemble methods and varying imputation methods. The proposed novel approach has produced classification results (comparable to state-of-the-art with missing data) for effectively determining heart disease. Moreover, the experiments conducted show that there is a significant increase in the precision. This increase in precision is very important in the medical domain, as false positives are more tolerable than false negatives (which could lead to loss of life). The proposed model is also efficient and can be easily modified for different imputation methods. For the experiments in this Chapter Mean, Mode, 0 Fill, and 1 Fill were considered. Further work is planned to study different imputation methods and apply the proposed approach to datasets that already have existing data missing.

5.6 Summary

The results obtained from any ML problem varies dramatically depending on the algorithm used, the hyperparameter tuning and how much data is missing. One of the key aspects to consider is the type of missing data. If the dataset has missing records, it can reduce the amount of overall training data or cause the model to produce inaccurate results. This problem is magnified in real-world applications as it is sometimes not possible to go back and retrieve the missing data due to an instrument failure or the costs associated with it. This

Chapter has proposed and demonstrated an alternative approach to dealing with missing data by utilising ensemble techniques. Varying imputation values for different types of missing data were used.

Section 5.2 has introduced some of the key concepts around missing data, the effects of missing data and different techniques that are commonly used in data to deal with these problems. Section 5.2.2 introduced the concept of ensemble learning and how it can be used to improve the results during ML training it also describes the three steps that need considering for ensemble learning. The hypothesis of the work is introduced in section 5.3 with a detailed methodology of how data will be removed randomly and systematically. Furthermore, a clear plan of testing is provided which shows all the different tests undertaken to demonstrate the results.

Section 5.4 shows that the resulting ML model produces results for heart disease classification that is comparable to current state-of-the-art research papers when reframing the problem to classify patients that have or do not have heart disease.

By using Logistic Regression and the complete Cleveland heart disease dataset, a baseline set of results was obtained that was used to compare the results from ensemble testing. Further work can be undertaken to improve this result by utilising the hyperparameter tuning and data rebalancing methods described in section 4.3.4.

The results produced by using ensemble techniques for missing data imputation can increase the overall sensitivity of the dataset. Table 5.5 provides in-depth results and shows how it is increased for an unseen test set as well. The ability to effectively impute data can have a large impact on the reliability and confidence of a ML model. Typically models trained on finite datasets lack confidence and are not applicable to larger cohorts of patients. The work undertaken in this Chapter has gone some way to prove that ML models can be modified for different medical use cases and can produce reliable results even if there is missing data within a patients record. By using this method, some medical tests or required samples that can take a long time to obtain could potentially be synthetically imputed into the dataset if it is a time-critical medical emergency.

Chapter 6

Detection of Early Onset Sepsis

6.1 Introduction

From an estimated 44000 deaths in the UK in 2018, to a estimated 52000 deaths in 2019, the number of people contracting sepsis leading to death is on the rise. Sepsis occurs when the body's immune system starts to send Infection-Fighting Chemicals (IFCs) to the body rather than just to the infected area. These IFCs can lead to inflammation and begin attacking healthy tissues. As a result, the body is no longer just fighting the infection, but is also fighting itself. Currently, researchers do not know why this happens and what exactly triggers sepsis. However, successful management of sepsis in a clinical setting is possible - though it requires prompt recognition of sepsis before the issue spreads (Aitkenhead and Dodds, 2018).

Medical Early Warning Systems (EWS) were first developed in the 1990s to predict the onset of sepsis. Most use some set of physiological parameters (including heart rate and systolic blood pressure) that are recorded at regular intervals. Weightings are given to values that deviate from the ‘normal range’ of what is expected, and this is used to predict sepsis before it progresses. However, whilst EWS have been widely adopted, there is little to no evidence that they improve patient outcomes due to concerns about reliability (McLymont and Glover, 2016). This Chapter looks to build on the work done to developed EWS and understand the process associated with building an EWS with heavily imbalanced datasets.

The remainder of this Chapter is structured as follows: Section 6.2 provides background on sepsis, how it is currently determined in medical care, and why detection is so vital. Sections 6.2.3 and 6.2.4 describe the under sampling and over sampling methods used in ML to date. Section 6.3 then introduces the methods used to develop an effective model, whilst also describing the dataset in more detail.Section 6.4 then presents the validated results using both ROC curves and confusion matrices for performance assessment. Finally section 6.6 presents conclusions, limitations and key areas for further work.

6.2 Related Work

6.2.1 Current Methods for Sepsis Prediction

Early detection of sepsis has been addressed in a number of studies. ML has proved an effective way of detecting sepsis with some very good results.

Doggart and Rutherford (2019) described an approach for detecting Sepsis by using a randomly under-sampled, boosted tree methodology which achieved an area under the AUROC score of 77.79%.

Vicar et al. (2019) used an LSTM classifier to detect the onset of sepsis. To overcome the fact that the dataset is heavily imbalanced, they applied Dice loss providing automatically weighted classes by the co-occurrence of features.

Tran et al. (2019) proposed a novel neural network called AEC-Net. The ACE-Net contains two main components; an auto encoder, and a Fully Connected Neural Network (FCNN). For each iteration, the loss of the auto encoder and FCNN are minimised. This process helped the model to provide better generalisation to unseen data. Finally, an ensemble pipeline of Random Forests, Gradient Boosting Decision Trees and the newly introduced AEC-Net was used to generate the normalised score (Tran et al., 2019).

Finishing third overall in the Physionet 2019 early onset sepsis detection challenge, Zabihi et al. (2019) used a systematic approach for sepsis prediction, by defining a new set of features to model the missingness of the clinical data. They developed a pipeline comprising of three main aspects: feature extraction, feature selection, and classification. The pipeline used an ensemble wrapper based, feature selection classifier which is similar to XGBoost.

Table 6.1 summarises the characteristics of the related literature. The main difference between these works and that presented in this Chapter, are the techniques used to rebalance the dataset so that the data contains a 50-50 split between both positive and negative cases. In addition the dataset used in this work makes use of the Physionet 2019 early onset sepsis detection challenge dataset, which is also used in a lot of the related literature. Those that use different datasets have been highlighted in Table 6.1. The table also contains information and results from different research papers and detailed (where available) the achieved area under the curve score.

Author	**Technique**	**Features**	**Dataset**	**Main Results**
Doggart and Rutherford (2019)	Randomly under Sampled Boosted Tree	Randomly under sampled tree boosted tree and ROC Curve metrics	Physionet set A	AUROC:77.79%
Vicar et al. (2019)	LSTM	LSTM classifier and dice loss for weighted classes based on the number of occurrences	Physionet set A	NUS: 0.281
Tran et al. (2019)	AEC-Net, Random Forest, Gradient boosting decision trees	An AEC-Net classifier was introduced and then ensembled with random forests and gradient boosting decision trees to produce the normalised utility score from the unseen data set	Physionet set A/B/C	NUS: -0.262 AUROC: 79.60%
Zabihi et al. (2019)	ensembled wrapper feature selection classifier	feature extraction, feature selection and classification are the three stages that make up the pipeline for this submission. Ensembled wrapper feature selection classifier was used	Physionet Complete Set	AUROC Score: 83.33%
Lyra et al. (2019)	Random Forest Classification	. . .	Physionet Set C	AUROC Score: 78.00% Acc: 0.730
Murugesan et al. (2019)	XGBoost	. . .	Physionet set A/B/C	NUS: 0.131
Zabihi et al. (2019)	BoostARoota	New feautures are added to represent the missingness of the clinical data.	Physionet set A/B/C	AUROC: 0.83% NUS: 0.339
Singh (2018b)	XGBoost	. . .	Phyiso Net Set A/B/C	AUROC: 0.8377
Abromavičius et al. (2020)	Decision Trees	Patients are divided based on their length of stay time within the ICU. An appropriate model is then applied.	Complete Physionet Dataset	AUROC: 0.313 NUS: 0.242
Yee et al. (2019)	Bayesian Networks	A data-driven, expert knowledge agnostic method was designed to build a screening algorithm for early detection of septic shock	ICU patients	AUROC: 0.81
Shashikumar et al. (2019)	AISE algorithm (based on a modified Weibull-Cox proportional hazards model)	...	27,527 patients	AUROC:0.83-0.85
Rosnati and Fortuin (2021)	MGP-AttTCN:	a logisitc regression classifier and the InSight baselines were compared to the scores generated from the interpreted ML algorithm	MIMIC-III dataset	AUROC

Table 6.1: Comparative analysis of related work in early sepsis detection

6.2.2 Data-Level Methods For Dealing With Class Imbalance

Class imbalance refers to a problem where the classes in the target variable are not represented equally. For example, in a binary classification problem with a dataset that contains 99% of observations of one class and 1% of observations of the other class, this would be an example of class imbalance. This can lead to difficulty in training models, as they may be biased towards the more prevalent class, and have difficulty in accurately identifying the less prevalent class.

There are many approaches to handling class imbalance. Data level solutions include many different forms of re-sampling such as: random over-sampling, random under-sampling, directed over-sampling, directed under-sampling, and combinations of the re-sampling techniques.

6.2.3 Oversampling

Random over-sampling is a non-heuritstic approach that deals with unbalanced classes through random replication/generation of the minority classes. However, random over-sampling can increase the chances of a model overfitting. Overfitting occurs because the existing minority class data is used to make copies within the answer plane. A classifier will then construct rules that are apparently accurate based on the data provided to it; however, this may just cover one replicated example. Furthermore, the addition of data can increase the time needed to train a ML model as it adds additional data into the training phase.

SMOTE

The SMOTE technique, was first introduced by Chawla et al. (2002). The technique generates new synthetic instances of the minority class. Utilising the K-Nearest Neighbour (k-NN) algorithm, SMOTE can generate new samples that are within the vicinity of existing samples in the minority class. The k-NN samples are chosen at random the along the lines which join all of the minority samples. Section 4.3.4 in Chapter 4 provides and indepth review of the SMOTE technique

6.2.4 Undersampling

Random under-sampling is a non-heuristic method that aims to rebalance class distributions by randomly removing examples from the majority class. This method is typically used to overcome the idiosyncrasies of ML algorithms. Unlike oversampling, which introduces synthetic data, undersampling can lose valuable information as data is randomly dropped. This can result in the model not fitting as well as required for the ML problem. If samples are drawn randomly, statistical analysis shows that the sample distribution can be used to estimate the population distribution that they were drawn from.

Random Under Sampling Boost (RUSBoost)

The RUSBoost algorithm is based on the SMOTEBoost algorithm (Afifi and Elashoff, 1966), which is, in turn, based on the AdaBoost.M2 algorithm (Townsend, 2014). SMOTEBoost improves upon AdaBoost by introducing an intelligent oversampling technique (Little and Rubin, 2002), which helps to balance the class distribution, while AdaBoost improves the classifier performance using this re-balanced data. RUSBoost achieves the same goal, but uses Randomly-Under Sampled (RUS) rather than SMOTE. The result is a simpler algorithm with faster model training times and favorable performance (Seiffert et al., 2008).

6.3 Methods & Material

6.3.1 Dataset

The work in this Chapter is based on data collected from Electronic Medical Record(s) (EMR) of two hospitals systems: Beth Israel Deaconess Medical Center and Emory University Hospital, which were made readily available on the PhysioNet database as part of the 2019 PhysioNet challenge (Reyna et al., 2020). The dataset includes data and labels for 40,336 patients from both hospitals.

The dataset consists of hourly vital sign summaries, lab values, and static patient descriptions. In total there were 40 features recorded hourly. There are over 15 million datapoints within the whole dataset.

