Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction

AKINJOLE, Abisola, SHOBAYO, Olamilekan <http://orcid.org/0000-0001-5889-7082>, POPOOLA, Jumoke, OKOYEIGBO, Obinna and OGUNLEYE, Bayode

mathematics

Article

Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction

Abisola Akinjole [1], Olamilekan Shobayo [1,*], Jumoke Popoola [1], Obinna Okoyeigbo [2] and Bayode Ogunleye [3]

[1] School of Computing and Digital Technologies, Sheffield Hallam University, Sheffield S1 2NU, UK; abisola.j.akinjole@student.shu.ac.uk (A.A.); j.popoola@shu.ac.uk (J.P.)

[2] Department of Engineering, Edge Hill University, Ormskirk L39 4QP, UK; obinna.okoyeigbo@edgehill.ac.uk

[3] Department of Computing & Mathematics, University of Brighton, Brighton BN2 4GJ, UK; b.ogunleye@brighton.ac.uk

* Correspondence: o.shobayo@shu.ac.uk

Abstract: Predicting credit default risk is important to financial institutions, as accurately predicting the likelihood of a borrower defaulting on their loans will help to reduce financial losses, thereby maintaining profitability and stability. Although machine learning models have been used in assessing large applications with complex attributes for these predictions, there is still a need to identify the most effective techniques for the model development process, including the technique to address the issue of data imbalance. In this research, we conducted a comparative analysis of random forest, decision tree, SVMs (Support Vector Machines), XGBoost (Extreme Gradient Boosting), ADABoost (Adaptive Boosting) and the multi-layered perceptron, to predict credit defaults using loan data from LendingClub. Additionally, XGBoost was used as a framework for testing and evaluating various techniques. Moreover, we applied this XGBoost framework to handle the issue of class imbalance observed, by testing various resampling methods such as Random Over-Sampling (ROS), the Synthetic Minority Over-Sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Random Under-Sampling (RUS), and hybrid approaches like the SMOTE with Tomek Links and the SMOTE with Edited Nearest Neighbours (SMOTE + ENNs). The results showed that balanced datasets significantly outperformed the imbalanced dataset, with the SMOTE + ENNs delivering the best overall performance, achieving an accuracy of 90.49%, a precision of 94.61% and a recall of 92.02%. Furthermore, ensemble methods such as voting and stacking were employed to enhance performance further. Our proposed model achieved an accuracy of 93.7%, a precision of 95.6% and a recall of 95.5%, which shows the potential of ensemble methods in improving credit default predictions and can provide lending platforms with the tool to reduce default rates and financial losses. In conclusion, the findings from this study have broader implications for financial institutions, offering a robust approach to risk assessment beyond the LendingClub dataset.

Keywords: credit default prediction; deep learning; ensemble learning; machine learning

MSC: 68T09

Citation: Akinjole, A.; Shobayo, O.; Popoola, J.; Okoyeigbo, O.; Ogunleye, B. Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction. *Mathematics* **2024**, *12*, 3423. https://doi.org/10.3390/math12213423

Academic Editors: Raymond Lee, Xinan Yang and Dong Li

Received: 10 September 2024
Revised: 3 October 2024
Accepted: 30 October 2024
Published: 31 October 2024

1. Introduction

Numerous financial institutions, such as banks and lending platforms, have relied on the interest and fees from loans as a source of revenue [1]. To maintain financial strength and profitability, these institutions must ensure that borrowers do not default on their payments. This objective becomes crucial when considering past financial crises, such as the economic breakdown in the late 2000s, where lending to individuals or businesses unable to repay their debts contributed significantly to the crises [2,3]. Predicting credit default risk, defined as the likelihood that a borrower will fail to fulfil their loan obligations, remains an important challenge for lenders, helping them avoid large losses and maintain public trust [4–7].

However, traditional rule-based systems, when presented with complex features and data, often struggle, which may be less accurate and may not produce reliable predictions [8–10]. Recent developments in machine learning and deep learning have shown promise in improving predictive accuracy, as they have significantly changed how loan applications are assessed, allowing lenders to make use of the characteristics of borrowers, such as age, employment status, length of employment, income, etc., to determine which of the loans will be fully paid or defaulted on [11]. Additionally, these methods can handle and process more data and uncover patterns that are difficult and that may be missed by expert analysts or rule-based systems [12–14]. Yet, challenges remain. The presence of class imbalance in credit default datasets and the diversity of techniques for handling outliers, normalisation, feature selection and model development create significant complications in building an effective model, as choosing a wrong or suboptimal technique can distort the data and reduce the efficiency of the predictive model [15–18].

In this paper, we address these complications by testing various methods to identify the optimal approach at each stage of the model development process. Our study systematically evaluates techniques for handling outliers, normalising data, splitting the dataset, balancing class distribution and selecting the most important feature. We also propose an ensemble model that combines machine learning and deep learning techniques with boosting algorithms such as Extreme Gradient Boosting (XGBoost) and Adaptive Boosting (ADABoost). Furthermore, by comparing class imbalance techniques like Random Over-Sampling (ROS), Random Under-Sampling (RUS), the Synthetic Minority Over-Sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Tomek Links, SMOTE-Tomek and SMOTE with the Edited Nearest Neighbours (SMOTE + ENNs) and assessing model performance with multiple metrics, especially recall, we provide a comprehensive solution for improving credit default risk predictions.

In summary, the main contributions of our research are as follows:

- We introduced a novel approach by leveraging XGBoost to make decisions at various stages of the model development process, which allowed us to select and use optimal techniques.
- We compared different class imbalance techniques, analysing their impact on model performance metrics.
- We proposed an ensemble model that enhances predictive accuracy over traditional machine learning models by combining boosting classifiers with machine learning and deep learning techniques.

Unlike prior studies that focus on a single aspect of the model development process, this research integrates techniques across multiple stages, resulting in a robust and scalable solution for credit default risk prediction. This study is arranged as follows: with the subsequent section exploring existing techniques and methodologies related to credit default prediction (Section 2), followed by the methodology made use of in this study (Section 3), then the results obtained (Section 4) and, finally, the conclusion and recommendations (Section 5).

2. Related Works

Loan default prediction is crucial for financial institutions to minimise losses, as default rates and profitability are highly correlated [19]; therefore, developing models that accurately predict loan defaults has become essential, with machine learning techniques increasingly being leveraged due to their significant improvement in predictability across various financial applications [20–22]. Several models, including logistic regression, random forest, decision tree, the Support Vector Machine (SVM), the Multilayer Perceptron (MLP), XGBoost and ADABoost have been widely used in credit default prediction. However, many studies have not adequately addressed the challenge of class imbalance, which can lead to biased models favouring the majority class (non-defaulters). Moreover, there has been a limited exploration of hybrid approaches that ensemble predictions from deep learning and machine learning models, particularly when combined with boosting classifiers, which as noted by authors in [23–25], are known to improve model robustness. Additionally, some studies have relied heavily on accuracy as the evaluation metric, which is insufficient in imbalanced datasets where the models may neglect the minority class (defaulters) altogether.

2.1. Supervised Learning Methods in the Prediction of Credit Default

Selecting the appropriate model to use depends on the type of tasks or problem that needs to be solved [12]. In their research, refs. [26,27] categorised various learning methods, showing that supervised learning methods are better for solving classification tasks (yes/no or true/false) and regression (predicting continuous values), while unsupervised learning methods are better suited for clustering (grouping data) and dimensionality reduction. Given that credit default prediction typically uses borrowers' characteristics as inputs and binary classification as the target variable, unsupervised learning methods are not often considered in this context.

In the use of supervised machine learning models, ref. [20] compared the performance of SVM and logistic regression models for the prediction of credit default using data from the portfolio of a Portuguese bank. The dataset (1992 non-defaulting customers and 1008 defaulting customers) was split into 75% for model training and 25% for the validation. While the SVM achieved a strong Receiver Operating Characteristic (ROC) score of 98% compared to logistic regression's 73%, the study noted the difficulty of selecting optimal parameter values for the SVM kernels. This current study addresses that difficulty by employing a grid search to identify optimal parameter values. Furthermore, the dataset size raises concerns about the generalisability of the model's performance to larger, more diverse credit portfolios.

Ref. [1] used a LendingClub dataset from 2007 to 2015, employing 70:30 training and a validation split to develop random forest and decision tree models. Similar to this current study, columns with null values and variables with strong correlation with other dependent variables (multicollinearity) were removed, and, additionally, to evaluate the performance of the models, the accuracy score was used. Although the study observed that random forest performed better than decision tree with 80% accuracy and 73% accuracy, respectively, this was the only major metric used in the evaluation, and, since the class imbalance was not handled, accuracy may not be the best metric to use, as the models will be biased to the non-defaulters. This study addresses this limitation by handling the class imbalance issue and using more evaluation metrics in the assessment of the models.

Ref. [28] employed a Light Gradient Boosting Machine (LightGBM) and XGBoost for the prediction of loan default using LendingClub data from July 2007 to June 2017. The study had an interesting approach to cleaning the data, as two separate cleaning processes, multi-observational and multi-dimensional methods, were used to identify and correct inconsistencies. The dataset was then randomly split into training and test sets, with 91.2% of the data used for training and 8.8% for validation in the multi-observational method, while the multi-dimensional method used 95.8% for training and 4.2% for validation. Both methods were used to develop the LightGBM and XGBoost, which are based on the Gradient Boosting Decision Tree (GBDT) and are known for efficiently dealing with massive and high-dimensional data. For XGBoost, the multi-observational method achieved an accuracy of 80.06% with an error rate of 19.94%, while the multi-dimensional method attained an accuracy of 79.9% with an error rate of 20.0%. For the LightGBM, the multi-dimensional method recorded 80.06% accuracy and 19.94% error, and the multi-observational method achieved 80.1% accuracy with a 19.9% error rate, ultimately concluding that the LightGBM slightly outperformed XGBoost in predicting loan defaults. This current study builds upon this approach, by balancing the data and testing diverse techniques to find the optimal solution.

In [29], XGBoost was proposed to build credit risk assessment models using data from a financial institution in Taiwan over an eight-year period between 2009 and 2016. Missing values were removed; additionally, outliers were handled using the Interquartile range (IQR) method. The study observed that most of the data used for credit scoring was imbalanced; therefore, they used the cluster-based under-sampling method to process the imbalanced data, testing various ratios to sample the dataset. The balanced data were split using an 80:20 ratio and applied to the models; furthermore, accuracy and the Area Under the ROC Curve (AUC) were used as validation metrics, as the proposed model was compared with other models, including logistic regression and the SVM. The authors observed that XGBoost outperformed the other models with an accuracy of 90% against 70% and 77% accuracy scores for logistic regression and the SVM, respectively, and AUC values of 94% against 77% and 87%, respectively. Although the study achieved impressive results with the XGBoost model, the relatively small dataset of 6271 records presents a potential limitation for generalising these results to larger datasets. Additionally, even though the authors addressed the class imbalance issue, they focused only on using cluster-based under-sampling, without considering other techniques that might be more effective or suitable.