Multiple measurements taken each hour had the average value calculated so that each record only had hourly samples - subsequently simplifying model development. Missing data and erroneous data was present in the dataset and required dealing with in the model development. The ground truth output for each sepsis patient was labelled in accordance with the Sepsis-3 clinical criteria (Reyna et al., 2020).

- $\mathbf{t_{suspicion}}$: Clinical suspicion of infection identified at the earliest timestamp of IV antibiotics and blood cultures within a given time interval. If IV antibiotics were given first, then the cultures must have been obtained within 24 hours. If cultures were obtained first, then IV antibiotic must have been ordered within 72 hours. In either case, IV antibiotics must have been administered for at least 72 consecutive hours.

- $\mathbf{t_{SOFA}}$: Occurrence of organ failure as identified by a two-point increase in the SOFA score within a 24-hour period.

- $\mathbf{t_{sepsis}}$: Onset of sepsis identified as the earlier of $t_{suspicion}$ and t_{SOFA} (as long as t_{SOFA} occurred no more than 24 hours before or 12 hours after $t_{suspicion}$(Reyna et al., 2020))

6.3.2 Classification Algorithms

Boosting is an example of an ensembling modeling technique that attempts to develop a strong classifier from a collection of weak classifiers. A set of weak models are built in a series of steps. Firstly, a model is developed from the training data. Secondly, a new model is built which will try and reduce the errors found in the first model iteration. This procedure is iterated, and additional models are added until either correct predictions are made for all of the training data or the maximum number of models has been added. Examples of boosting algorithms are:

- **AdaBoost (AB):** AdaBoost is used for binary classification and creates a strong classifier from a weak one. It is an ensemble meta-algorithm.

For each new model instance, the boosting method emphasises training instances that were misclassified. AB minimises the experimental loss.

- **Random Forest (RF):** Random forests are fundamentally a collection of Decision Trees. Each tree is constructed with a subset of the data therefore making sure that each tree differs as much as possible. Classification decisions are made by the majority decision given from the trained trees. A detailed explanation of Random Forests is in Section 2.7.5.

6.3.3 Experimental Setup

To build an effective ML pipeline that can accurately infer if a patient will get sepsis faster than existing support methods, we utilise the ML pipeline development framework defined in Figure 6.1. The framework is made up of the following steps:

1. The dataset is split into three sections. Similarly to the work defined in Chapter 4 the data is split: 70%, 20%, and 10%.

2. The 70% training data is used to train three different models making use of the Logistic Regression, ADAboost and Random Forests algorithms, to find out which would give the best result based on a defined performance metric.

3. The models then uses oversampling and under-sampling techniques to rebalance the dataset and ensure the class balance is consistent.

4. The final model is tested on the 10% unseen data which will show how well the model generalises to unseen data.

The experiments conducted in this body of research were completed using Python and making use of the Sci-kit learn collection of tools called "imblearns" which has the facility to use SMOTE and RUS protocols and build varying reblancing pipelines (Pedregosa et al., 2011). Throughout the experimental process these software packages were used to implement: feature selection, class balancing and a series of ML classifiers. Missing values were all treated the same and removed from the dataset, as the amount of records with missing data was negligible. Each different classifier was tested using both resampling methods, to see the effects before they were both combined and tried together. Combining the two classification algorithms and investigating two methods of sampling data and the combination of them both, meant there were several different experiments that will be run. Ten fold cross-validation was used to mitigate variability in the results and remove bias. The resampling methods were only implemented during the model development process and will not be used on the unseen validation dataset.

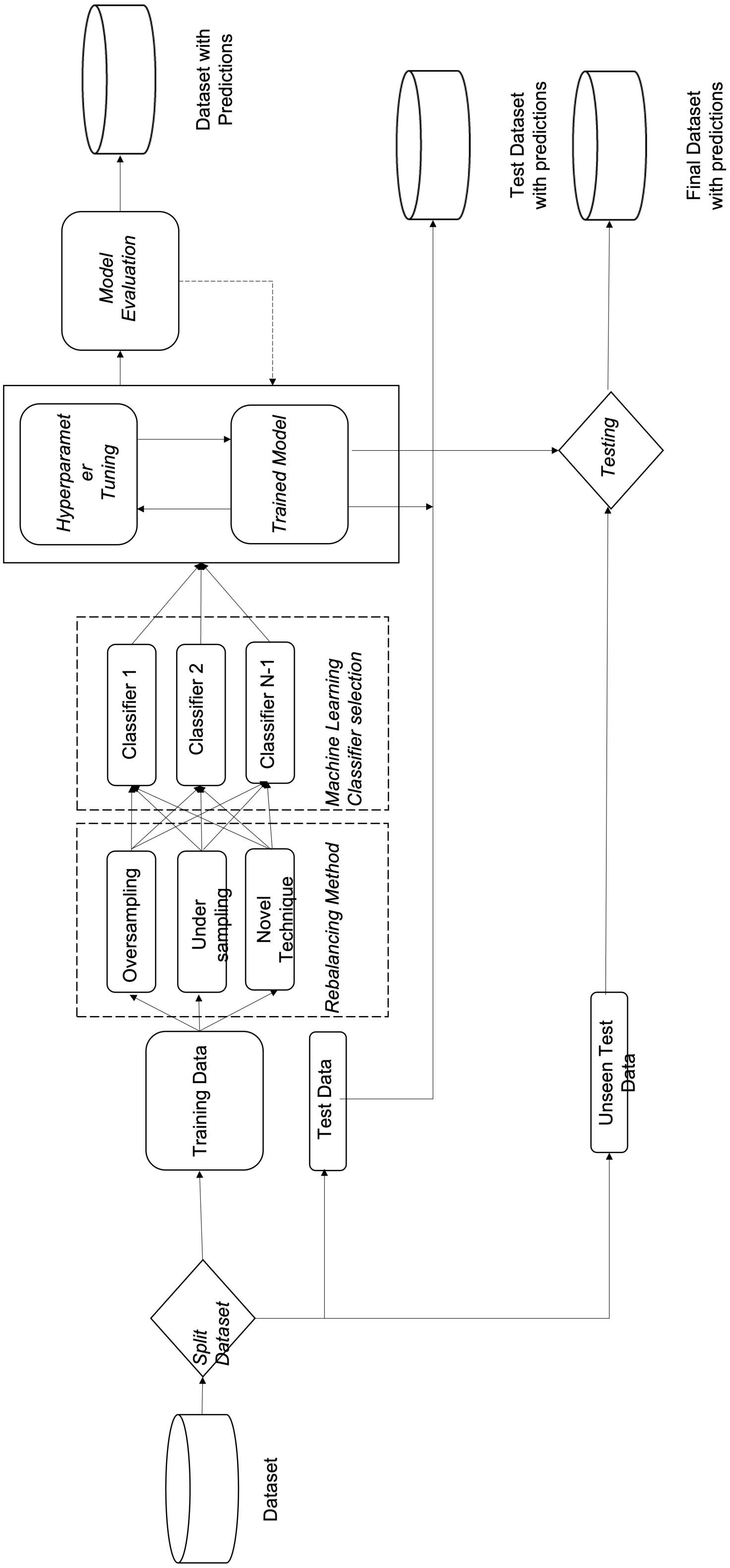


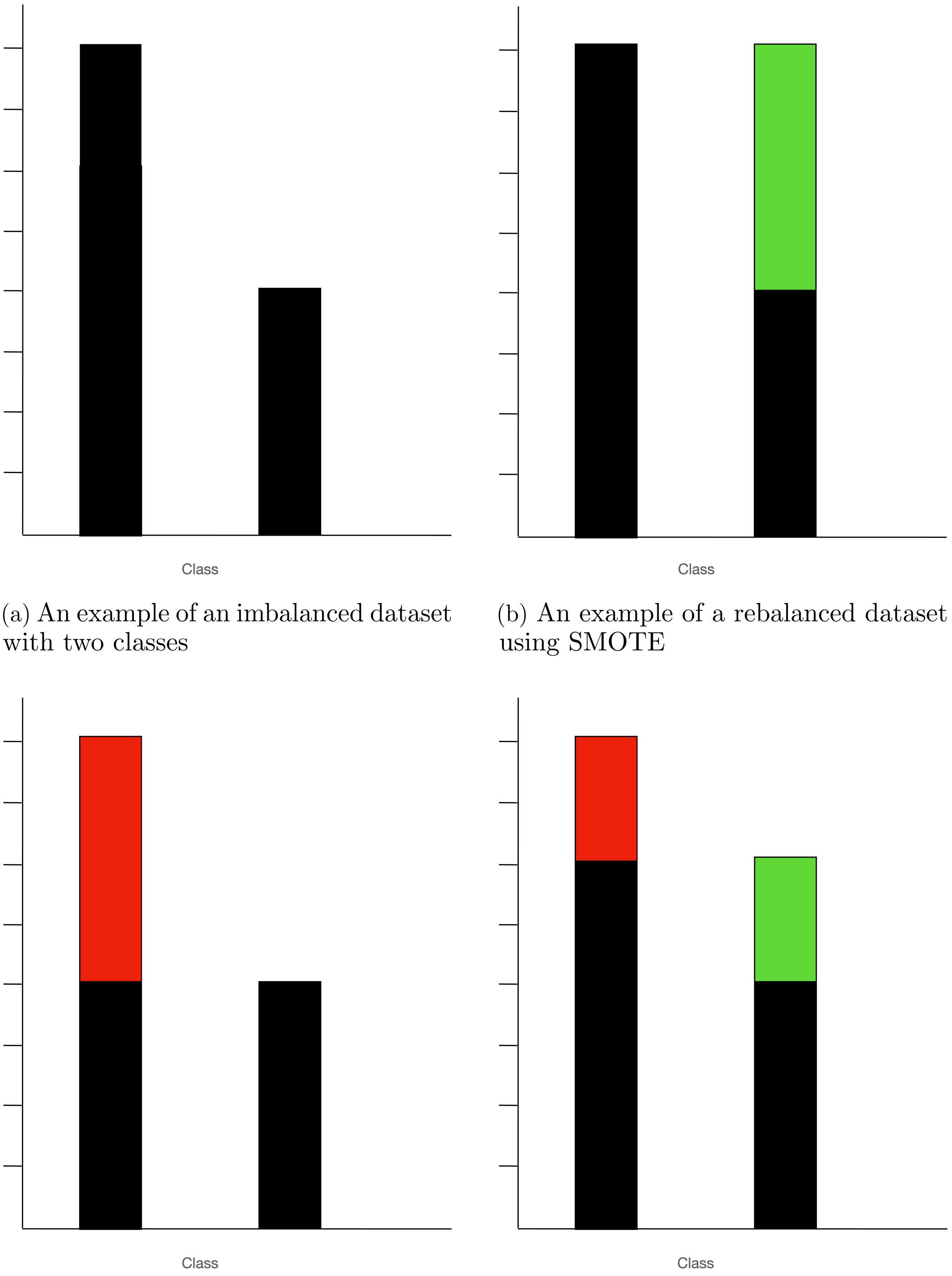
Figure 6.1: Machine Learning Pipeline for Sepsis Detection

6.3.4 Combining Rebalancing Methodologies

Section 6.3.3 introduces the concept of combining different rebalancing methodologies. SMOTE used on heavily imbalanced data can result in optimised models that overfit and do not generalise well on unseen data. The introduction of synthetic data is only produced by using existing data within the class. This technique decides on feature space and assumes that features with similar attributes are close together which indicates that they belong in the same class programmatically. No expert or domain knowledge is considered in this process. Similarly, random under-sampling removes cases randomly from the majority set. As data (especially in the medical domain) can be challenging to acquire, removing data is only sometimes the best solution due to tight class boundaries. Furthermore, some data could be removed that shows key aspects and differences between the models.

Throughout this Chapter, the SMOTE and random under-sampling methodologies will be used; however, combining them will be considered. Figure 6.2a – Figure 6.2d show all the different effects of under-sampling and over-sampling on a dataset. A green section indicates data to be added, and a red section indicates data that is removed. The combination of random under-sampling and SMOTE will be used for this work. Different amounts of data are removed and synthesised to demonstrate the effects of doing it. Four combinations are used: 5%, 10%, 15% and 20% of the total dataset will be synthetic data, and the rest will be under-sampled to present a classification problem with equal balance in each dataset.

The same framework described in section 6.3.3 and shown in Figure 6.1 will be used for the combination of resampling methods, enabling the comparison of results.



(a) An example of an imbalanced dataset with two classes

(b) An example of a rebalanced dataset using SMOTE

(c) An example of a balanced dataset using undersampling techniques

(d) Novel technique of using a combination of different rebalancing techniques

Figure 6.2: The effects of different re-sampling methods on datasets.

6.4 Experimental Results

Table 6.2: Table to show results from varying rebalancing techniques applied the sepsis dataset.

Classifier	Technique	Synthetic Data (%)	AUROC
Logisitc Regression	-		0.69
Logisitc Regression	SMOTE		0.74
Logisitc Regression	Undersampling		0.76
Logisitc Regression	Combination	5	0.75
Logisitc Regression	Combination	10	0.78
Logisitc Regression	Combination	15	0.79
Logisitc Regression	Combination	20	0.77
Random Forest	-		0.73 (0.20)
Random Forest	SMOTE		0.74
Random Forest	Undersampling		0.69
Random Forest	Combination	5	0.7
Random Forest	Combination	10	0.7
Random Forest	Combination	15	0.7
Random Forest	Combination	20	0.71
AdaBoost	-		0.78 (0.04)
AdaBoost	SMOTE		0.79
AdaBoost	Undersampling		0.77
AdaBoost	Combination	5	0.79
AdaBoost	Combination	10	0.79
AdaBoost	Combination	15	0.81
AdaBoost	Combination	20	0.84

Table 6.2 provides the results for the different experiments outlined in sections 6.3 and 6.3.4. Each classifier is tested with no rebalancing technique. The runs act as a baseline classifier so that the results can be compared. Logistic Regression obtained an AUROC score of 0.69 whereas Random Forests and AdaBoost produced far superior results 0.77 and 0.78. One of the critical drawbacks of Random Forests is the initial split up of the features; this is demonstrated as the standard deviation for the baseline classifier was 0.2. Standard deviation scores are not included within table 6.2 but are described

in the text required.

The AdaBoost algorithm performed the best of the three classifiers when synthetic data was introduced for the testing with SMOTE. The ensemble nature of the algorithm has demonstrated that it can discriminate between different levels of sepsis detection.

As the dataset is heavily imbalanced, under-sampling the dataset could demonstrate how ineffective the model is at determining different levels of sepsis, as the mistakes are magnified due to less data in the dataset[1]. These results have been seen in the Random Forest algorithm which saw a downgrade in performance compared to the baseline and SMOTE tests.

For each of the combination methods typically, there is a trend that as more synthetic data is introduced, it is easier to discriminate between classes. This could demonstrate that there is a wide feature space and well-defined boundaries between results. For all classifiers this has remained true, however when introducing 20% synthetic data to the Logisitic Regression classifier, there was a downgrade in performance of AUROC score – 0.77 compared to 0.79 for the 15% synthetic data. As outlined previously, all versions of rebalancing techniques have some element of randomisation. This randomisation may be a factor as to why the performance has degraded.

The combination of 20% synthetic data via SMOTE and under-sampling with the AdaBoost classifier has provided the best AUROC score of 0.84. This result is comparable to those described in the literature, most notably

[1] A dataset containing 100 patients who may have cancer, the target variable indicated 99 do not have cancer, but 1 does. An optimised algorithm could demonstrate that it has achieved 99% accuracy. Still, it cannot determine those who have cancer as there is not enough information

(Doggart and Rutherford, 2019) who made use of a Boosted Tree algorithm. The XGboost, AdaBoost, Boosted Treed classifiers are based on converting weak learners into strong learners. Adaboost is usually robust to overfitting something demonstrated in our results with a standard deviation of 0.02 (when the 20% combination). These results were not presented by other researchers and cannot be compared.

6.5 Discussion

The work introduced in this Chapter demonstrated that existing EWS for determining sepsis needs to be updated. The detrimental effects of not detecting sepsis early shows that it is an area of research that needs to be developed. As part of the Physionet 2019 challenge, the freely available sepsis dataset has gone some way to help with this research problem Reyna et al. (2020). The results presented demonstrate that it is possible to detect sepsis early with simple ML algorithms that can be trained very quickly to produce valid results. The results show that boosted algorithms tend to perform better on the data, as this approach allows the model to pick a result based on multiple weak learners.

The results shown in table 6.2 have shown that they are comparable to the current state-of-the-art literature in this field. While the combination methodology introduced in section 6.3.4 has shown promise as an alternative approach to using either under-sampling or over-sampling, the results obtained were consistently higher or comparable. Further experiments could be undertaken on different problems with heavily imbalanced datasets to demonstrate and further prove the hypothesis.

Randomness is a crucial component of any resampling technique and must be caveated to make the user aware of any methodologies used. Multiple tests and cross-validation methods should be used to try and combat this randomness and reduce the variance within the results.

The resulting AdaBoost algorithm has shown promise in determining sepsis early. Further work could be completed to optimise the algorithm by using some of the hyperparameter techniques that have been introduced in section 4.3.4 of Chapter 4.

6.6 Conclusion

The current study indicates that different sampling ML techniques as part of an optimised model, can be used to rebalance datasets to detect outcomes accurately. The method introduced that combined both over sampling and under sampling techniques provided the best AUROC scores. In this case, our ML pipeline understands all of the psychological features used to detect sepsis within a hospital. Where applicable studies will make sure that the dataset has a balanced output to prevent the class imbalance problem. The phiysonet dataset used in this body of work had a disparity of the classes in the variables. The majority class was for patients who had sepsis, conversely, for the minority class patients did not have sepsis.

Patient data for 40,336 patients were used to train and test the model. The Random Forest classifier achieved a maximum AUROC of 0.71, it was the AdaBoost algorithm that achieved the highest AUROC score of 0.84.

These results indicate that the produced ML model has fewer limitations compared to traditional early sepsis detection approaches. Additional steps including feature removal and datasets from different locations, should be further explored to augment sepsis detection which will result in improved patient care by preventing sepsis before it becomes fatal to a patient.

Chapter 7

Conclusions

Driven by the increase in computational power and larger amounts of storage and memory, and coupled with the vast amounts of data being generated and stored by all industries, ML is being used to perform a wide range of complex tasks with high levels of accuracy.

Modelling and inferring predictions to support medical practitioners is a key area of research that needs to be continually reviewed and designed. ML techniques such as, but not limited to, Linear Regression, Random Forests and AdaBoost algorithms can produce effective models for specific use cases in medicine and, in some cases, outperform existing methods. Nevertheless the main drawback with these techniques is that the amount of training data required to develop an accurate model with high confidence and put it into production is not typically available in the research domain. Furthermore, the lack of training data limits the results that can be obtained by certain algorithms such as Neural Networks.

7.1 Machine Learning in Health Care

The optimised ML framework proposed in Chapter 4 aims to improve upon current methods used to predict the mortality of patients admitted into ICU. With more people admitted into ICU daily it is critical that medical professionals make informed decisions in a timely manner. These decisions can ultimately effect the chances of a patient's survival.

Unlike other industries such as the automotive and the aerospace industry, the medical industry is slow to uptake new concepts due to the rigorous amount of testing required (Topol, 2019; Syed, 2016). The current techniques used in mortality prediction in ICU are trained on frozen datasets (a single period of time, usually out of date instantly) that don't necessarily consider the effects of different cohorts, time of admission, or how treatments change over time. An additional proposed methodology in Chapter 4 demonstrated how online training can be applicable to ICU mortality prediction to continuously update the models as treatments and patient cohorts evolve over time.

7.2 Missing Data Imputation

"Garbage in, Garbage out" is the cornerstone of training ML models. It means that the effectiveness of a ML algorithm is defined by the data used to train it. Commonly researchers will just use ad-hoc techniques to remove data, including case deletion and simple imputation techniques. Chapter 5 has sought to outline the dangers of missing data and show some of the issues that can occur in the medical domain when dealing with missing data. Many

algorithms make use of ensemble methods to help models generalise better to unseen data – combining different algorithms and voting on the prediction that occurs the most. A key contribution of Chapter 5 was the development of a novel ensemble based imputation method to replace single imputation techniques. This reduces the time needed for researchers to think about which imputation techniques are most effective for their use case, whilst also proving that combining different imputation methods can improve the sensitivity of the output.

Chapter 5 also described the development of a simple ML pipeline for detecting coronary heart disease. Section 5.4 alludes to how the developed pipeline outperformed current state of the art techniques.

7.3 Rebalancing Datasets

Chapter 6 has shown that using ML techniques can detect sepsis 6 hours before traditional methods can. A key issue that was apparent in the development of the optimised ML pipeline is the unbalanced nature of sepsis datasets. A major contribution of Chapter 6 is the rigorous performance comparison of different rebalancing algorithms described in literature and the effects they have on detecting sepsis. It was shown that rebalancing data can result in the overfitting of models which can result in poor performance in the real-world.

The effectiveness of the rebalanced optimised ML model developed in Chapter 6 was demonstrated on a real-world problem. The results presented prove that, for the unbalanced sepsis dataset, rebalancing the dataset by reducing the size of the majority class is an effective way of ensuring that the model does not overfit. In contrast, Chapter 6 shows that when introducing synthetic data via SMOTE or other techniques, the resulting models are prone to overfitting on the training data. Furthermore, a novel rebalancing technique that made use of both SMOTE and undersampling was introduced and tested on the dataset. These results demonstrated that the combination of resampling techniques can improve the overall accuracy of a model and reduce the level of overfitting of the model.

7.4 Future Challenges

7.4.1 Uptake Of The Techniques Within The Industry

There is still a long way to go until ML algorithms are accurate enough across a global population to be used as support tools for medical practitioners. Although ML research in the medical domain is becoming more prominent, there often isn't enough data to confirm that the model is ready for deployment due to the ethics surrounding the sharing of electronic health records. It is not easy to obtain medical data that is anonymous and complete. However, AI is already being integrated into decision support tools in medicine in some limited applications. To increase the development of medical decision support tools using AI, the medical industry must make large datasets readily available[1] for research and industry projects.

[1] That are also consistent across the globe - i.e. all of the data and features collected are the same

7.4.2 Automated Machine Learning

Automated ML is the process of applying ML algorithms to real-world problems without the need for manual selection of model architectures, training of the model, or tuning of hyperparameters to provide the best results. This process can dramatically reduce the time needed to build ML models. Furthermore, it is more user-friendly for non-ML practitioners and can be more accurate than basic hard coded algorithms. The Tree-based Pipeline Optimization Tool (TPOT) library utilises the same ML frameworks used in Chapter 4 and additional research into the results that can be achieved from applying these techniques. Understanding and providing a full description of how the techniques behave is a key aspect of any ML problem. Cloud providers such as Amazon Web Services (AWS) and Microsoft Azure have been producing many "black box" toolsets that take input data and produce an output with very limited descriptions of how the model is developed, what algorithm is used, and how the features are chosen. Even though these methods have been widely used by companies with limited resources, they have received some criticism due to their lack of transparency.