Deep learning is another method that has been crucial in the prediction of credit or loan defaults. Originating from Artificial Neural Networks (ANNs), it uses a multilayered neural network and processing to imitate the complexity of the human brain in decision making [30,31]. In [32], the authors made use of a deep learning model to predict consumer loan defaults using a dataset with 1000 observations obtained from responses to a questionnaire created by the authors. This study used Keras, a neural network library that runs on TensorFlow. Although this research made use of a deep learning model in the prediction of bad loans, it is not directly comparable to this current study, given the mode of data collection, which involved selecting eleven top banks and distributing a survey to only participants who had taken out loans, which is significantly different from the dataset used in this current study. However, similar to this current study, ref. [32] employed stratified random sampling.

The assessment and prediction of lending risk using an MLP with three hidden layers was presented by [33] with the LendingClub dataset from the period of 2007 to 2015 for model development and evaluation. To handle categorical features, one-hot encoding method was used to convert the features to numerical values. Additionally, the output variable was classified into three categories using TensorFlow: safe loans, risky loans and bad loans, with a majority of the data belonging to safe loans. The class imbalance issue was handled using the SMOTE, with 80% of the data used in training. Furthermore, accuracy served as the measure of the model's performance when compared with other models. The MLP, with an accuracy of 93.2%, outperformed other models, including logistic regression (77.1%), decision tree (50.5%), the linear SVM (78.9%), ADABoost (85.2%) and the MLP with one hidden layer (62.8%). In that study, no under-sampling or hybrid method was used to handle class imbalance. This current study explores this topic by using various methods to handle class imbalance.

2.2. Combining Predictions Using Ensemble Learning Techniques for the Prediction of Credit Default

Ensemble learning is a branch of machine learning where multiple learners (models) are trained to solve the same problem [34,35]. Instead of using a single model, ensemble learning combines the output of multiple models to obtain better predictions [23]. The primary idea behind ensemble learning is that the combination of these models can perform better than any of the individual models alone. According to [35], there are two steps involved in ensemble learning. The first step is to build different models, while the second step involves combining predictions from the models. Commonly used methods include voting, stacking, bagging, and boosting. These methods create different models by manipulating the training data and model outputs to improve the performance and obtain better predictions [25,35,36].

Various studies employed these methods when predicting credit default risks. [37] used different ensemble approaches, including the bagging and stacking methods to assemble various models, including the SVM in the evaluation of credit risks. The authors used a British credit card application approval dataset. The dataset contained 1225 borrowers' detailed information, including 323 defaults. To balance the data, the authors oversampled the defaults by duplicating each case twice, which increased the number of defaults to 969. The data were further split. A total of 80% was used to develop a multiagent SVM-based ensemble learning system across different stages. Additionally, they employed GridSearch Cross-Validation (GridSearchCV) to determine the parameters of the SVM and the kernel and then developed several ensemble approaches using weighted averaging. They further compared the performance with a quadratic discriminant analysis, linear discriminant analysis, logit regression, a Feed-Forward Neural Network (FNN) and an SVM model, using total accuracy, Type I and Type II error rate and the total accuracy for evaluation. The authors noted that the weight averaging approach combining Adaptive Linear Neural Network-based (ALNN) models outperformed the other models; however, the SVM-based multiagent approach outperformed the ALNN model. While this study explored different ensemble learning methods and kernel diversities, their research was focused on SVM-based approaches alone. In contrast, this current study explores various models, including the SVM. Additionally, this study uses a larger dataset and tests various sampling methods, unlike the approach [37] used in handling the class imbalance.

In the use of ensemble methods, ref. [22] used logistic regression and MLP models to predict credit default, randomly using the 70:30 ratio to split the LendingClub data from 2011 to 2013. The Gini coefficient was used for feature selection, as it measures the separation capability of the model. The authors subsequently combined the models with two ensemble techniques. The first method involved averaging the probabilities from both models to obtain the final predictions, while the second method used logistic regression as a meta-model in a stacking ensemble, taking the output probabilities from both models as input. The first ensemble method performed better than all the other models with an accuracy of 84.1% and an AUC of 67.3%, while the MLP model had an accuracy of 76.14% and an AUC of 67.27%. The error ratio also reflected this trend, with the first ensemble method yielding a lower error ratio of 15.89%, compared to 21.29% for logistic regression and 23.86% for the MLP. While [22]'s study focused on traditional and ensemble methods, this current study differs, given that boosting classifiers are also ensembled. Additionally, the issue of class imbalance is addressed.

The issue of class imbalance cannot be overemphasised when it comes to loan datasets, as the model will always be biased toward the non-defaulters if not properly handled. In [38], the authors made use of diverse over-sampling and under-sampling techniques for the prediction of credit card default. Additionally, they used two ensemble methods, bagging and stacking, as well as K-Nearest Neighbour (KNN), random forest, Logistic Model Trees (LMTs) and Gradient Boosted Decision Tree (GBDT) model. Moreover, three datasets—a Taiwan client credit dataset with 30,000 observations and 6636 defaults, a South-German client credit dataset with 1000 observations and 300 defaults and, lastly, a Belgium client credit dataset with 284,299 observations (492 frauds) from September 2013—were used to build the models. Class imbalance was handled using near miss, cluster centroid and random under-sampling methods. Additionally, Adaptive Synthetic Sampling (ADASYN), the SMOTE, the k-means SMOTE, the borderline SMOTE, SMOTE Tomek and the random over-sampling method were tested. The data were balanced after using a Min-Max scaler to normalise the numerical features. The balanced data were then split using the 70:30 ratio to train and test the models. Their study found that over-sampling techniques, particularly SMOTE combined with the GBDT, outperformed the others in terms of accuracy (82.5%), precision (82.0%), recall (81.8%) and AUC (89.0%). Although this current study tests various sampling techniques as performed by [38], it is different, as it identifies the best method to normalise the data and further ensembles the boosting classifiers with other traditional machine learning models, as well as with the MLP (three hidden layers).

Model Optimisation Methods

In [39], the SMOTE was applied to balance the data used to build a smart application for loan approval prediction. The data used were from Kaggle repository and contained 806 observations and 12 features. Missing values and outliers were handled; moreover, the data were normalised and important features selected, with 75% of the data used in training logistic regression, decision tree, random forest, the SVM, the KNN, Gaussian naïve bayes, ADABoost, dense neural networks, long short-term memory and recurrent neural networks, measuring their performance with accuracy, precision, recall and F1-score. Similar to this current study, the voting approach was used to combine the models by taking two approaches: firstly, combining the predictions from all the models and also combining three of the best performing models. Ref. [39] observed that the deep learning models were less effective when dealing with loan datasets compared to the traditional machine learning models, with the second approach outperforming the other models. Although [39] handled class imbalance, this current study tested other sampling techniques and used more data for the prediction. Additionally, other techniques were explored to improve the models' performance, similar to [40,41], as feature selection techniques were used to optimise the models for credit default risk predictions. Ref. [40] used features extracted from convolution neural networks, as well as Pearson correlation and Recursive Feature Elimination (RFE) to select the best features to build a deep learning-optimised stacking model to predict joint loan risk, concluding that feature selection played a big part in the performance of the final stacking model, with a 6% increase in joint loan approval. Conversely, [41] used only RFE to select the features used to develop fused logistic regression, random forest and Categorical Boosting (CatBoost) models using the blended method. Additionally, they balanced the loan dataset using ADASYN. Furthermore, the authors highlighted the impact of feature selection, with the fused model performing better than the individual models when evaluated on accuracy, recall and F1-score.

Few studies performed hyperparameter tuning using GridSearchCV. In this regard, ref. [42] used GridSearchCV to obtain the parameters to build the MLP, logistic regression, random forest, the SVM, decision tree, XGBoost, LightGBM and a 2-layered neural network for credit risk prediction, with XGBoost also serving as the model used to test the class balancing method, as well as to obtain the feature importance within the model. Additionally, to deal with class imbalance, the authors randomly sampled the default loans and non-default loans, thereby under-sampling the data. Accuracy, recall, precision and F1-score served as the performance evaluators of the models, with the study identifying XGBoost as the best performing model. This study highlighted the effectiveness of GridSearchCV in model optimisation.

This section illustrates how machine and deep learning techniques have been used in the prediction of credit default risk. Logistic regression, random forests, decision trees, SVMs, and MLPs have been popularly used. Furthermore, the effectiveness of boosting classifiers and ensemble techniques in improving model performance and dealing with large datasets have been documented. Authors like [29,38,39] have emphasised the importance of handling class imbalance, although different ratios have been used to split the dataset. Additionally, hyperparameter tuning and feature selection have been effective at enhancing model performance, with metrics like accuracy, precision, recall and the AUC-ROC commonly used for the evaluation. In conclusion, accurately detecting credit defaults remains a concern to financial institutions, especially the role it plays in reducing financial losses [43], and, while previous studies have applied various methods to accurately predict credit defaults, no technique has been set as the best. Furthermore, the combining of boosting classifiers, testing different sampling techniques and validating the models with various performance metrics remains an area with room for improvement; therefore, this current paper aims to solve this issue with a slightly different approach and methodology with respect to the existing literature.

3. Methodology

This section discusses the methods used in this study, starting with the data collection process to the model deployment stage. The data collected were from LendingClub, which is a lending platform that provides detailed information of each loan that was issued from 2007 to 2018. Given the focus of this study, only the confirmed good and bad loans were used; therefore, the target is defined as follows:

$$\text{Target(y)} = \begin{cases} 0: & \text{where loan status = “Fully Paid”} \\ 1: & \text{where loan status = “Charged off”} \end{cases} \quad (1)$$

To balance the system’s efficiency and have a representative of the data, 30% of the data was sampled using a stratified sampling method [44], which resulted in a sample size of 403,593, consisting of 323,025 non-defaults and 80,568 defaults, across 152 variables. This approach ensured that there were sufficient data without overwhelming the system.

The framework of our approach, as illustrated in Figure 1, consists of different stages, where diverse techniques were tested (when required) to identify the most effective approach. The process is sequential, which means that each stage must be completed before the next stage begins. The subsequent sections outline the data preparation and analysis stages.

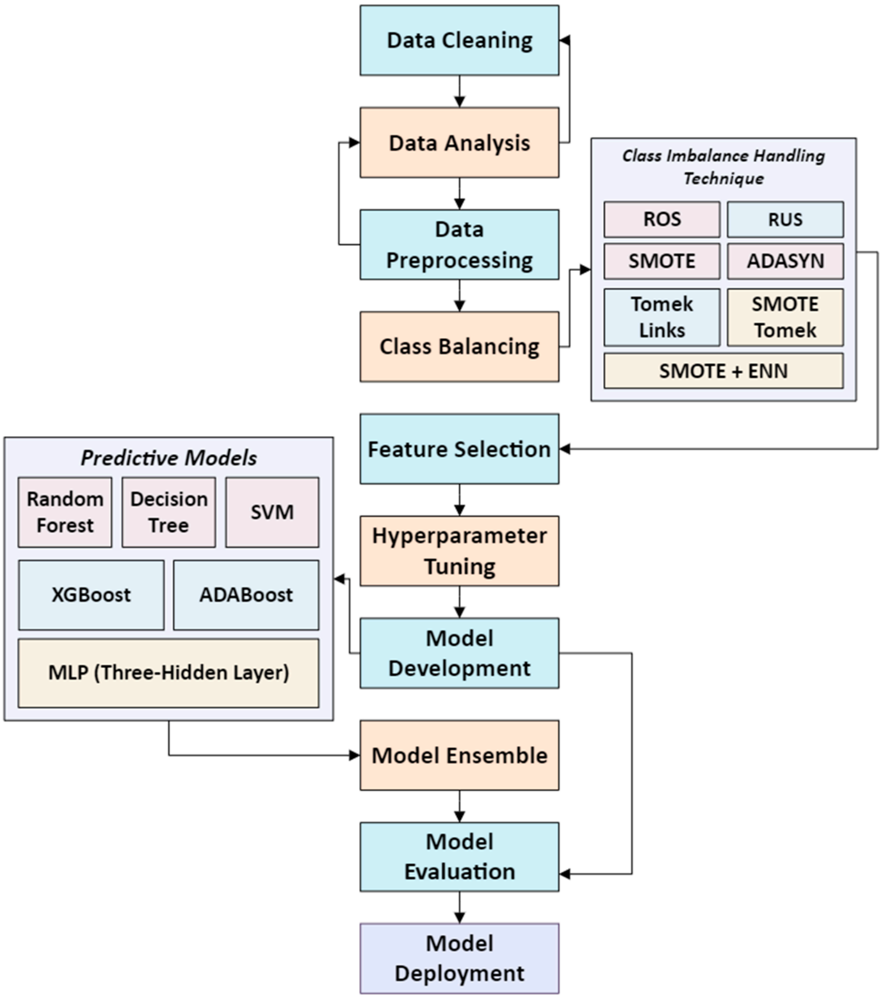

Figure 1. Research framework.

3.1. Data Cleaning

The next stage involved data cleaning and pre-processing, which prepared the data for proper analysis, ensuring the quality and reliability of the data. The approach used was similar to [28], which involved performing multi-observation cleaning such as handling missing values, identifying and correcting errors or inconsistencies and removing features that could potentially bias the analysis. The dataset contained no duplicates; however, there were 104 features with missing values. The columns with more than 50% of their data missing were excluded from the analysis. Additionally, categorical features with a large number of missing values that were deemed as not useful for the analysis were removed: @emp_title’ with 142,402 unique values and @title’ with 21,976 unique values that were similar to the @purpose’ feature and @emp_length’, which showed similar bad loan rates (%) across its group, were removed. Moreover, to avoid losing vital information from the numerical columns, the strategy to handle the missing values was derived based on the distribution (skewness and kurtosis), SciPy, was used in calculating the Fisher–Pearson coefficient [45]:

$$\text{Skewness} = \frac{m_3}{{m_2}^{3/2}} \tag{2}$$

where

- ith central point (m_i) is defined as:

$$m_i = \frac{1}{N}\sum_{n=1}^{N}(x[n] - \bar{x})^{i} \tag{3}$$

- N = sample size
- $\bar{x}$ = mean

Median imputation, which is robust to outliers, was used for skewed features, while mode imputation was carried out on features that were multimodal [38,46] to preserve data integrity.

The Pearson correlation coefficient (r) is a filter method that measures the relationship between variables [47] and is consistent with the approach used by [1]. Variables above 90% r with other features were removed, as they could cause multicollinearity, which may mislead the model’s performance [48]. It can be calculated as the following:

$$\text{Correlation (r)} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \tag{4}$$

3.2. Data Analysis

In this work, we focus our attention to developing a predictive model for credit loan defaults and not on the reasons that customers default in their payments. We provided the exploratory analysis of the data, to perform further pre-processing, carried out after the initial cleaning stage to draw conclusions about the data [49]. Descriptive analysis, such as count, mean, median, and standard deviation, were used to summarise the numerical features and identify errors; furthermore, data visualisation was used to analyse the features and remove the ones that do not add any information, which allowed some features to be excluded, and the state information @addr_state’ to be converted to the region, so as not to completely miss out on any benefit that the location might have. Additionally, the descriptive analysis showed that there were outliers and possible errors in some features; for instance, @annual_inc’ had a maximum value of GBP 9,522,972, which is a possible error and would likely affect the debt-to-income ratio @dti’. The analysis showed a maximum DTI of 999.00%.

While the full summary of all features was analysed, because of clarity and brevity, only features with significant outliers or extreme values (@annual_inc’ and @dti’) are presented in Table 1. These features were chosen due to their high variability, as indicated by the large standard deviations relative to their means, suggesting the presence of extreme values that required attention during the data cleaning and preprocessing stages [46]. Specifically, the standard deviation for @annual_inc’ is approximately equal to its mean, reflecting the wide range of incomes in the dataset, including a few exceptionally high values. Similarly, the standard deviation of @dti’ is high due to outlier values that likely represent errors or extreme cases in the data.

Table 1. Descriptive analysis of annual income and DTI.

features	count	mean	std	50%	max
annual_inc	403,593	76,278.3	71,140.2	65,000	9,522,972
dti	403,593	18.26	10.38	17.62	999

These extreme values were further confirmed, as shown in Figure 2. The errors and outliers were handled in the data pre-processing stage.

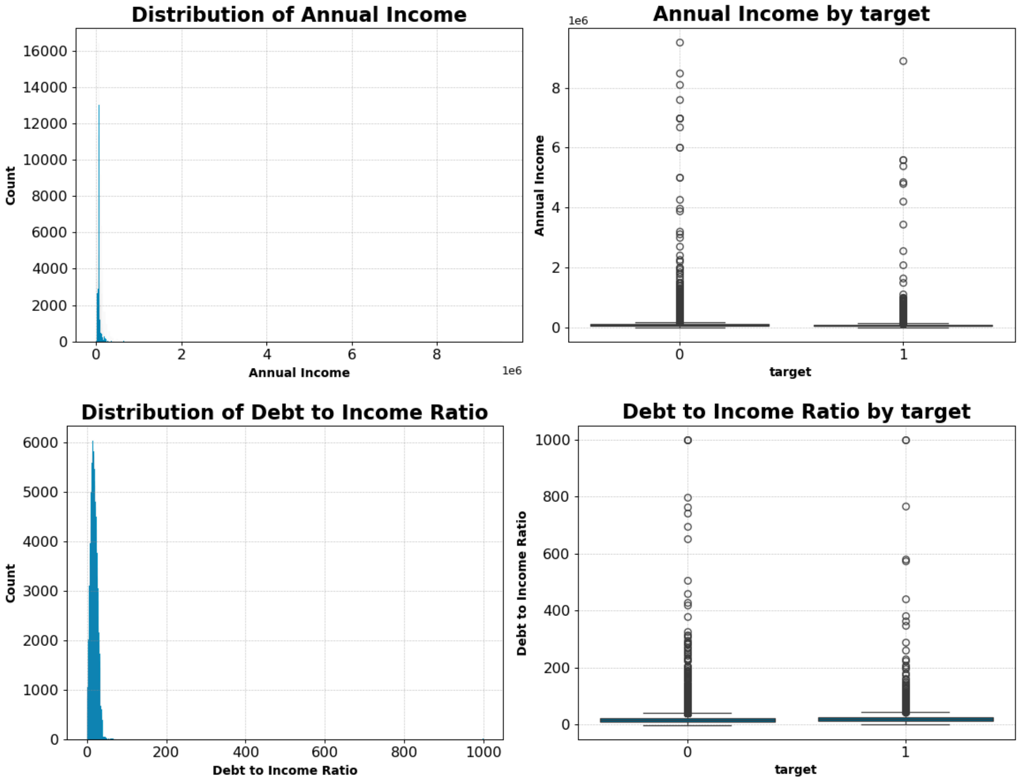

Figure 2. Visualisation of annual income and debt to income ratio (DTI).

3.3. Data Pre-Processing

This stage is very important in getting the data ready for model development. Here, observed errors were removed, outliers were treated, features were binned and combined to capture more information, categorical data were one-hot encoded and transformed to numerical data and, finally, the values were normalised [50,51]. At each stage, different techniques were tested, using XGBoost, to identify the best technique to utilise, similar to [42], who used XGBoost to test the balancing methods and feature selection used on the other models. This model was selected because it is simple yet powerful and is known for its ability to generalise well to other models; moreover, it is efficient, which helped to save time and improve the performance of the models [52]. Furthermore, after the observed errors in features like @annual_inc′ and @dti′ were removed, and categorical features shown in Table 2 (full data description can be seen in Table A1) were one-hot encoded, the outliers observed in the numerical features were handled, and the data were normalised.

Table 2. One-hot encoded features.

Features	Categories
home_ownership	(Any, Mortgage, Other, Own, Rent)
verification_status	(Not Verified, Verified)
purpose	(car, credit_card, debt_consolidation, educational, home_improvement, house, major_purchase, medical, moving, other, renewable_energy, small_business, vacation, wedding)
initial_list_status	(F: Fractional, W: Whole)
application_type	(Individual, Joint App)
region	(MidWest, NorthEast, SouthEast, SouthWest, West)
annual_inc_binned *	(Very Low, Low, Medium, High, Very High)
revol_bal_binned *	

* Binned annual income and revolving balance that may capture non-linear relationships.

3.3.1. Handling Outliers

Outliers are extreme values that are different from the rest of the data and can influence some models, which is why it needs to be addressed. The best technique to handle the outliers was identified to reduce the effect of the outliers while retaining as much data as possible. Z-score, IQR, clip and winsorize methods [53,54] were tested:

- Standard score (z-score): Informs how far a data value (V) deviates from the mean (μ), in regard to the standard deviation (σ). A Z-score (Z) greater than 3 shows the extreme values. It is calculated as follows:

$$Z = \frac{(\mathrm{V} - \mu)}{\sigma} \tag{5}$$

- Interquartile Range (IQR): Q1 (first quartile: 25%) and Q3 (third quartile: 75%) were used for the calculation, and values that fall outside these bounds are considered outliers.
- Clip: Considers the values below and above the 1st and 99th quartile as outliers.
- Winsorize: Limits the extreme values to a specified percentile.