Appendix A

Mortality Prediction Results

In this section additional results, graphs and diagrams are presented. The results are detailed in Chapter 4 but do not need to be included in the body of text.

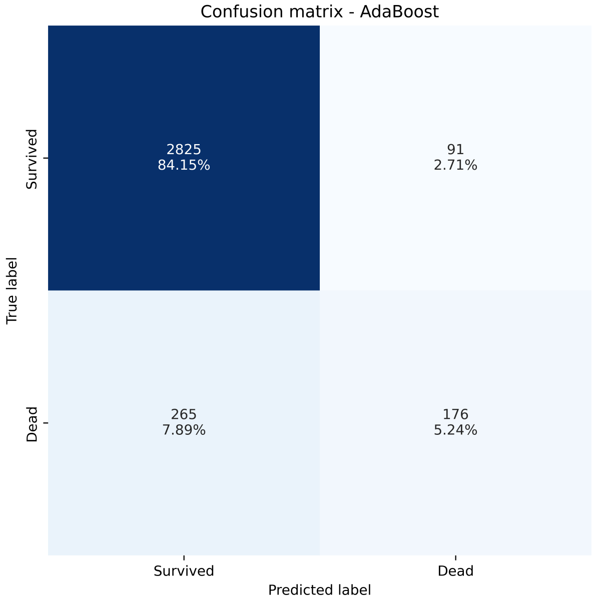


Figure A.1: Resulting confusion matrix produced from preliminary testing using an AdaBoost classifier to infer mortality in critical care units

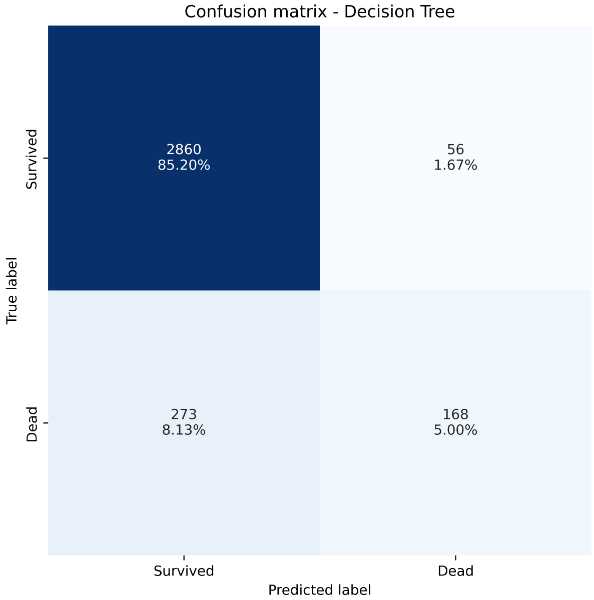


Figure A.2: Resulting confusion matrix produced from preliminary testing using a Decision Tree to infer mortality in critical care units

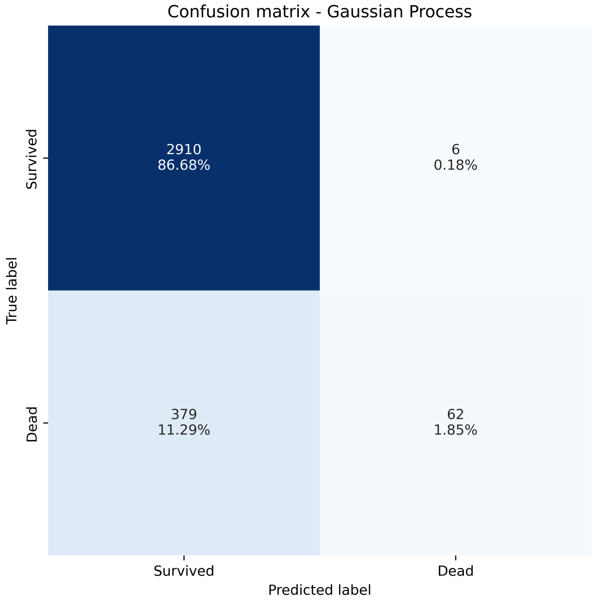


Figure A.3: Resulting confusion matrix produced from preliminary testing using the Gaussian Process classifier to infer mortality in critical care units

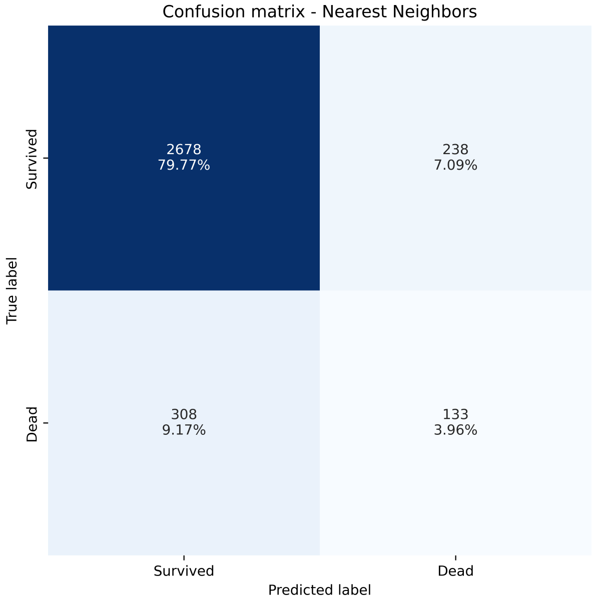


Figure A.4: Resulting confusion matrix produced from preliminary testing using a K-Nearest Neighbour classifier to infer mortality in critical care units

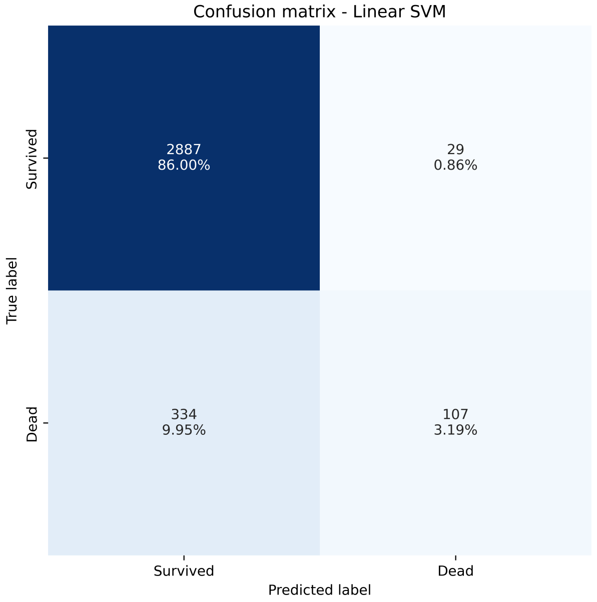


Figure A.5: Resulting confusion matrix produced from preliminary testing using a Linear SVM to infer mortality in critical care units

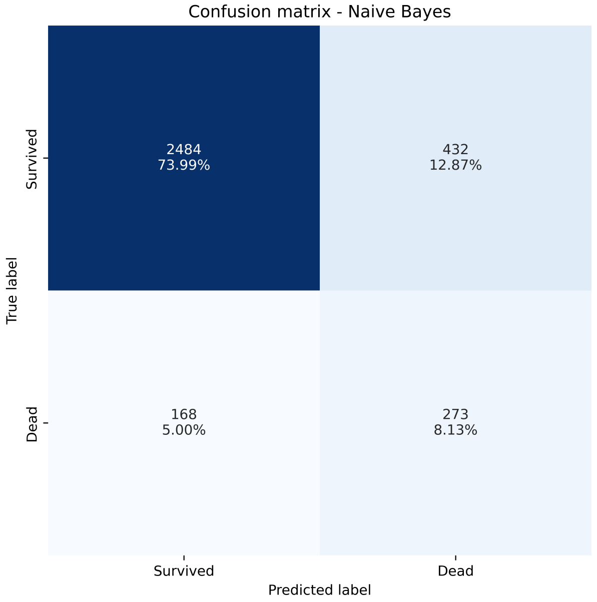


Figure A.6: Resulting confusion matrix produced from preliminary testing using a Naive Bayes to infer mortality in critical care units

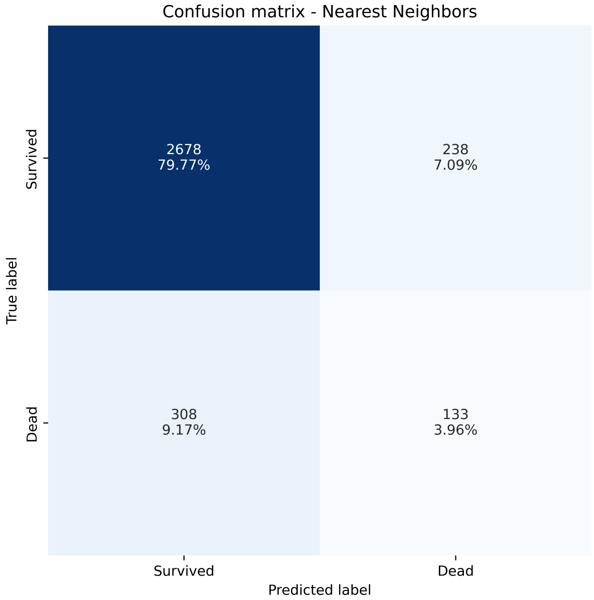


Figure A.7: Resulting confusion matrix produced from preliminary testing using a Neural Network to infer mortality in critical care units

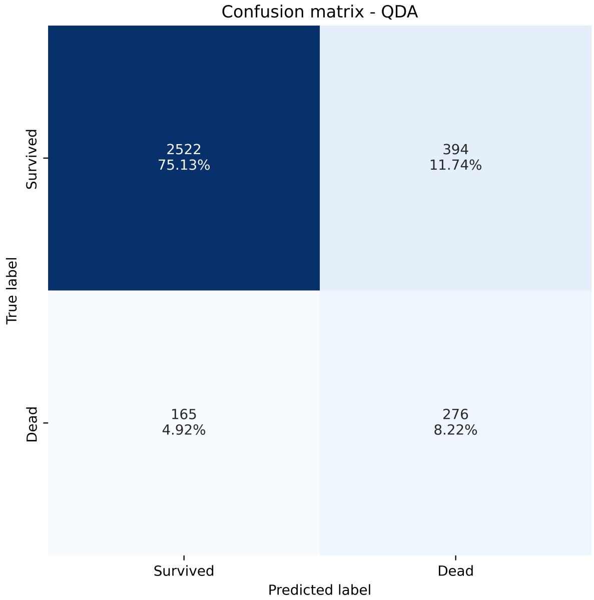


Figure A.8: Resulting confusion matrix produced from preliminary testing using QDA to infer mortality in critical care units

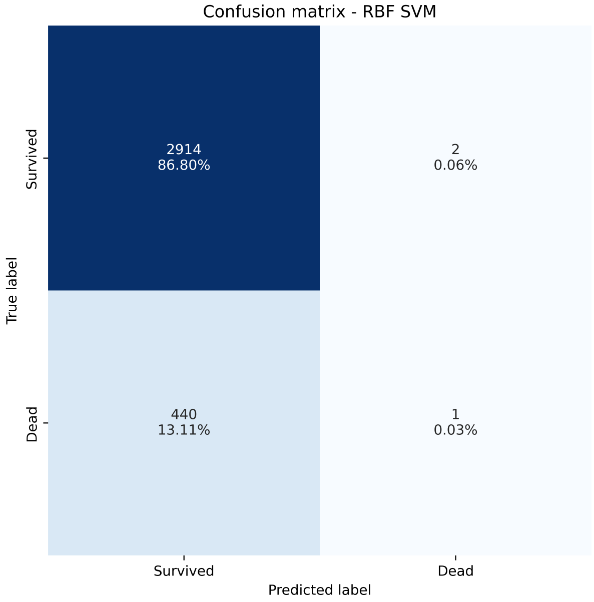


Figure A.9: Resulting confusion matrix produced from preliminary testing using a RBF SVM to infer mortality in critical care units

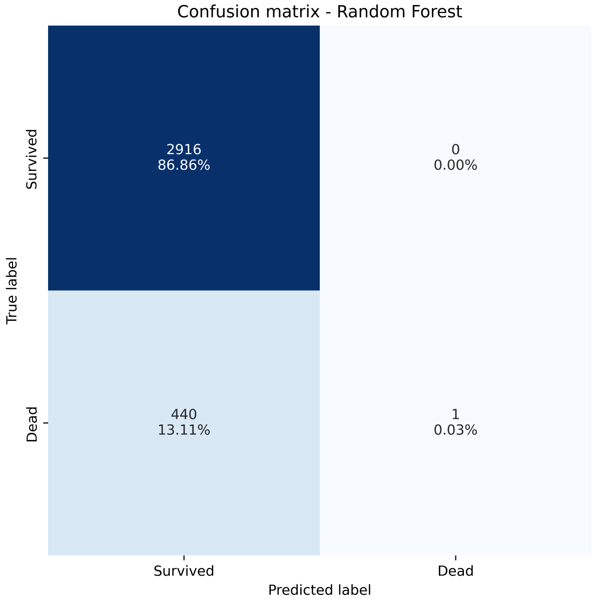


Figure A.10: Resulting confusion matrix produced from preliminary testing using a Random Forest classifier to infer mortality in critical care units

Bibliography

A Abernethy and S Khozin. Clinical drug trials may be coming to your doctor's office. *Wall Street Journal*, 2017. URL `https://scholar.google.com/scholar_lookup?title=Wall+StreetJournal&author=A+Abernethy&author=S+Khozin&publication_year=2017&`. 2022-12-19.

Vytautas Abromavičius, Darius Plonis, Deividas Tarasevičius, and Artūras Serackis. Two-stage monitoring of patients in intensive care unit for sepsis prediction using non-overfitted machine learning models. *Electronics*, 9(7):1133, 2020. ISSN 2079-9292. doi: 10.3390/electronics9071133. URL `https://www.mdpi.com/2079-9292/9/7/1133`. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Michael D. Abràmoff, Philip T. Lavin, Michele Birch, Nilay Shah, and James C. Folk. Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. *NPJ digital medicine*, 1:39, 2018. ISSN 2398-6352. doi: 10.1038/s41746-018-0040-6.

A.A. Afifi and R.M. Elashoff. Missing observations in multivariate statistics i. *Review of the literature. Journal of the American Statistical Association.*, 61(315):595–604., 1966.

Daiqiao Ai, Jingxing Wu, Hanxuan Cai, Duancheng Zhao, Yihao Chen, Jiajia Wei, Jianrong Xu, Jiquan Zhang, and Ling Wang. A multi-task FP-GNN framework enables accurate prediction of selective PARP inhibitors. *Frontiers in Pharmacology*, 13, 2022. ISSN 1663-9812. URL `https://www.frontiersin.org/articles/10.3389/fphar.2022.971369`.

Susan Aitkenhead and Sarah Dodds. Nhs england: Beating sepsis with early detection and prompt treatment, September 2018. URL `https://www.england.nhs.uk/blog/beating-sepsis-with-early-detection-and-prompt-treatment/`.

Farman Ali, Shaker El-Sappagh, S. M. Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, and Kyung-Sup Kwak. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63:208–222, 2020a.

ISSN 1566-2535. doi: 10.1016/j.inffus.2020.06.008. URL `https://www.sciencedirect.com/science/article/pii/S1566253520303055`.

Syed Arslan Ali, Basit Raza, Ahmad Kamran Malik, Ahmad Raza Shahid, Muhammad Faheem, Hani Alquhayz, and Yogan Jaya Kumar. An optimally configured and improved deep belief network (OCI-DBN) approach for heart disease prediction based on ruzzo–tompa and stacked genetic algorithm. *IEEE Access*, 8:65947–65958, 2020b. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2985646. Conference Name: IEEE Access.

Fathima Aliyar Vellameeran and Thomas Brindha. A new variant of deep belief network assisted with optimal feature selection for heart disease diagnosis using IoT wearable medical devices. *Computer Methods in Biomechanics and Biomedical Engineering*, 25(4):387–411, 2022. ISSN 1025-5842. doi: 10.1080/10255842.2021.1955360. URL `https://doi.org/10.1080/10255842.2021.1955360`. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10255842.2021.1955360.

Tiago Alves, Alberto Laender, Adriano Veloso, and Nivio Ziviani. Dynamic prediction of ICU mortality risk using domain adaptation. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1328–1336, 2018. doi: 10.1109/BigData.2018.8621927.

Rebecca R. Andridge and Roderick J. A. Little. A review of hot deck imputation for survey non-response. *International statistical review = Revue internationale de statistique*, 78(1):40–64, 2010. ISSN 0306-7734. doi: 10.1111/j.1751-5823.2010.00103.x. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3130338/`.

Sheena Angra and Sachin Ahuja. Machine learning and its applications: A review. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, pages 57–60, 2017. doi: 10.1109/ICBDACI.2017.8070809.

Jacob Aron. Forget the turing test – there are better ways of judging AI, 2015. URL `https://www.newscientist.com/article/dn28206-forget-the-turing-test-there-are-better-ways-of-judging-ai/`.

D. G. T. Arts, N. F. de Keizer, M. B. Vroom, and E. de Jonge. Reliability and accuracy of sequential organ failure assessment (SOFA) scoring. *Critical Care Medicine*, 33(9):1988–1993, 2005. ISSN 0090-3493. doi: 10.1097/01.CCM.0000178178.02574.AB. URL `https://journals.lww.com/ccmjournal/Abstract/2005/09000/Reliability_and_accuracy_of_Sequential_Organ.14.aspx`.

A.Sabay, L.Harris, V.Bejugama, and K. Jaceldo-Siegl. Overcoming small data limitations in heart disease prediction by using surrogate data. *SMU Data Science Review*, 1(3), 2018.

Awan-Ur-Rahman. *What is Artificial Neural Network and How it mimics the Human Brain?*, 2019. URL `https://medium.com/analytics-vidhya/what-is-artificial-neural-network-and-how-it-mimics-the-human-brain-f92c45564e20`.

Stephanie Baker, Wei Xiang, and Ian Atkinson. Continuous and automatic mortality risk prediction using vital signs in the intensive care unit: a hybrid neural network approach. *Scientific Reports*, 10(1):21282, 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-78184-7. URL `https://www.nature.com/articles/s41598-020-78184-7`. Number: 1 Publisher: Nature Publishing Group.

Amir Bein, Cicely W. Fadel, Ben Swenor, Wuji Cao, Rani K. Powers, Diogo M. Camacho, Arash Naziripour, Andrew Parsons, Nina LoGrande, Sanjay Sharma, Seongmin Kim, Sasan Jalili-Firoozinezhad, Jennifer Grant, David T. Breault, Junaid Iqbal, Asad Ali, Lee A. Denson, Sean R. Moore, Rachelle Prantil-Baun, Girija Goyal, and Donald E. Ingber. Nutritional deficiency in an intestine-on-a-chip recapitulates injury hallmarks associated with environmental enteric dysfunction. *Nature Biomedical Engineering*, 6 (11):1236–1247, 2022. ISSN 2157-846X. doi: 10.1038/s41551-022-00899-x. URL `https://www.nature.com/articles/s41551-022-00899-x`. Number: 11 Publisher: Nature Publishing Group.

Rohit Bharti, Aditya Khamparia, Mohammad Shabaz, Gaurav Dhiman, Sagar Pande, and Parneet Singh. Prediction of heart disease using a combination of machine learning and deep learning. *Computational Intelligence and Neuroscience*, 2021:8387680, 2021. ISSN 1687-5273. doi: 10.1155/2021/8387680.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

Alexandre Breant, Francis Turina-Malard, and Bertrand Kleinmann. Strategic alliances: the right prescription to survive the healthcare revolution, 2018. URL `https://ceptonstrategies.com/en/strategic-alliances/`. Section: 2018.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL `https://doi.org/10.1023/A:1010933404324`.

Jason Brownlee. *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*, 2015. URL `https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`.

Maryam Bukhari, Khalid Bajwa, Saira Gilani, Muazzam Maqsood, Mehr Durrani, Irfan Mehmood, Hassan Ugail, and Seungmin Rho. An Efficient Gait Recognition Method for Known and Unknown Covariate Conditions. *IEEE Access*, PP:1–1, December 2020. doi: 10.1109/ACCESS.2020.3047266.

Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada, 2019.

Denis Campbell and Denis Campbell Health policy editor. NHS waiting times 'driving people to turn to private treatment', 2017. ISSN 0261-3077. URL `https://www.theguardian.com/society/2017/sep/11/nhs-waiting-times-driving-people-to-turn-to-private-treatment`.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. ISSN 1076-9757. doi: 10.1613/jair.953. URL `http://arxiv.org/abs/1106.1813`. arXiv: 1106.1813.

Martin Childs. John McCarthy: Computer scientist known as the father of AI | the independent | the independent, 2011. URL `https://www.independent.co.uk/news/obituaries/john-mccarthy-computer-scientist-known-as-the-father-of-ai-6255307.html`.

Devin Coldewey. Quris combines AI with 'patient on a chip' to speed drug development and reduce animal testing, 2021. URL `https://techcrunch.com/2021/10/18/quris-combines-ai-with-patient-on-a-chip-to-speed-drug-development-and-reduce-animal-testing/`.

Gary S. Collins, Johannes B. Reitsma, Douglas G. Altman, and Karel G. M. Moons. Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD statement. *Annals of Internal Medicine*, 162(1):55–63, 2015. ISSN 1539-3704. doi: 10.7326/M14-0697.

Jeff Craven. FDA approved more first-in-class drugs, gave more accelerated approvals in 2021 | RAPS. A news article on the accelaration of first-in-class drugs being approved by the FDA, 2022. URL `https://www.raps.org/news-and-articles/news-articles/2022/1/fda-approved-more-first-in-class-drugs-more-with-a`.

Daniel Crevier. *AI: The Tumultuous History of the Search for Artificial Intelligence*. -, 1993a. Pages: 386.

Daniel Crevier. Daniel crevier. AI: The tumultuous history of the search for artificial intelligence. *Journal of the History of the Behavioral Sciences*, 31(3):273–278, 1993b. ISSN 1520-6696. doi:

10.1002/1520-6696(199507)31:3⟨273::AID-JHBS2300310314⟩3.0.CO;2-1. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/1520-6696%28199507%2931%3A3%3C273%3A%3AAID-JHBS2300310314%3E3.0.CO%3B2-1. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/1520-6696%28199507%2931%3A3%3C273%3A%3AAID-JHBS2300310314%3E3.0.CO%3B2-1.

Nello Cristianini and Elisa Ricci. Support vector machines. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 928–932. Springer US, 2008. ISBN 978-0-387-30162-4. doi: 10.1007/978-0-387-30162-4_415. URL https://doi.org/10.1007/978-0-387-30162-4_415.

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802, 2019. ISSN 2405-8440. doi: 10.1016/j.heliyon.2019.e01802. URL https://www.sciencedirect.com/science/article/pii/S2405844018353404.