3.3.2. Data Normalisation

Features in a dataset with a different range can affect some models. This concern was handled by scaling the features using the following normalisation techniques [53,55]:

- Standard scaler: Scales the new value (n) to follow a normal distribution; however, it can be affected by outliers. It is calculated as follows:

$$\mathrm{n} = \frac{\mathrm{n_i} - \mathrm{n_{mean}}}{\sigma} \tag{6}$$

- Min-max scaler: Scales the data to [0, 1] range. Although it is not as sensitive to outliers as the standard scaler, it, however, can be influenced by them. It is calculated as follows:

$$\mathrm{n} = \frac{\mathrm{n - minimum(n)}}{\mathrm{maximum(n) - minimum(n)}} \tag{7}$$

- Robust scaler: Uses the median and the IQR, which reduces the effect of outliers. It is calculated as follows:

$$n = \frac{n_i - n_{median}}{IQR} \quad (8)$$

3.3.3. Evaluation Metrics

To assess the effectiveness of the models, including the model used in testing (XGBoost), various metrics were used [56]. Additionally, they were used in this stage to identify the best pre-processing techniques to use.

- Accuracy, which measures the ratio of the correct predictions (both "positive" defaults and "negative" non-defaults) to the total number of predictions [28,29]:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FT+FN}} \quad (9)$$

- Precision, which measures the proportion of the actual defaults among all default predictions [38,42]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (10)$$

- Recall, which is also known as sensitivity or the True Positive Rate (TPR), measures the proportion of the actual defaults that are correctly identified [38–40], calculated as follows:

$$\text{Recall(Sensitivity)} = \frac{\text{TP}}{\text{TP+FN}} \quad (11)$$

- AUC, which measures the ability of the model to differentiate between defaulters and non-defaulters across all classification thresholds and is particularly useful in an imbalanced dataset, and the ROC curve plots the TPR against the False Positive Rate (FRP) [22,29,38].

3.3.4. Identifying Data Pre-Processing Techniques

Recall, precision and accuracy were metrics used in the selection process, as they provided a comprehensive view of how the models performed across different dimensions, as stated in Section 3.3.3. The results of testing various outlier and normalisation techniques are presented in Table 3, demonstrating how different combinations of pre-processing methods impact model performance.

Table 3. Identifying outlier and normalisation pre-processing methods.

Outlier Technique	Normalisation Technique	Accuracy	Recall	Precision	AUC
z_score	Minmax	0.7961	0.0445	0.5444	0.6969
z_score	Standard	0.7963	0.0449	0.5497	0.6971
z_score	Robust	0.7963	0.0449	0.5497	0.6972
iqr	Minmax	0.8275	0.0035	0.5882	0.6407
iqr	Standard	0.8274	0.0035	0.5263	0.6411
iqr	Robust	0.8274	0.0031	0.5294	0.6410
winsorize	Minmax	0.8040	0.0567	0.5544	0.7032
winsorize	Standard	0.8044	0.0584	0.5625	0.7038
winsorize	robust	0.8045	0.0582	0.5664	0.7039
clip	Minmax	0.8036	0.0544	0.5473	0.7048
clip	Standard	0.8040	0.0556	0.5571	0.7049
clip	Robust	0.8038	0.0550	0.5516	0.7050

The combination of the ◎winsorize′ method for outlier handling and the ◎robust scaler′ for normalisation was the optimal pre-processing strategy, as indicated by slightly higher recall (0.0582) and precision (0.5664) compared to other combinations. The ◎Robust scaler′ was selected because it is less sensitive to outliers, as it uses the interquartile range to scale features, which is particularly helpful for datasets with significant outliers. Additionally, Figure 3 shows the distribution of the two critical features ◎annual_inc′ and ◎dti′ after the outliers were handled, highlighting how the distribution of these features was adjusted, with extreme values capped, ensuring that they do not disproportionately influence the model's predictions.

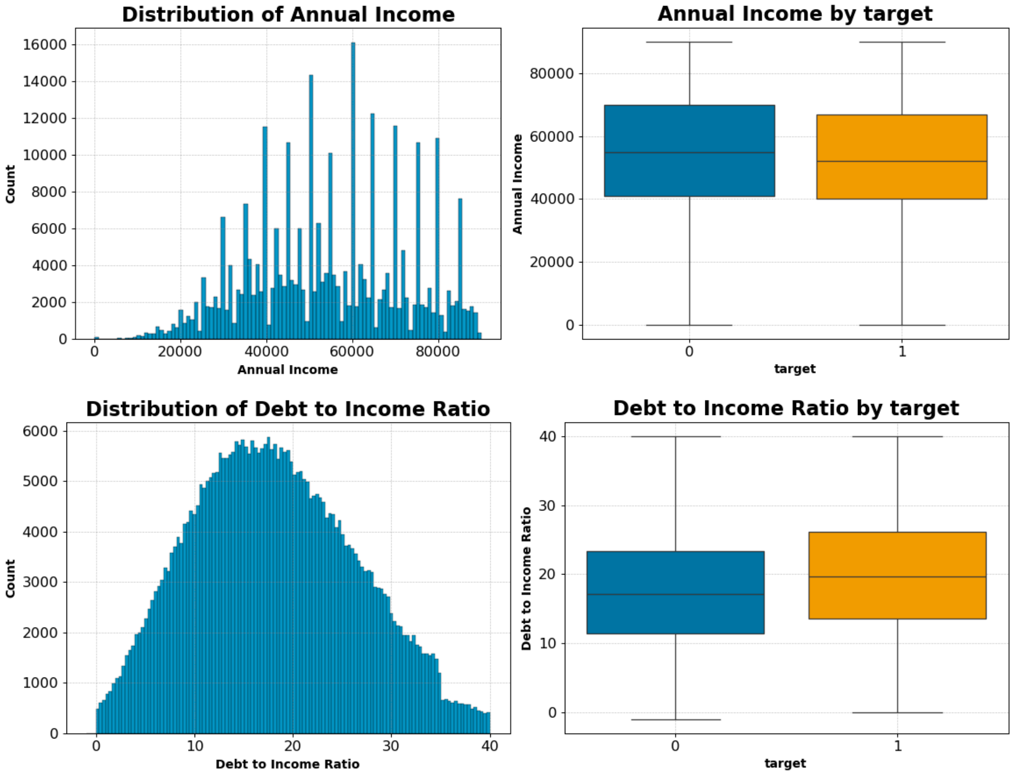

Figure 3. Visualisation of the features after handling outliers.

The ◎winsorize′ method was selected to handle the outliers, and the ◎robust scaler′ the data normalisation, because of their ability to improve recall without sacrificing the overall performance, motivated by the need to prioritise correctly identifying defaults, which is important in credit risk prediction.

3.4. Addressing Class Imbalance

Class imbalance is a significant issue in predictive modelling, particularly in fields like credit risk prediction, where most observations belong to the non-default class [33,37–41]. This issue can lead to biased models that perform well on the majority class while underperforming on the minority class [29,52]. This outcome was evidenced in Section 3.3.4, when the outlier and normalisation technique were tested, as the model's accuracy, which measures the proportion of correct predictions that the model made [28], was significantly biased towards the non-defaults. In contrast, recall, which measures the proportion of the actual defaults correctly predicted by the model [38], showed the model's weaker performance on the minority class. This finding further highlights why the class imbalance issue needs to be addressed, especially given this study's goal of developing a model that accurately predicts credit defaults. Additionally, as shown in Figure 4, the

LendingClub dataset is imbalanced with 80% more non-default class than the default class, making it very important to address this issue.

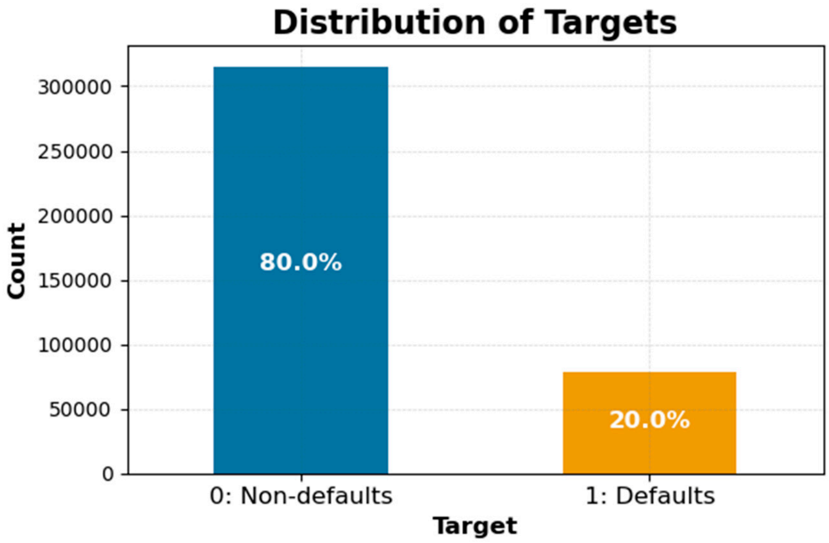

Figure 4. Distribution of targets.

There are several techniques that can be used to tackle the issue of class imbalance, but no single one is regarded as the best. While popular techniques such as the SMOTE and ADASYN are used frequently [38–41], this research requires the identification of the best technique to use; therefore, different techniques were tested, as shown in Figure 5.

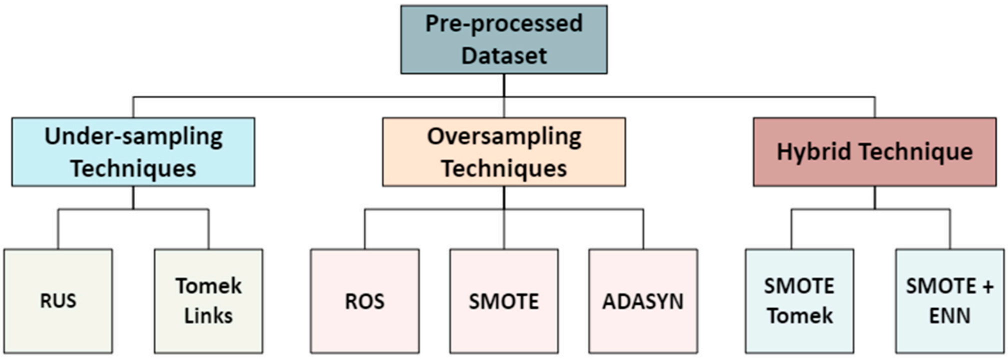

Figure 5. Class imbalance handling techniques.

The techniques evaluated are as follows:

1. Random Over-Sampling (ROS): It works by randomly adding data samples from the minority class to the dataset until the whole dataset is balanced, and, although ROS uses the majority class (non-defaults) to balance the minority class (defaults), it may cause the model to perform well for the training set and poorly for the testing set (overfitting), which is due to the duplicated samples, as the model may learn only from the defaults (in this case) and may not generalise well to new data [52,57]. It can be represented as

$$\text{New } S_{\text{minority}} = S_{\text{minority}} \cup \{S_{\text{minority}} \text{ duplicated until } |S_{\text{minority}}| = N_{\text{majority}}\} \quad (12)$$

where

- S_{minority} = Minority class samples.
- N_{majority} = Number of majority class samples.

- $|S_{\text{minority}}|$ = Current size of the minority class.

2. Random Under-Sampling (RUS): It works by filling the minority class with data from the majority class, thereby reducing the majority class until the whole dataset is balanced [52]. This technique may cause a loss of information for the majority, which may affect the model when learning new patterns. Unlike ROS, this technique may not lead to overfitting [58]. It can be shown as follows:

$$\text{New } S_{\text{majority}} = S_{\text{minority}} \cup \{S_{\text{majority}} \text{ duplicated until } |S_{\text{majority}}| = N_{\text{minority}}\} \tag{13}$$

3. SMOTE: It works by generating synthetic data through interpolating between the existing minority class data samples and their nearest neighbour, thereby adding new data points without adding duplicates, and, since the minority class samples are increased without duplication, it may prevent overfitting [38,59]. It generates synthetic data (x_{new}) with the following:

$$x_{\text{new}} = x_i + \lambda \times (x_{nn} - x_i) \tag{14}$$

where

- x_i = minority class.
- x_{nn} = one of the nearest neighbours.
- λ = a random value between [0, 1].

4. ADASYN: It works in a similar way to the SMOTE, but it focuses on generating more synthetic samples for the harder-to-classify minority class [38,41]. The number of the synthetic sampled i, (G_i) is calculated as follows:

$$G_i = d_i \times G \tag{15}$$

where

- d_i = ratio of the majority neighbours.
- G = total synthetic samples needed.

This technique improves classification by focussing on more challenging data.

5. Tomek Links is an under-sampling technique that cleans up the data by locating and removing ambiguous or noisy data samples that are near the decision boundary. Given a majority class (x_1) and a minority class (x_2), if they are the nearest neighbours and they belong to different classes, they form a Tomek Link, and removing them will help to clean the boundary between classes [59,60]. It can be represented as the following:

$$\text{Remove } (x_1, x_2) \begin{cases} \text{if } x_1 \text{ and } x_2 \text{ are nearest neighbours} \\ + \\ \text{if } x_1 \text{ and } x_2 \text{ belong to different classes} \end{cases} \tag{16}$$

6. SMOTE-Tomek: It is a combination of the SMOTE and Tomek Links. Firstly, the SMOTE is used to generate the synthetic data samples for the minority class, and, then, Tomek Links are removed to clean up the boundaries between classes, thereby improving the quality of the synthetic data [56].
7. SMOTE+ENN: It is a hybrid technique that improves the quality of the synthetic data created by the SMOTE, as the Edited Nearest Neighbour (ENN) is used to remove instances of misclassification of the nearest neighbour [58,61].

ENN cleaning:

$$|\text{If } x_i \text{ is misclassified by its k nearest neighbours, remove } x_i| \tag{17}$$

This technique can be represented as follows:

$$S_{balanced} = \text{ENN}(\text{SMOTE}(S_{minority}, S_{majority})) \tag{18}$$

- $\text{SMOTE}(S_{minority}, S_{majority})$ generates x_{new}.
- The ENN removes the noisy data.

To address the issue of class imbalance in the LendingClub dataset, the resampling techniques were tested using XGBoost and evaluated using accuracy, precision, recall and the AUC. After balancing the data, the next step involved testing various splits to determine the best split to use, with 20%, 25%, 30%, 35% and 40% test ratios evaluated. Ultimately, an 80:20 split was selected for the model development in the next stage, ensuring that the appropriate features were chosen.

3.5. Feature Selection

Feature selection is a crucial step in model development, and the goal here is to obtain features that can be used in simple and efficient models, as deploying a model with a large number of features can be computationally expensive; therefore, this stage facilitates the reduction and removal of redundant features that may not be useful for model development [38,62]. The primary method used in this stage was the wrapper feature selection method, Recursive Feature Elimination with Cross-Validation (RFECV), which is a method that iteratively uses learning algorithms to select the best features to make use of by evaluating the performance of the model [63]. This method aims to find the features that gives the best performance using a scoring metric (scorer). Given that the focus was to correctly predict defaults, recall was used. Additionally, the ⊛step' parameter was set to 1, which indicates that one feature is removed per iteration; moreover, redundant features were also removed, thereby ensuring that the best features were selected for the model development process.

3.6. Model Development Process

This process involved using the selected features in the development of predictive models to identify the best performing model that can be used to identify credit default risk. Additionally, hyperparameter tuning and ensemble methods are further used to optimise the models.

3.6.1. Predictive Models

1. Decision tree has a tree structure that works by recursively splitting the data into subsets of the tree based on a decision rule [1,28]. In this study, it selected the best feature to split based on Gini index criteria—the impurity of a node and the values closer to 0 are the purer nodes—and is calculated as follows:

$$\text{Gini}= 1 - \sum_{i=1}^{c} (p_i)^2 \tag{19}$$

where

- p_i = proportion of the data sample that belongs to the class i in a tree node.
- c = number of classes.

2. Random forest is an ensemble learning method that combines the predictions obtained from training multiple decision trees to obtain the final predictions [60,64]. The final predictions were made using majority voting. Since it is a combination of decision trees, it used the Gini as well for splitting [39,42].
3. The SVM finds the optimal hyperplane that separates the data into different classes [20,43]. It used kernel functions to handle non-linear separation by mapping input features into high-dimensional spaces [37,43]. The hyperplane is calculated as follows:

$$h(x_i) = \text{sign}(w \cdot x_i + b) \tag{20}$$

4. XGBoost builds an ensemble of weak learners in an iterative manner in order to improve on the models' performance [28,29,42]. It used gradient boosting with specific loss functions l and regularisation terms $\Omega(f)$:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t\text{-}1)} + f_t(X_i)\right) + \Omega(f_t) \tag{21}$$

where

- $L^{(t)}$ = total loss at iteration t.
- n = data points.
- $l\left(y_i, \hat{y}_i^{(t-1)} + f_t(X_i)\right)$ represents the loss function that measures the difference between the true and predicted labels.
- $\hat{y}_i^{(t-1)}$ is the previous iteration's predicted class.
- $f_t(X_i)$, is the current model's prediction.

5. ADABoost focuses on creating strong classifiers by combining multiple weak classifiers. It trains weaker learners on the errors made by the previous ones, and, when there is a misclassification, it assigns more weight to them [23,24,39]. Final predictions are calculated by the following:

$$H(x) = \text{sign}\left[\sum_{t=1}^{T} \alpha_t h_t(x)\right] \tag{22}$$

where

- T = weak classifiers.
- α_t = weight for the weak classifier.
- $h_t(x)$ = predictions for the weak classifier.
- $sign$ = determines the final prediction.

6. The MLP is a feedforward type of ANN that consists of the inner layer, multiple hidden layers and an outer layer, and each layer is made up of neurons connected to those in the previous and following layers [31,33]. Each neuron uses an activation function to introduce non-linearity. Common activation functions include the ReLU (Rectified Linear Unit), which is widely used due to its efficiency in solving the vanishing gradient problem, which may be encountered with other functions, and the sigmoid function, used in the output layer for binary classification tasks [21,42]. The architecture of the MLP is important in determining the model's capacity to capture and learn complex patterns. During training, the MLP uses backpropagation to adjust the weights of the connections based on the error in the output, minimising the loss function [30,32]. The loss function is defined as follows:

$$L = -\frac{1}{n}\sum_{i=1}^{n} [y_i log\ (\hat{y}_i) + (1 - y_i)\ log(1 - y_i)\ ] \tag{23}$$

where

- n = data points.
- y_i = true class label [0, 1].
- $\hat{y}_i$ = predicted probability of the positive class.

In this study, the choice of hyperparameters to use—such as maximum depth, regularisation, number of neurons, number of layers, and learning rates—was determined using GridSearchCV, an optimisation technique discussed in the next section.

3.6.2. Hyperparameter Tuning

Hyperparameter Tuning is an important technique that can help optimise each model, as it directly influences their capacity to generalise to unseen data [35,42]. In this study, we used GridSearchCv to identify the optimal parameters used for model development. This technique tests combinations of predefined hyperparameter values through cross-validation. The values tested and presented in Table A2 were predominantly chosen to strike a balance between reducing overfitting and increasing the performance of the models.

For instance, for decision trees and ensemble methods, parameters like max_depth and min_sample_leaf were chosen to control the tree's complexity and prevent overfitting. Values (10, 20) for max_depth and (100, 200, 500) for n_estimators options in random forest were chosen to allow the model to capture complexity in the data without becoming too complex and to balance bias and variance [65]. For the decision tree, the min_samples_leaf and min_samples_split values were chosen to help control overfitting by ensuring that the splits do not occur with too few samples. Additionally, the criterion of ‘gini′ and ‘entropy′ enables the model to explore different methods of node impurity [25,28].

For the SVM, the C parameter controlled the regularisation strength, where a lower value of C allowed for a larger margin, which simplified the decision boundary. The different kernel types (‘linear′, ‘rbf′, ‘poly′) were tested to determine the best way to transform the input space for better classification performance [20,37]. Additionally, for XGBoost, regularisation parameters such as reg_alpha (L1 regularisation) and reg_lambda (L2 regularisation) were tested. These parameters helped to control sparsity as well as the magnitude of the model's weights, enhancing the ability to handle noise in the data. The learning_rate (0.1, 0.2) and n_estimators (200, 300, 500) were chosen based on standard practise, as a lower learning rate usually needs more estimators for optimal performance, but it avoided exceeding the minimum during gradient descent [28,29].

In ADABoost, the learning_rate (0.15, 0.2) was set to find a balance between how much each learner contributes and the risk of overfitting. For the MLP, the hidden_layer_sizes (100, 100, 100) and (150, 150, 150) were chosen to strike a balance between depth and complexity, which ensured that there was enough capacity to capture complex patterns in the data. The ‘adaptive′ and ‘constant′ learning rate were tested, where ‘adaptive′ adjusts based on the model performance and ‘constant′ ensures a stable learning pace throughout the training process [42,66]. The best parameters identified through GridSearchCV tuning and used for the model development are shown in Table 4.

Table 4. GridSearchCV best parameters used for model development.

Models	Params
Random Forest	{‘max_depth′: 20, ‘min_samples_leaf′: 1, ‘min_samples_split′: 2, ‘n_estimators′: 500}
Decision Tree	{‘class_weight′: None, ‘criterion′: ‘gini′, ‘max_depth′: 15, ‘max_leaf_nodes′: None, ‘min_impurity_decrease′: 0.01, ‘min_samples_leaf′: 1, ‘min_samples_split′: 2}
SVM	{‘C′: 1, ‘degree′: 2, ‘gamma′: 1, ‘kernel′: ‘rbf′}
XGBoost	{‘colsample_bytree′: 0.9, ‘learning_rate′: 0.1, ‘max_depth′: 20, ‘n_estimators′: 200, ‘reg_alpha′: 1, ‘reg_lambda′: 1.5, ‘subsample′: 1.0}
ADABoost	{‘learning_rate′: 0.15, ‘n_estimators′: 300}
MLP	{‘activation′: ‘relu′, ‘alpha′: 0.001, ‘batch_size′: ‘auto′, ‘early_stopping′: True, ‘hidden_layer_sizes′: (150, 150, 150), ‘learning_rate′: ‘constant′, ‘solver′: ‘adam′}

3.6.3. Ensemble Techniques

In this stage, some of the models, as well as the top three best performing models, are combined using the voting and stacking method. The first method, soft voting, takes the average probability predictions from the models as the final prediction [39,56]. Additionally, the second method uses the stacking method for the combination. Here, a meta-model was used to obtain the final prediction. The meta-model learns how best to aggregate the predictions to make the final prediction [35,67]. The ensemble methods used in this work and how they are combined are shown in Figures 6 and 7, respectively.