R. Das, I. Turkoglu, and A. Sengur. Effective diagnosis of heart disease through neural network ensembles. *Expert Systems with Applications*, 36: 7675–7680, 2009.

B.V. Dasarathy and B. V. Sheela. Composite classifier system design: concepts and methodology,. *Proceedings of the IEEE.*, 7(5):708–713., 1979.

Rodrigo Octávio Deliberato, Guilherme Goto Escudero, Lucas Bulgarelli, Ary Serpa Neto, Stephanie Q Ko, Niklas Soderberg Campos, Berke Saat, Edson Amaro, Fabio Silva Lopes, and Alistair EW Johnson. SEVERITAS: An externally validated mortality prediction for critically ill patients in low and middle-income countries. *International Journal of Medical Informatics*, 131:103959, 2009. ISSN 1386-5056. doi: 10.1016/j.ijmedinf.2019.103959. URL https://www.sciencedirect.com/science/article/pii/S1386505619305131.

R. Detrano, A. Jonosi, W. Steinbrunn, M. pfisterer, J-J Schmid, S. Sandhau, K.H. Guppy, S.Lee, and V. Froelicher. International application of new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

Ayon Dey. Machine learning algorithms: A review. -, 7:6, 2016.

Peter Doggart and Megan Rutherford. Randomly Under Sampled Boosted Tree for Predicting Sepsis From Intensive Care Unit Databases. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, September 2019. doi: 10.23919/CinC49843.2019.9005549. ISSN: 2325-8861.

Yiran Dong and Chao-Ying Joanne Peng. Principled missing data methods for researchers. *SpringerPlus*, 2:222, 2013. ISSN 2193-1801. doi: 10.1186/2193-1801-2-222. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3701793/`.

Saptarshi Dutta. Personalized medicine through machine learning, 2021. URL `https://www.analyticsvidhya.com/blog/2021/06/personalized-medicine-through-machine-learning/`.

Muhammed Umar Farrukh, Richard Wainwright, Keeley Crockett, David McLean, and Neil Dagnall. Building actionable personas using machine learning techniques. In *IEEE XPlore*. IEEE, 2022. ISBN 978-1-66548-768-9. URL `https://ieeexplore.ieee.org/xpl/conhome/1811304/all-proceedings`.

Aaron J. Fisher, John D. Medaglia, and Bertus F. Jeronimus. Lack of group-to-individual generalizability is a threat to human subjects research. *Proceedings of the National Academy of Sciences of the United States of America*, 115(27):E6106–E6115, 2018. ISSN 1091-6490. doi: 10.1073/pnas.1711978115.

Jake Frankenfield. Artificial intelligence: What it is and how it is used, 2022a. URL `https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp`.

Jake Frankenfield. Machine learning, 2022b. URL `https://www.investopedia.com/terms/m/machine-learning.asp`.

Thomas R. Frieden. Evidence for health decision making - beyond randomized, controlled trials. *The New England Journal of Medicine*, 377(5):465–475, 2017. ISSN 1533-4406. doi: 10.1056/NEJMra1614394.

David Fumo. Types of machine learning algorithms you should know, 2017. URL `https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861`.

C. A. W. Glas. Missing data. In Penelope Peterson, Eva Baker, and Barry McGaw, editors, *International Encyclopedia of Education (Third Edition)*, pages 283–288. Elsevier, 2010. ISBN 978-0-08-044894-7. doi: 10.1016/B978-0-08-044894-7.01346-4. URL `https://www.sciencedirect.com/science/article/pii/B9780080448947013464`.

J Goodrich. How IBM's deep blue beat world champion chess player garry kasparov - IEEE spectrum, 2021. URL `https://spectrum.ieee.org/how-ibms-deep-blue-beat-world-champion-chess-player-garry-kasparov`.

Karen Grace-Martin. When listwise deletion works for missing data, 2014. URL `https://www.theanalysisfactor.com/when-listwise-deletion-works/`.

Lin Gu, Xiaowei Zhang, Shaodi You, Shen Zhao, Zhenzhong Liu, and Tatsuya Harada. Semi-supervised learning in medical images through graph-embedded random forest. *Frontiers in Neuroinformatics*, 14, 2020. ISSN 1662-5196. URL `https://www.frontiersin.org/articles/10.3389/fninf.2020.601829`.

Kevin Gurney. An introduction to neural networks. In *An Introduction to Neural Networks*. CRC Press, 1997. doi: 10.1201/9781315273570.

Lei Han, Peng Sun, Yali Du, Jiechao Xiong, Qing Wang, Xinghai Sun, Han Liu, and Tong Zhang. Grid-wise control for multi-agent reinforcement learning in video game AI. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2576–2585. PMLR, 2019. URL `https://proceedings.mlr.press/v97/han19a.html`. ISSN: 2640-3498.

Amin Ul Haq, Jian Ping Li, Muhammad Hammad Memon, Shah Nazir, and Ruinan Sun. A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. *Mobile Information Systems*, 2018:e3860146, 2018. ISSN 1574-017X. doi: 10.1155/2018/3860146. URL `https://www.hindawi.com/journals/misy/2018/3860146/`. Publisher: Hindawi.

SHUJUN Huang, NIANGUANG CAI, PEDRO PENZUTI PACHECO, SHAVIRA NARANDES, YANG WANG, and WAYNE XU. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics & Proteomics*, 15(1):41–51, 2017. ISSN 1109-6535. doi: 10.21873/cgp.20063. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5822181/`.

R.A. Hughes, J.Heron, J.A.C. Sterne, and K.Tilling. Accounting for missing data in statistical analyses: multiple imputation is not always the answer. *International Journal of Epidemiology*, 48(4):1294–1304, /08/01 2019. doi: 10.1093/ije/dyz032. URL `https://academic.oup.com/ije/article/48/4/1294/5382162`.

Grand View Research Inc. AI in healthcare market worth $31.3 billion by 2025: Grand view research, inc., 2019. URL `https://www.prnewswire.com/news-releases/ai-in-healthcare-market-worth-31-3-billion-by-2025-grand-view-research-inc-300975059.html`. The global artificial intelligence in healthcare market size introduction.

John P. A. Ioannidis and Muin J. Khoury. Evidence-based medicine and big genomic data. *Human Molecular Genetics*, 27:R2–R7, 2018. ISSN 1460-2083. doi: 10.1093/hmg/ddy065.

Gheorghe Iordanescu, Palamadai N. Venkatasubramanian, and Alice M. Wyrwicz. Automatic segmentation of amyloid plaques in MR images using unsupervised SVM. *Magnetic Resonance in Medicine*, 67(6):1794–1802, 2012. ISSN 0740-3194. doi: 10.1002/mrm.23138. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3311764/`.

Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1):162, 2007. ISSN 1471-2288. doi: 10.1186/s12874-017-0442-1. URL `https://doi.org/10.1186/s12874-017-0442-1`.

J. Larry Jameson, Anthony S. Fauci, Dennis L. Kasper, Stephen L. Hauser, Dan L. Longo, and Joseph Loscalzo. Alzheimer's disease and other dementias. In *Harrison's Manual of Medicine*. McGraw-Hill Education, 20 edition, 2020. URL `accessmedicine.mhmedical.com/content.aspx?aid=1167068989`.

Antoine Jamin, Pierre Abraham, and Anne Humeau-Heurtier. Machine learning for predictive data analytics in medicine: A review illustrated by cardiovascular and nuclear medicine examples. *Clinical Physiology and Functional Imaging*, 41(2):113–127, 2021. ISSN 1475-097X. doi: 10.1111/cpf.12686. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cpf.12686`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cpf.12686.

Mortaza Jamshidian and Matthew Mata. 2 - advances in analysis of mean and covariance structure when data are incomplete**this research was supported in part by the national science foundation grant DMS-0437258. In Sik-Yum Lee, editor, *Handbook of Latent Variable and Related Models*, Handbook of Computing and Statistics with Applications, pages 21–44. North-Holland, 2007. doi: 10.1016/B978-044452044-9/50005-7. URL `https://www.sciencedirect.com/science/article/pii/B9780444520449500057`.

Sai Balasubramanian J.D, M. D. Amazon's growth in healthcare is unparalleled, 2022. URL `https://www.forbes.com/sites/saibala/2022/02/21/amazons-growth-in-healthcare-is-unparalleled/`. Section: Healthcare.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

doi: 10.1098/rsta.2015.0202. URL https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202. Publisher: Royal Society.

Esra Mahsereci Karabulut, Selma Ayşe Özel, and Turgay İbrikçi. A comparative study on the effect of feature selection on classification accuracy. *Procedia Technology*, 1:323–327, 2012. ISSN 2212-0173. doi: 10.1016/j.protcy.2012.02.068. URL https://www.sciencedirect.com/science/article/pii/S2212017312000692.

Britt E. Keuning, Thomas Kaufmann, Renske Wiersema, Anders Granholm, Ville Pettilä, Morten Hylander Møller, Christian Fynbo Christiansen, José Castela Forte, Harold Snieder, Frederik Keus, Rick G. Pleijhuis, Iwan C. C. van der Horst, and Healics Consortium. Mortality prediction models in the adult critically ill: A scoping review. *Acta Anaesthesiologica Scandinavica*, 64(4):424–442, 2020. ISSN 1399-6576. doi: 10.1111/aas.13527. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/aas.13527. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/aas.13527.

S.S. Khan, A.Ahmad, and A.Mihailidis. Bootstrapping and multiple imputation ensemble approaches for missing data. *Journal of Intelligent 'I&' Fuzzy Systems*, 02/01 2018. URL https://arxiv.org/abs/1802.00154v5.

Kirill A. Konovalov, Ilona Christy Unarta, Siqin Cao, Eshani C. Goonetilleke, and Xuhui Huang. Markov state models to study the functional dynamics of proteins in the wake of machine learning. *JACS Au*, 1(9):1330–1341, 2021. ISSN 2691-3704. doi: 10.1021/jacsau.1c00254. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8479766/.

Andrew A. Kramer. Predictive mortality models are not like fine wine. *Critical Care*, 9(6):636, 2005. ISSN 1364-8535. doi: 10.1186/cc3899. URL https://doi.org/10.1186/cc3899.

Ajitesh Kumar. *Hold-out Method for Training Machine Learning Models*, 2022. URL https://vitalflux.com/hold-out-method-for-training-machine-learning-model/.

Carson Lam, Anna Siefkas, Nicole S. Zelin, Gina Barnes, R. Phillip Dellinger, Jean-Louis Vincent, Gregory Braden, Hoyt Burdick, Jana Hoffman, Jacob Calvert, Qingqing Mao, and Ritankar Das. Machine learning as a precision-medicine approach to prescribing COVID-19 pharmacotherapy with remdesivir or corticosteroids. *Clinical Therapeutics*, 43(5):871–885, 2021. ISSN 1879-114X. doi: 10.1016/j.clinthera.2021.03.016.

C. Beulah Christalin Latha and S. Carolin Jeeva. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, 16:100203, 2019. ISSN 2352-9148. doi: 10.1016/j.imu.2019.100203. URL https://www.sciencedirect.com/science/article/pii/S235291481830217X.

J. R. Le Gall, S. Lemeshow, and F. Saulnier. A new simplified acute physiology score (SAPS II) based on a european/north american multicenter study. *JAMA*, 270(24):2957–2963, 1993. ISSN 0098-7484. doi: 10.1001/jama.270.24.2957.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL `https://www.nature.com/articles/nature14539`. Number: 7553 Publisher: Nature Publishing Group.