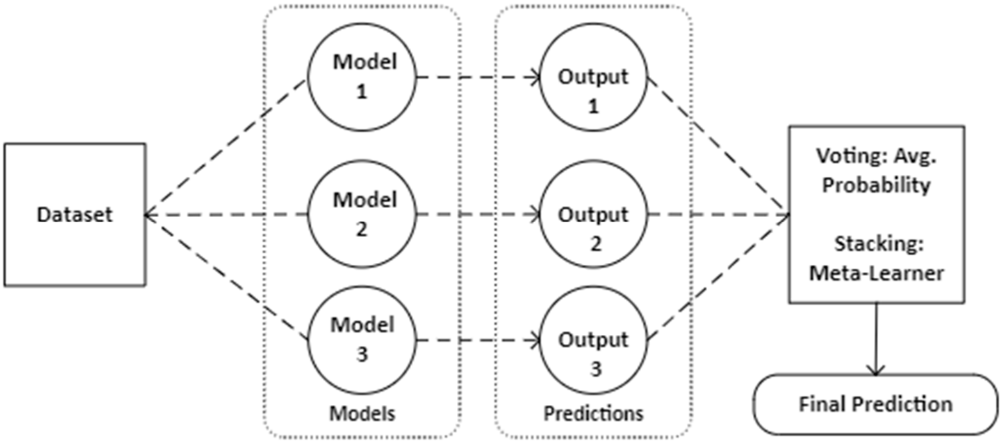

Figure 6. Ensemble methods.

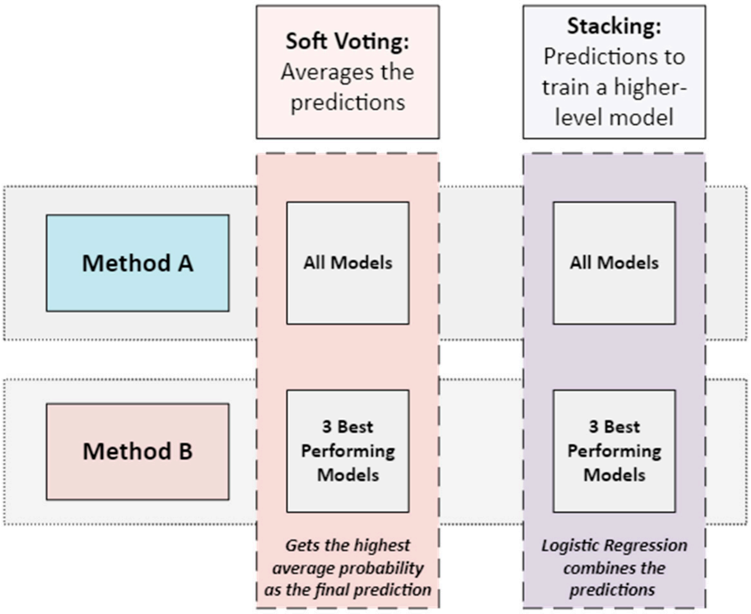
Soft Voting: Averages the predictions
Stacking: Predictions to train a higher-level model
Method A
All Models
All Models
Method B
3 Best Performing Models
3 Best Performing Models
Gets the highest average probability as the final prediction
Logistic Regression combines the predictions

Figure 7. Methods used in combining the model predictions.

4. Results

In this section, we present the results of the model development process discussed in Section 3.6. Additionally, we compare these results with those from other related studies, serving as a baseline to further validate the results obtained in this research. However, before we discuss the model performance, we first discuss the results related to addressing class imbalance and feature selection as outlined in Sections 3.4 and 3.5. Therefore, Table 5 shows the results of employing the sampling techniques, while Figure 8 shows the results from using RFECV.

Table 5. Sampling implementation.

Method	Accuracy	Precision	Recall	AUC
None	0.8047	0.5362	0.1101	0.7171
ROS	0.6874	0.6807	0.7062	0.7559
SMOTE	0.8766	0.9684	0.7787	0.9284
ADASYN	0.8745	**0.9686**	0.7690	0.9266
RUS	0.6500	0.6465	0.6683	0.7079
Tomek-Links	0.7947	0.5368	0.1377	0.7197
SMOTE-Tomek	0.8762	0.9679	0.7779	0.9295
SMOTE + ENN	**0.9049**	0.9461	**0.9202**	**0.9654**

As presented in Table 5, ROS performed better than RUS, which coincides with the observation made by [38] that the over-sampling technique always performed better than the under-sampling technique. While ADASYN, the SMOTE and SMOTE-Tomek showed impressive results, the SMOTE + ENN showed the most impressive performance across all the metrics; hence, by combining both the SMOTE and the ENN, the data were not only being balanced but also the noise or ambiguous data samples that may affect the model's performance were removed [68]. Additionally, with a recall of 92.02%, it showed that the model captures the minority class correctly, which is sometimes more important than obtaining a high accuracy. Furthermore, given the result, the SMOTE + ENN was used to balance the dataset.

Additionally, Figure 8 shows how the recall changes as the features are added, with the optimal features identified when the score plateaus. It also shows the standard deviation of the CV scores, which show the variability and stability of the model's performance across the folds, with 48 features identified as the optimal number of features to obtain the optimal recall score of 92.16%.

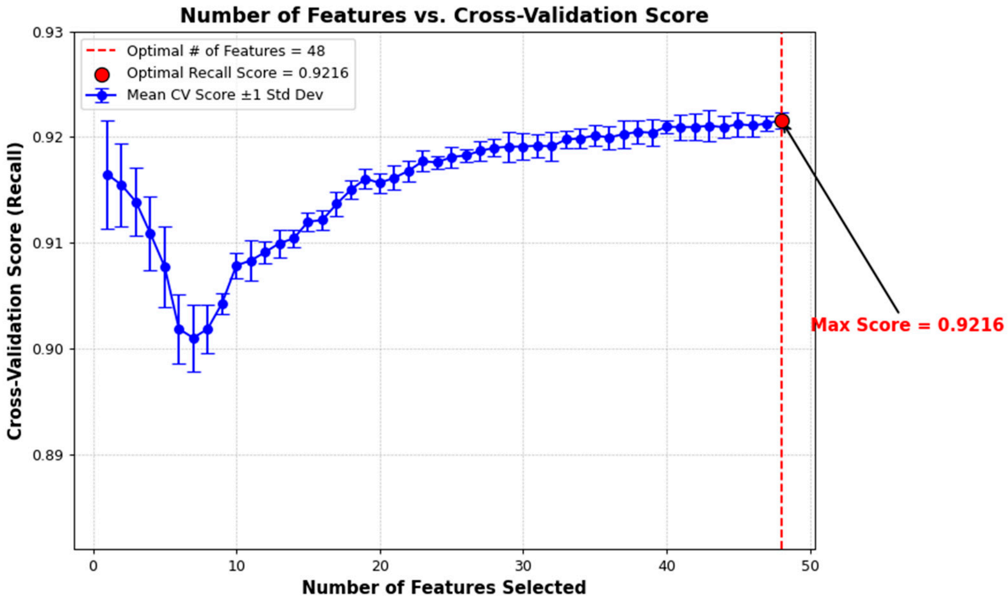

Figure 8. Features selected.

Additionally, the results for the feature importance are shown in Figure 9 below, with the interest rate, credit score and the loan term identified as the most important features in the prediction of credit defaults. Based on these findings and the best parameters listed in Table 4 (Section 3.6.3), the models were subsequently developed.

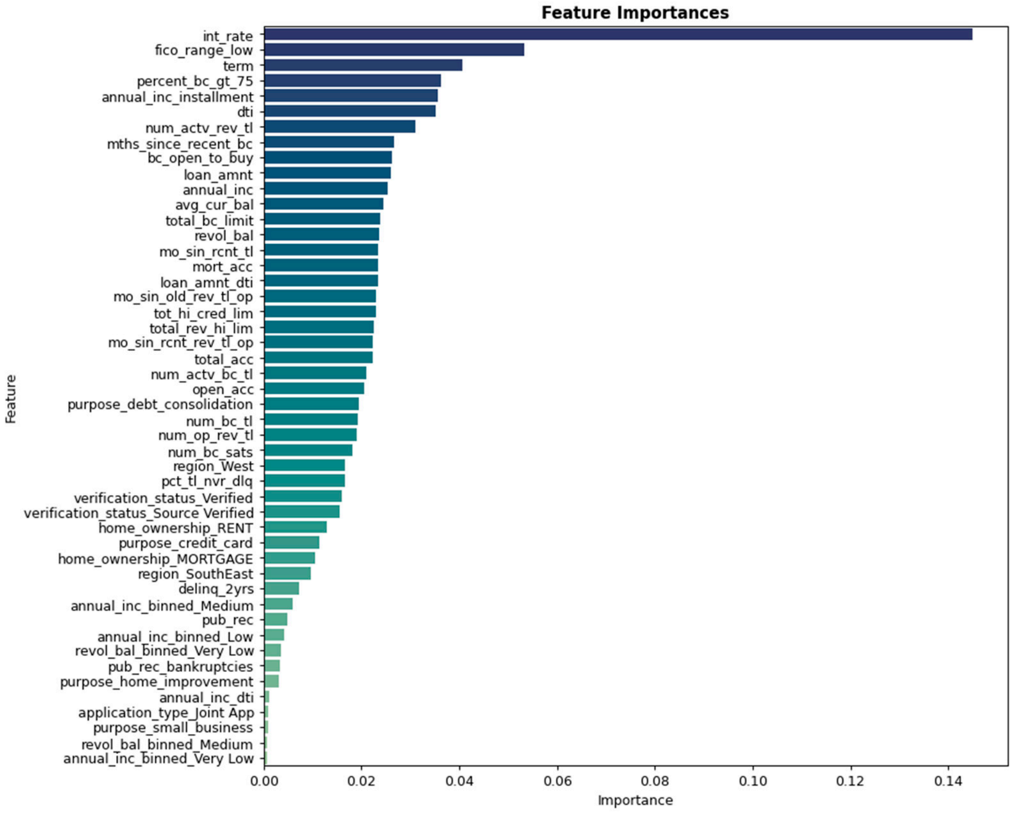

Figure 9. Feature importance.

4.1. Model Performance Evaluation

4.1.1. Individual Model Performance

The metrics discussed in Section 3.3.3 were used in the evaluation of the model performance, with the performance of the individual models presented in Table 6.

Table 6. Individual model result.

Model	Accuracy	Precision	Recall	AUC
Random Forest *	0.8987	0.8996	0.9656	0.9589
Decision Tree	0.7778	0.7743	**0.9713**	0.7256
SVM	0.7318	0.9476	0.6601	0.8824
XGBoost *	**0.9156**	**0.9478**	0.9330	**0.9726**
ADABoost	0.8458	0.8548	0.9439	0.9305
MLP *	0.8775	0.9008	0.9305	0.9229

* Indicates models’ part of the ensemble with 3 base-learners.

Models with ensembled techniques like random forest and XGBoost outperformed simpler models like decision tree and the SVM, which shows the advantages of combining model predictions to improve their effectiveness. The models—random forest and XGBoost—showed strong performances with an accuracy of 89.87% and 91.56%, respectively. Additionally, the recalls, which indicate that the models can effectively identify default cases, were 96.56% (random forest) and 93.30% (XGBoost), with high AUC scores of 95.89% and 97.29%, which suggest that the models were able to effectively distinguish between the defaulters and the non-defaulters. Similarly, with a recall of 92.48%, the MLP had a solid performance; however, the SVM had the lowest score, despite having a high precision value of 94.76%, which may suggest that the SVM is not able to address the complexity of the credit default data.

The ROC curve in Figure 10 shows the performance of the individual models across different thresholds, displaying how well the models separate the non-default and the default class; furthermore, it shows that all the models performed well.

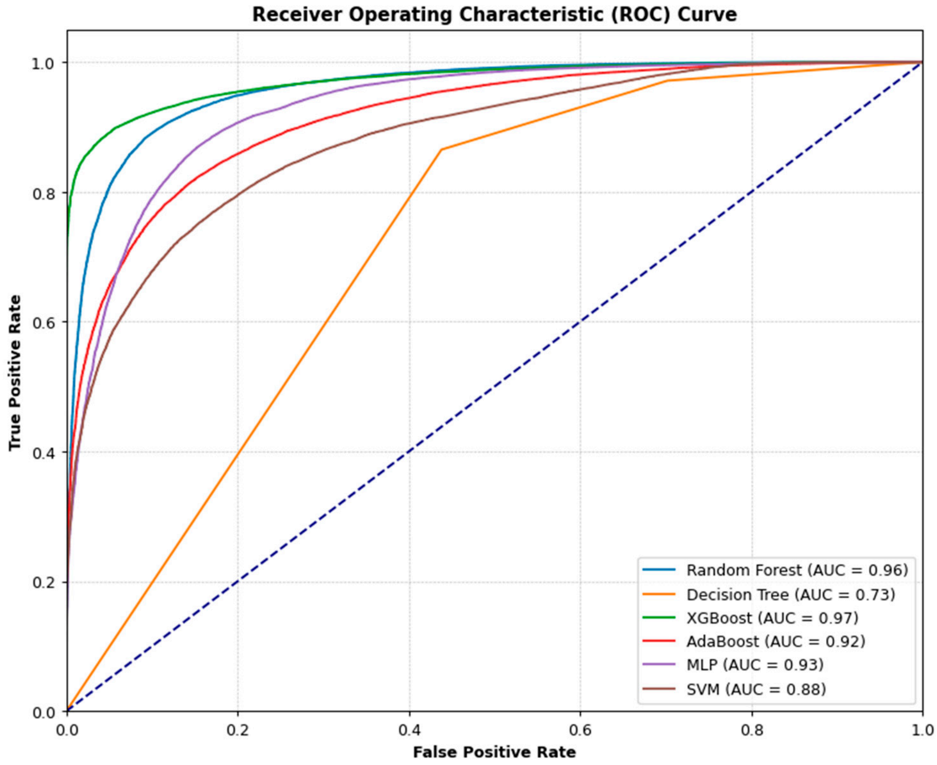

Figure 10. ROC curve (individual models).

For the individual models, XGBoost with an accuracy of 91.56%, a precision of 94.78% and an AUC of 97.26% achieved the best results, which shows how effective the model is at handling complex credit default data. The performance can be attributed to XGBoost's ability to create better predictions by combining the predictions from weaker learners, as well as the built-in regularisation that helps to prevent overfitting, giving it an edge, especially when compared to the other models, shown in Figure 11.

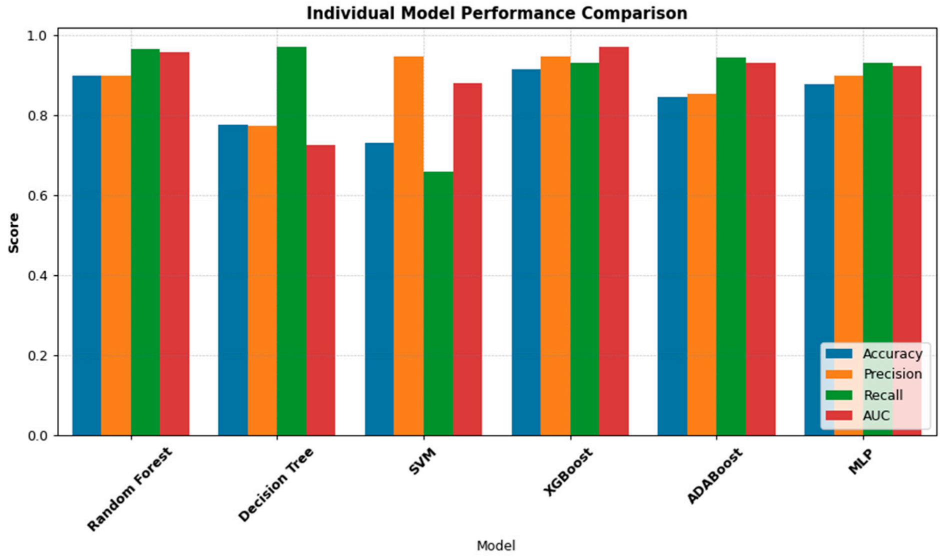

Figure 11. Comparative result (individual model).

4.1.2. Ensemble Model Performance

As detailed in Section 3.6.3, the predictions from the individuals can be combined to create stronger learners. The result of combining the models' predictions is shown in Table 7.

Table 7. Ensemble model result.

Model	Accuracy	Precision	Recall	AUC
Voting A	0.9109	0.9099	**0.9710**	0.9703
Voting B	0.9166	0.9314	0.9532	0.9687
Stacking A	**0.9369**	**0.9559**	0.9555	**0.9781**
Stacking B	0.9188	0.9409	0.9454	0.9708

Combining the models led to an overall increase in the performance, especially when the predictions are combined with the ensemble model, which uses a learner model (meta-model)—Stacking. Method A, which combines the predictions from all the models, achieved the highest result overall (Stacking A), with an accuracy of 93.69%, a precision of 95.59 and an AUC of 97.81%, which suggests that combining the models with the stacking technique led to an improvement in the performance.

XGBoost and the ensemble methods—Voting A, B and Stacking A, B—performed well, as seen in Figure 12; however, Stacking A's performance is impressive, as it was identified as the best model with the ability to effectively separate the classes. This result demonstrates the need for the inclusion of more algorithms in the ensemble process; therefore, we propose this technique for the classification of credit default risk. Furthermore, with a precision and recall of approximately 96%, the model shows how well the technique can enhance the individual models, as it uses a meta-model to learn how best to combine predictions.

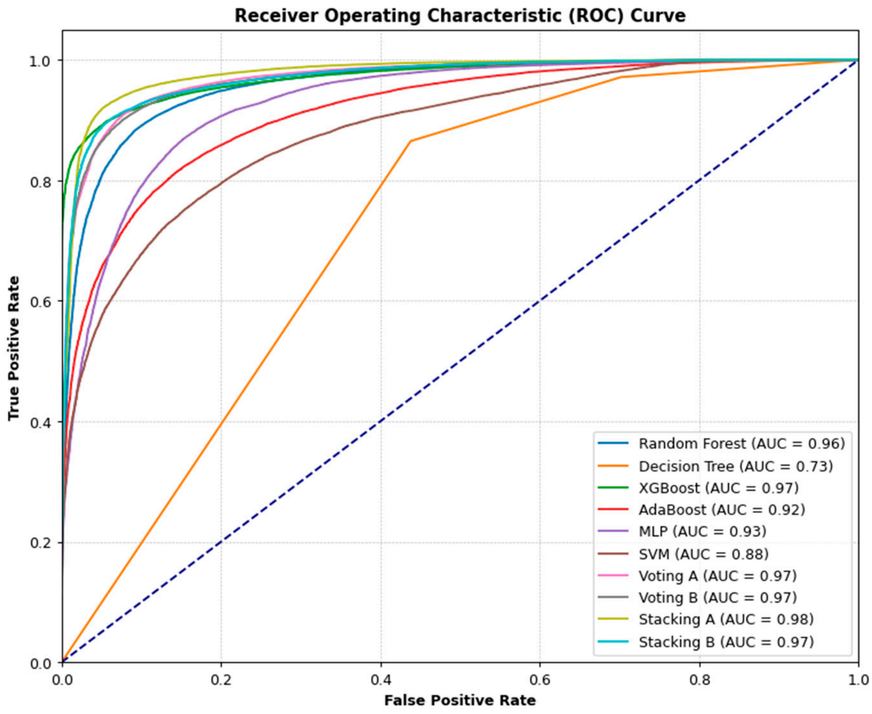

Figure 12. ROC curve (all models).

The comparative result can be seen in Figure 13 below, with Stacking A outperforming the other models, with an AUC of 98%, thereby showing how effective ensemble methods can be at optimising the performance of the models.