Charles Chin Han Lew, Gabriel Jun Yung Wong, Chee Keat Tan, and Michelle Miller. Performance of the acute physiology and chronic health evaluation II (APACHE II) in the prediction of hospital mortality in a mixed ICU in singapore. *Proceedings of Singapore Healthcare*, 28(3):147–152, 2019. ISSN 2010-1058. doi: 10.1177/2010105818812896. URL `https://doi.org/10.1177/2010105818812896`. Publisher: SAGE Publications Ltd.

Tanya Lewis. A brief history of artificial intelligence | live science, 2014. URL `https://www.livescience.com/49007-history-of-artificial-intelligence.html`.

Hui Li. Which machine learning algorithm should i use?, 2017. URL `https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/`.

Tian Li. Tilted empirical risk minimization. Section: FATE, 2021. URL `https://blog.ml.cmu.edu/2021/04/02/term/`.

Ying Lin. 10 artificial intelligence statistics you need to know in 2021 [infographic], 2020. URL `https://www.oberlo.com/blog/artificial-intelligence-statistics`.

Alex P. Lind and Peter C. Anderson. Predicting drug activity against cancer cells by random forest models based on minimal genomic information and chemical properties. *PLoS ONE*, 14(7):e0219774, 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0219774. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6622537/`.

R.J. Little and D. Rubin. *Statistical Analysis with Missing Data, Second Edition.* John Wiley & Sons Inc., 2002.

Yuchi Liu, Hailin Shi, Hang Du, Rui Zhu, Jun Wang, Liang Zheng, and Tao Mei. Boosting semi-supervised face recognition with noise robustness. *Computer Vision and Pattern Recognition*, 2021. doi: 10.48550/arXiv.2105.04431. URL `https://arxiv.org/abs/2105.04431v1`.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489. URL `http://ieeexplore.ieee.org/document/1056489/`.

Simon Lyra, Steffen Leonhardt, and Christoph Hoog Antink. Early Prediction of Sepsis Using Random Forest Classification for Imbalanced Clinical Data. In *2019 Computing in Cardiology (CinC)*, pages 1–4, September 2019. doi: 10.23919/CinC49843.2019.9005769. ISSN: 2325-8861.

Laura Martignon. Risk literacy in school. In *Conditions for risk assessment as a topic for probabilistic education*, 2010.

Pamela McCorduck and Cli Cfe. *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence.* CRC Press, 2004. ISBN 978-1-00-006529-9. Google-Books-ID: r2C1DwAAQBAJ.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. ISSN 1522-9602. doi: 10.1007/BF02478259. URL `https://doi.org/10.1007/BF02478259`.

Natalie McLymont and Guy W. Glover. Scoring systems for the characterization of sepsis and associated outcomes. *Annals of Translational Medicine*, 4(24):527, December 2016. ISSN 2305-5839. doi: 10.21037/atm.2016.12.53.

David Menager. Is the turing test a valid test of artificial intelligence?, 2018. URL `https://becominghuman.ai/is-the-turing-test-a-valid-test-of-artificial-intelligence-6695b6e4304`.

Induparkavi Murugesan, Karthikeyan Murugesan, Lingeshwaran Balasubramanian, and Malathi Arumugam. Interpretation of Artificial Intelligence Algorithms in the Prediction of Sepsis. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, September 2019. doi: 10.23919/CinC49843.2019.9005667. ISSN: 2325-8861.

Nature. AI diagnostics need attention. *Nature*, 555(7696):285, 2018. ISSN 1476-4687. doi: 10.1038/d41586-018-03067-x.

T. Nguyen, A. Khosravi, D. Creighton, and et al. Classification of healthcare data using genetic fuzzy systems and wavelets. *Expert Systems with Applications*, 42:2184–2197, 2015.

NHS. Nhs choices, 2015. URL `https://digital.nhs.uk/data-and-information/publications/statistical/hospital-adult-critical-care-activity/2015-16`.

Ziad Obermeyer and Ezekiel J. Emanuel. Predicting the future — big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216–1219, 2016. ISSN 0028-4793. doi: 10.1056/NEJMp1606181. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5070532/`.

Elizabeth M. Painter. Demographic characteristics of persons vaccinated during the first month of the COVID-19 vaccination program — united states, december 14, 2020–january 14, 2021. *MMWR. Morbidity and Mortality Weekly Report*, 70, 2021. ISSN 0149-21951545-861X. doi: 10.15585/mmwr.mm7005e1. URL `https://www.cdc.gov/mmwr/volumes/70/wr/mm7005e1.htm`.

Ravi B. Parikh, Ziad Obermeyer, and Amol S. Navathe. Regulation of predictive analytics in medicine. *Science (New York, N.Y.)*, 363(6429): 810–812, 2019. ISSN 0036-8075. doi: 10.1126/science.aaw0029. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6557272/`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

K.I. Penny and I. Atkinson. Approaches for dealing with missing data in health care studies. *Journal of Clinical Nursing*, 21(19-20):2722–2729, Oct 2012. doi: 10.1111/j.1365-2702.2011.03854.x.

David Petersson. AI vs. machine learning vs. deep learning: Key differences | TechTarget, 2021. URL `https://www.techtarget.com/searchenterpriseai/tip/AI-vs-machine-learning-vs-deep-learning-Key-differences`.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00116251. URL `http://link.springer.com/10.1007/BF00116251`.

Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019. ISSN 0028-4793. doi: 10.1056/NEJMra1814259. URL `https://www.nejm.org/doi/full/10.1056/NEJMra1814259`. Publisher: Massachusetts Medical Society _eprint: https://www.nejm.org/doi/pdf/10.1056/NEJMra1814259.

Matthew A. Reyna, Christopher S. Josef, Russell Jeter, Supreeth P. Shashikumar, M. Brandon Westover, Shamim Nemati, Gari D. Clifford, and Ashish Sharma. Early Prediction of Sepsis From Clinical

Data: The PhysioNet/Computing in Cardiology Challenge 2019. *Critical Care Medicine*, 48(2):210–217, February 2020. ISSN 0090-3493. doi: 10.1097/CCM.0000000000004145. URL `http://journals.lww.com/10.1097/CCM.0000000000004145`.

Frank Rosenblatt. Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309, 1960. ISSN 2162-6634. doi: 10.1109/JRPROC.1960.287598. Conference Name: Proceedings of the IRE.

Margherita Rosnati and Vincent Fortuin. MGP-AttTCN: An interpretable machine learning model for the prediction of sepsis. *PloS One*, 16(5): e0251248, 2021. ISSN 1932-6203. doi: 10.1371/journal.pone.0251248.

M Rouse and B Botelho. What is cognitive computing? - definition from WhatIs.com, 2018. URL `https://www.techtarget.com/searchenterpriseai/definition/cognitive-computing`.

Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons, 2004. ISBN 978-0-471-65574-9. Google-Books-ID: bQBtw6rx_mUC.

A. Sabay, L.Harris, V.Bejugama, and K. Jaceldo-Siegl. Overcoming small data limitations in heart disease prediction by using surrogate data. *SMU Data Science Review*, 1(3), 2018.

Y. Sakr, C. Krauss, A. C. K. B. Amaral, A. Réa-Neto, M. Specht, K. Reinhart, and G. Marx. Comparison of the performance of SAPS II, SAPS 3, APACHE II, and their customized prognostic models in a surgical intensive care unit. *British Journal of Anaesthesia*, 101(6):798–803, 2008. ISSN 0007-0912, 1471-6771. doi: 10.1093/bja/aen291. URL `https://www.bjanaesthesia.org/article/S0007-0912(17)34100-4/fulltext`. Publisher: Elsevier.

Ali A. Samir, Abdullah R. Rashwan, Karam M. Sallam, Ripon K. Chakrabortty, Michael J. Ryan, and Amr A. Abohany. Evolutionary algorithm-based convolutional neural network for predicting heart diseases. *Computers & Industrial Engineering*, 161:107651, November 2021. ISSN 0360-8352. doi: 10.1016/j.cie.2021.107651. URL `https://www.sciencedirect.com/science/article/pii/S0360835221005556`.

Ian Sample and Alex Hern. Scientists dispute whether computer 'eugene goostman' passed turing test. *The Guardian*, 2014. ISSN 0261-3077. URL `https://www.theguardian.com/technology/2014/jun/09/scientists-disagree-over-whether-turing-test-has-been-passed`.

A. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 1959. doi: 10.1147/rd.33.0210.

Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):160, 2021. ISSN 2661-8907. doi: 10.1007/s42979-021-00592-x. URL https://doi.org/10.1007/s42979-021-00592-x.

Raniya R. Sarra, Ahmed M. Dinar, Mazin Abed Mohammed, and Karrar Hameed Abdulkareem. Enhanced heart disease prediction based on machine learning and x^2 statistical optimal feature selection model. *Designs*, 6(5):87, 2022. ISSN 2411-9660. doi: 10.3390/designs6050087. URL https://www.mdpi.com/2411-9660/6/5/87. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Charlie Schmidt. M. d. anderson breaks with IBM watson, raising questions about artificial intelligence in oncology. *Journal of the National Cancer Institute*, 109(5), 2017. ISSN 1460-2105. doi: 10.1093/jnci/djx113.

Nicholas J. Schork. Randomized clinical trials and personalized medicine: A commentary on deaton and cartwright. *Social Science & Medicine (1982)*, 210:71–73, 2018. ISSN 1873-5347. doi: 10.1016/j.socscimed.2018.04.033.

Sebastian Schuchmann. History of the first AI winter, 2019. URL https://towardsdatascience.com/history-of-the-first-ai-winter-6f8c2186f80b.

Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. RUSBoost: Improving classification performance when training data is skewed. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008. doi: 10.1109/ICPR.2008.4761297. ISSN: 1051-4651.

Tawseef Shaikh and Rashid Ali. *Applying Machine Learning Algorithms for Early Diagnosis and Prediction of Breast Cancer Risk: ICCCN 2018, NITTTR Chandigarh, India*, pages 589–598. -, 01 2019. ISBN 978-981-13-1216-8. doi: 10.1007/978-981-13-1217-5_57.

Supreeth P Shashikumar, Christopher Josef, Ashish Sharma, and Shamim Nemati. DeepAISE - an end-to-end development and deployment of a recurrent neural survival model for early prediction of sepsis. *unpublished*, 2019.

Alex Shenfield, Marcos Rodrigues, Hossam Nooreldeen, and Jeronimo Moreno-Cuesta. A novel hybrid differential evolution strategy applied to classifier design for mortality prediction in adult critical care admissions. In *2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2017. ISBN 978-1-4673-8988-4. doi: 10.1109/CIBCB.2017.8058544. URL http://ieeexplore.ieee.org/document/8058544/.

Vardhan Shorewala. Early detection of coronary heart disease using ensemble techniques. *Informatics in Medicine Unlocked*, 26:100655, 2021. ISSN 2352-9148. doi: https://doi.org/10.1016/j.imu.2021.100655. URL https://www.sciencedirect.com/science/article/pii/S235291482100143X.

Graham Singer. The history of the modern graphics processor. A description of GPUs developments to date., 2022. URL https://www.techspot.com/article/650-history-of-the-gpu/.

Seema Singh. Cousins of artificial intelligence, 2018a. URL https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55.

Seema Singh. *An Introduction To Clustering*, 2018b. URL https://medium.datadriveninvestor.com/an-introduction-to-clustering-61f6930e3e0b.

Chris Smith. The history of artificial intelligence. *N/A*, page 27, 2006.

V. Smolyakov. Ensemble learning to improve machine learning results, -03-07T14:56:13.538Z 2019. URL https://blog.statsbot.co/ensemble-learning-d1dcd548e936.

Luis R. Soenksen, Timothy Kassis, Susan T. Conover, Berta Marti-Fuster, Judith S. Birkenfeld, Jason Tucker-Schwartz, Asif Naseem, Robert R. Stavert, Caroline C. Kim, Maryanne M. Senna, José Avilés-Izquierdo, James J. Collins, Regina Barzilay, and Martha L. Gray. Using deep learning for dermatologist-level detection of suspicious pigmented skin lesions from wide-field images. *Science Translational Medicine*, 13(581):eabb3652, 2021. doi: 10.1126/scitranslmed.abb3652. URL https://www.science.org/doi/10.1126/scitranslmed.abb3652. Publisher: American Association for the Advancement of Science.

Jonathan A C Sterne, Ian R White, John B Carlin, Michael Spratt, Patrick Royston, Michael G Kenward, Angela M Wood, and James R Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ*, 338, 2009. ISSN 0959-8138. doi: 10.1136/bmj.b2393. URL https://www.bmj.com/content/338/bmj.b2393.

Matthew Syed. *Black Box Thinking: Marginal Gains and the Secrets of High Performance*. John Murray, 1st edition edition, 2016. ISBN 978-1-4736-1380-5.

Phil Taylor. Merck will assess quris' AI 'patient-on-a-chip' drug safety tech -, 2022. URL https://pharmaphorum.com/news/merck-will-assess-quris-ai-patient-on-a-chip-drug-safety/.

Melanoma UK Editoral Team. 2020 MELANOMA SKIN CANCER REPORT, 2020. URL `https://www.melanomauk.org.uk/2020-melanoma-skin-cancer-report`. A brief introduction into melanoma skin cancer.

The American Cancer Society Medical Editoral Team. Melanoma skin cancer statistics, 2022. URL `https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html`.

Robert D. Terry. Neuropathological changes in alzheimer disease. In Lars Svennerholm, Arthur K. Asbury, Ralph A. Reisfeld, Konrad Sandhoff, Kunihiko Suzuki, Guido Tettamanti, and Gino Toffano, editors, *Progress in Brain Research*, volume 101 of *Biological Function of Gangliosides*, pages 383–390. Elsevier, 1994. doi: 10.1016/S0079-6123(08)61964-0. URL `https://www.sciencedirect.com/science/article/pii/S0079612308619640`.

Eric Topol. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, illustrated edition edition, 2019. ISBN 978-1-5416-4463-2.

N. Townsend. Cardiovascular disease statistics. *British Heart foundation centre on population approaches for non communicable disease prevention, Oxford 2014*, 2014.

Lucas R. Trambaiolli, Ana C. Lorena, Francisco J. Fraga, Paulo A.M. Kanda, Renato Anghinah, and Ricardo Nitrini. Improving alzheimer's disease diagnosis with machine learning techniques. *Clinical EEG and Neuroscience*, 42(3):160–165, 2011. ISSN 1550-0594, 2169-5202. doi: 10.1177/155005941104200304. URL `http://journals.sagepub.com/doi/10.1177/155005941104200304`.

C.T. Tran, M.Zhang, P.Andreae, B.Xue, and L.T. Bui. Multiple imputation and ensemble learning for classification with incomplete data. *Intelligent and Evolutionary Systems*, 2016. URL `https://www.springerprofessional.de/en/multiple-imputation-and-ensemble-learning-for-classification-wit/11030076`.

Luan Tran, Manh Nguyen, and Cyrus Shahabi. Representation Learning for Early Sepsis Prediction. In *2019 Computing in Cardiology (CinC)*, pages 1–4, September 2019. doi: 10.23919/CinC49843.2019.9005565. ISSN: 2325-8861.

My Chau Tu, Dongil Shin, and DongKyoo Shin. Effective diagnosis of heart disease through bagging approach. In *2009 2nd International Conference on Biomedical Engineering and Informatics*, pages 1–4, 2009. doi: 10.1109/BMEI.2009.5301650. ISSN: 1948-2922.

Shreshth Tuli, Nipam Basumatary, Sukhpal Singh Gill, Mohsen Kahani, Rajesh Chand Arya, Gurpreet Singh Wander, and Rajkumar Buyya. HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. *Future Generation Computer Systems*, 104:187–200, 2020. ISSN 0167-739X. doi: 10.1016/j.future.2019.10.043. URL `https://www.sciencedirect.com/science/article/pii/S0167739X19313391`.

A. M. Turing. Computing machinery and intelligence. *Mind, New Series*, 59 (236):433–460, 1950. URL `http://www.jstor.org/stable/2251299`.

Anannya Uberoi. Introduction to dimensionality reduction, 2017. URL `https://www.geeksforgeeks.org/dimensionality-reduction/`. Section: Machine Learning.

Laurence Vermeer and Mark Thomas. Pharmaceutical/high-tech alliances; transforming healthcare? digitalization in the healthcare industry. *Strategic Direction*, 36(12):43–46, 2020. ISSN 0258-0543. doi: 10.1108/SD-06-2020-0113. URL `https://doi.org/10.1108/SD-06-2020-0113`. Publisher: Emerald Publishing Limited.

Tomas Vicar, Petra Novotna, Jakub Hejc, Marina Ronzhina, and Radovan Smisek. Sepsis Detection in Sparse Clinical Data Using Long Short-Term Memory Network with Dice Loss. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, September 2019. doi: 10.23919/CinC49843.2019.9005786. ISSN: 2325-8861.

J. Vijayashree and H. Parveen Sultana. A machine learning framework for feature selection in heart disease classification using improved particle swarm optimization with support vector machine classifier. *Programming and Computer Software*, 44(6):388–397, 2018. ISSN 1608-3261. doi: 10.1134/S0361768818060129. URL `https://doi.org/10.1134/S0361768818060129`.

Douglas P. Wagner and Elizabeth A. Draper. Acute physiology and chronic health evaluation (APACHE II) and medicare reimbursement. *Health Care Financing Review*, 1984:91–105, 1984. ISSN 0195-8631. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4195105/`.

Richard Wainwright and Alex Shenfield. Human Activity Recognition Making Use of Long Short-Term Memory Techniques. *ATHENS JOURNAL OF SCIENCES*, 6(1):19–34, February 2019. ISSN 22418466. doi: 10.30958/ajs.6-1-2. URL `https://www.athensjournals.gr/sciences/2019-6-1-2-Wainwright.pdf`.

Richard Wainwright and Alex Shenfield. Machine learning for mortality risk prediction with changing patient demographics. *20th IEEE Conference on*

Computational Intelligence in Bioinformatics and Computational Biology, 2023.

Katrina Wakefield. A guide to the types of machine learning algorithms, 2022. URL https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html.

Shalika Walker, Waqas Khan, Katarina Katic, Wim Maassen, and Wim Zeiler. Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. *Energy and Buildings*, 209:109705, 2020. ISSN 0378-7788. doi: 10.1016/j.enbuild.2019.109705. URL https://www.sciencedirect.com/science/article/pii/S0378778819319139.

Q Wang. How to apply AI in project management, 2019. URL https://pmworldlibrary.net/wp-content/uploads/2019/03/pmwj80-Apr2019-Wang-how-to-aply-AI-in-project-management.pdf.

Xu Wang and Yusheng Xu. An improved index for clustering validation based on silhouette index and calinski-harabasz index. *IOP Conference Series: Materials Science and Engineering*, 569(5):052024, 2019. ISSN 1757-8981, 1757-899X. doi: 10.1088/1757-899X/569/5/052024. URL https://iopscience.iop.org/article/10.1088/1757-899X/569/5/052024.

Kevin Warwick and Huma Shah. Can machines think? a report on turing test experiments at the royal society. *Journal of Experimental & Theoretical Artificial Intelligence*, 28, 2015. doi: 10.1080/0952813X.2015.1055826.

Eric A. Weiss. Turing award winners. In *Encyclopedia of Computer Science*, pages 1795–1797. John Wiley and Sons Ltd., 2003. ISBN 978-0-470-86412-8.

B.J. Wells, K.M. Chagin, A.S. Nowacki, and M.W. Kattan. Strategies for handling missing data in electronic health record derived data. *EGEMS (Washington, DC)*, 1(3):1035, 2013. doi: 10.13063/2327-9214.1035.

Junwei Xiao, Jianfeng Lu, and Xiangyu Li. Davies bouldin index based hierarchical initialization k-means. *Intelligent Data Analysis*, 21(6):1327–1338, 2017. ISSN 1088-467X. doi: 10.3233/IDA-163129. URL https://content.iospress.com/articles/intelligent-data-analysis/ida163129. Publisher: IOS Press.

Christopher R Yee, Niven R Narain, Viatcheslav R Akmaev, and Vijetha Vemulapalli. A data-driven approach to predicting septic shock in the intensive care unit. *Biomedical Informatics Insights*, 11:1178222619885147, 2019. ISSN 1178-2226. doi: 10.1177/1178222619885147. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6829643/.

Chang Ho Yoon, Robert Torrance, and Naomi Scheinerman. Machine learning in medicine: should the pursuit of enhanced interpretability be abandoned? *Journal of Medical Ethics*, 2021. ISSN 0306-6800, 1473-4257. doi: 10.1136/medethics-2020-107102. URL `https://jme.bmj.com/content/early/2021/05/18/medethics-2020-107102`. Publisher: Institute of Medical Ethics Section: Clinical ethics.

Ke Yu, Mingda Zhang, Tianyi Cui, and Milos Hauskrecht. Monitoring ICU mortality risk with a long short-term memory recurrent neural network. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 25:103–114, 2020. ISSN 2335-6928. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6934094/`.

Kun-Hsing Yu and Isaac S. Kohane. Framing the challenges of artificial intelligence in medicine. *BMJ quality & safety*, 28(3):238–241, 2019. ISSN 2044-5423. doi: 10.1136/bmjqs-2018-008551.

Morteza Zabihi, Serkan Kiranyaz, and Moncef Gabbouj. Sepsis Prediction in Intensive Care Unit Using Ensemble of XGboost Models. In *2019 Computing in Cardiology (CinC)*, pages Page 1–Page 4, September 2019. doi: 10.23919/CinC49843.2019.9005564. ISSN: 2325-8861.

Xinhua Zhang. Structural risk minimization. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 929–930. Springer US, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_793. URL `https://doi.org/10.1007/978-0-387-30164-8_793`.

Na Zhou, Chuan-Tao Zhang, Hong-Ying Lv, Chen-Xing Hao, Tian-Jun Li, Jing-Juan Zhu, Hua Zhu, Man Jiang, Ke-Wei Liu, He-Lei Hou, Dong Liu, Ai-Qin Li, Guo-Qing Zhang, Zi-Bin Tian, and Xiao-Chun Zhang. Concordance study between IBM watson for oncology and clinical practice for patients with cancer in china. *The Oncologist*, 24(6):812–819, 2019. ISSN 1549-490X. doi: 10.1634/theoncologist.2018-0255.

Xiao-Yun Zhou, Yao Guo, Mali Shen, and Guang-Zhong Yang. Application of artificial intelligence in surgery. *Frontiers of Medicine*, 14(4):417–430, 2020. ISSN 2095-0225. doi: 10.1007/s11684-020-0770-0. URL `https://doi.org/10.1007/s11684-020-0770-0`.

Min Zhu, Jing Xia, Molei Yan, Guolong Cai, Jing Yan, and Gangmin Ning. Dimensionality reduction in complex medical data: Improved self-adaptive niche genetic algorithm. *Computational and Mathematical Methods in Medicine*, 2015:e794586, 2015. ISSN 1748-670X. doi: 10.1155/2015/794586. URL `https://www.hindawi.com/journals/cmmm/2015/794586/`. Publisher: Hindawi.