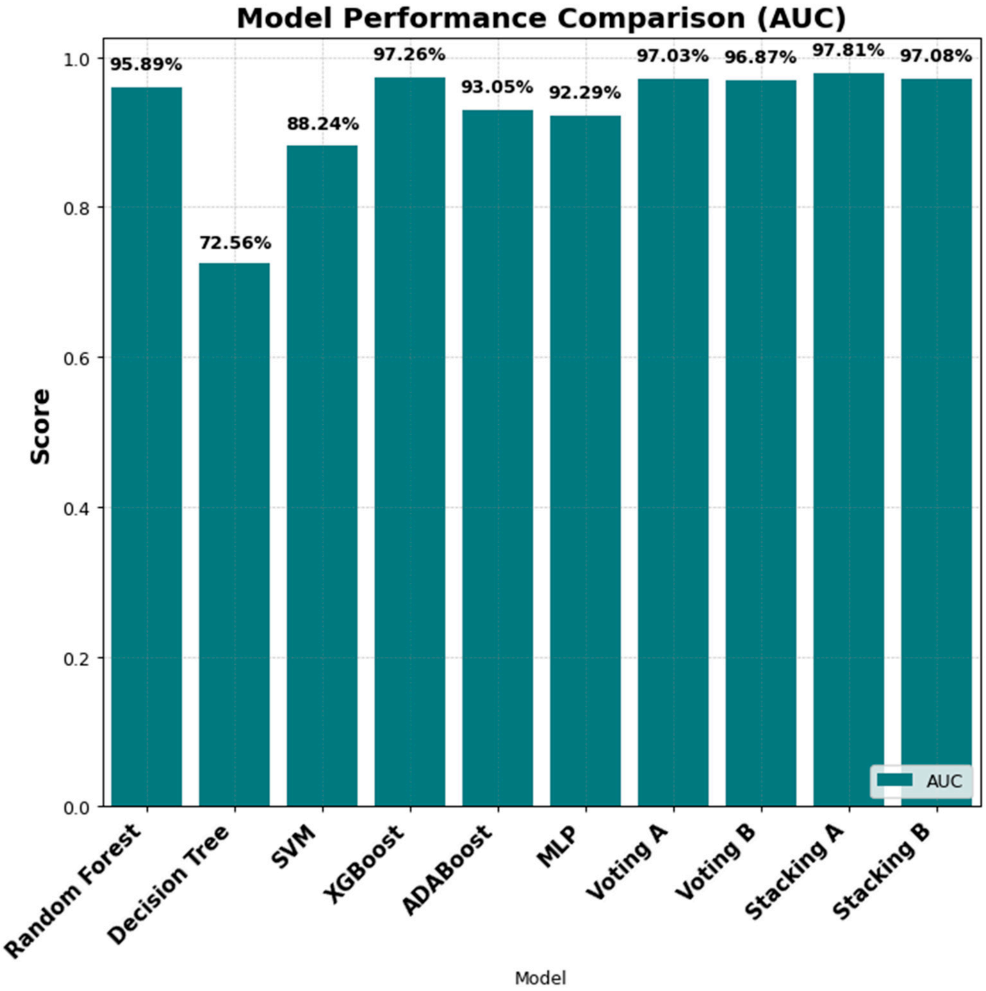

Figure 13. Comparative result (all models).

4.2. Comparison with Baseline Results

The performance of the proposed model was compared with other related works that used datasets similar to the one used in this study, the LendingClub dataset. This comparison served as the baseline for further evaluation, with the results summarised in Table 8 below:

Table 8. Baseline comparison.

Reference	Class Imbalance Method	Models	Ensemble Technique	Data Split	Performance Metrics	Best Model	Best Model Score
This study	ROS, RUS, SMOTE, ADASYN, Tomek Links, SMOTE-Tomek, SMOTE + ENN	Random Forest, Decision Tree, SVM, XGBoost, ADABoost, MLP (three hidden layer)	Voting, Stacking	80:20	Accuracy, Precision, Recall, AUC	Stacking Based Model	94%, 96%, 96%, 98%
[1]	None	Random Forest, Decision Tree	None	70:30	Accuracy	Random Forest	80%
[27]	None	LightGBM, XGBoost	None	91:9	Accuracy, Error Rate	LightGBM	80%, 20%

[29]	Cluster-based under-sampling	Logistic Regression, SVM, XGBoost, Group Method of Data Handling	None	80:20	Accuracy, AUC	XGBoost	90%, 94%
[33]	SMOTE	Logistic Regression, Decision Tree, SVM, ADA-Boost, MLP (one-hidden layer), MLP (three hidden layer)	None	80:20	Accuracy	MLP (three hidden layer)	93%

The proposed method showed a higher accuracy when compared to baseline results. [1] achieved an accuracy score of 80% using random forest on a 70:30 split. Ref. [27] reached an accuracy of 80% with the LightGBM on a 91:9 split, while [29] attained a 90% accuracy and a 94% AUC with XGBoost using Cluster-based under-sampling on an 80:20 split. Furthermore, ref. [33] achieved an accuracy of 93% with the MLP using the SMOTE. In contrast, our proposed model performed better with 94% accuracy, 96% precision, 96% recall and 98% AUC, which shows how well the model predicts credit default risk. The combination of identifying the optimal method to use at each stage, including the advanced class imbalance method (the SMOTE + ENN) and the ensemble technique (stacking), significantly contributed to this improved performance.

Additionally, SHAPs (SHapley Additive exPlanations) with the XGBoost model were used to analyse and explain the features as shown in Table A3, with the features resulting in negative values, contributing to the predictions being lower, thereby increasing the likelihood of defaults. For instance, a negative ‘int_rate’ means that higher interest rate pushed the prediction towards default, while positive values contributed to the predictions being higher. For instance, positive ‘fico_range_low’ means that higher credit scores moved the prediction away from being classed as a default. With the proposed model developed, tested and evaluated, this research shows how well the used methodology worked, as testing the techniques to identify the most suitable one contributed to the overall performance of the models; additionally, combining the predictions from weaker classifiers contributed as well.

5. Conclusions and Recommendations

In this study, we experimented with data pre-processing techniques such as feature normalisation, class imbalance handling and feature extraction to determine the optimal solution of feature pre-processing for the LendingClub data. Additionally, different machine learning and deep learning models were explored to predict the likelihood of loan default: random forest, decision tree, Support Vector Machines (SVMs), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (ADABoost) and Multi-Layered Perceptrons (MLPs) with a three hidden layer. We also experimented with different ensemble techniques such as voting and stacking for model optimisation. The contributions of this research are outlined below:

1. Developed Model: The stacking ensemble model that combines the predictions from all the models were developed and identified as the best performing model. The proposed model is capable of precisely gauging default risk, with a recall of 95.5%, which is the true default rate.
2. Identifying Suitable Techniques: The performance of diverse methods was tested at different stages to identify suitable methods to make use of, with the identification of suitable methods for data pre-processing (data normalisation and outlier handling techniques) tested and documented in Section 3.3. Additionally, the exploration of

diverse sampling methods to handle the issue of class imbalance, with the performance of each method, was documented in Sections 3 and 4.

3. An explainable model: SHapley Additive exPlanations (SHAPs) were used in creating an explainable model that shows the contribution of each feature in making the predictions, as shown in Table A3, which explains why each application or borrower is classed as default or non-default.

Moreover, when comparing the results from this study with the baseline results as shown in Section 4.2, this study achieved a higher accuracy (93.7%) with the proposed model. This study can help lending platforms with the reduction in credit default. It can also help in further research, as the performance of diverse techniques and models were explored and documented.

One of the limitations of this study is data availability. The data collection process was a bit constrained due to the limited large financial dataset being publicly available; moreover, having a recent dataset would have been beneficial in this project. Additionally, the techniques and models used were memory intensive, which influenced the decision to make use of XGBoost as a test model similar to the approach used by [42], and, since this study was carried out under a time constraint, this action further put a limitation on the test and refinements that could have been performed.

Despite the identified limitations, they do not detract from the contribution; rather, they serve as an area for improvement and exploration. Furthermore, the techniques and findings obtained from this study may create interesting avenues for future research. For instance, in the selection of suitable techniques to use, instead of using XGBoost as a test model, each technique can be tested on each model to see if different techniques work better with different models. Additionally, the methodology can be tested on other credit datasets, to further validate the selected framework.

Author Contributions: Conceptualization, A.A. and O.S.; methodology, A.A. and O.S.; software, A.A. and O.S.; validation, O.S., J.P., B.O. and O.O.; formal analysis, A.A. and O.S.; investigation, A.A. and O.S.; resources, O.S.; data curation, A.A.; writing—original draft preparation, A.A. and O.S.; writing—review and editing, O.S., J.P., B.O. and O.O.; supervision, O.S.; project administration, J.P. and O.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The LendingClub dataset used in this research is available in Kaggle, a data science repository. The code can be accessed from GitHub: Credit Default Risk.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Feature description.

Features	Description	Remark
id	ID for the loan	Excluded
loan_amnt	Loan applied for by the borrower	Included
funded_amnt	Amount committed to the loan	Excluded: Post-Loan
funded_amnt_inv	Amount committed by the investors	Excluded: Post-Loan
term	Loan term, either 36 or 60 months	Included
int_rate	Interest rate on the loan	Included
instalment	The monthly payment owed by the borrower if the loan originates	Excluded: Merged with annual_inc
grade	Loan grade	Excluded: Assigned by LendingClub

sub_grade	Loan subgrade	Excluded: Assigned by LendingClub
emp_title	Borrower's job title	Excluded
emp_length	Length of employment: 0: < 1 year 10: ≥ 10 years	Excluded
home_ownership	Type of home owned by the borrower	Included
annual_inc	Income provided by the borrower	Included
verification_status	If the income is verified by LendingClub	Included
loan_status	The status of the loan	Included: Converted
pymnt_plan	Payment plan for the loan	Excluded: Post-Loan
url	The page's URL	Excluded: Ambiguous
purpose	Purpose of the loan	Included
title	Title of the loan	Excluded
zip_code	Borrower's zip code	Excluded: PPI
addr_state	Borrower's state	Included: Converted
dti	The debt-to-income ratio using the total debt payments (per month) to the total debt obligations, (minus the mortgage, loan) divided by the income (per month)	Included
delinq_2yrs	Incidences of delinquency in the past 2 yrs that are 30+ days past due	Excluded: Post-Loan
fico_range_low	Borrower's lower FICO boundary range	Included
fico_range_high	Borrower's higher FICO boundary range	Excluded: Correlation
inq_last_6mths	Last 6 months inquiries minus inquiries on mortgage and auto	Excluded
open_acc	Total credit lines opened by the borrower	Included
pub_rec	Number of derogatory public records	Included
revol_bal	Total credit revolving balance	Included
revol_util	Amount of credit the borrower is using, which is relative to all revolving credit	Excluded
total_acc	Total credit lines owned by the borrower	Included
initial_list_status	Listing status of the loan	Excluded
out_prncp	Principal left for the funded amount	Excluded: Post-Loan
out_prncp_inv	Principal left for the investors' funded amount	Excluded: Post-Loan
total_pymnt	Total payments on the funded amount	Excluded: Post-Loan
total_pymnt_inv	Total payments on the investors' funded amount	Excluded: Post-Loan
total_rec_prncp	Total principal paid	Excluded: Post-Loan
total_rec_int	Total interest paid	Excluded: Post-Loan
total_rec_late_fee	Total late fees paid	Excluded: Post-Loan
recoveries	Post gross charge-off recovery	Excluded: Post-Loan
collection_recovery_fee	Post collection charge-off fee	Excluded: Post-Loan
last_pymnt_d	Last payment date (month)	Excluded: Post-Loan
last_pymnt_amnt	Last payment amount paid	Excluded: Post-Loan
last_credit_pull_d	Recent credit pulled by LendingClub for the loan	Excluded: Post-Loan
last_fico_range_high	The last borrower's upper FICO boundary range pulled by LendingClub	Excluded: Post-Loan
last_fico_range_low	The last borrower's lower FICO boundary range pulled by LendingClub	Excluded: Post-Loan
collections_12_mths_ex_med	Collections in 12 months minus the medical collections	Excluded: Post-Loan

policy_code	publicly available policy_code = 1 new products not publicly available policy_code = 2	Excluded: Single value column
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers	Excluded
acc_now_delinq	The number of accounts on which the borrower is now delinquent	Excluded: Post-Loan
tot_coll_amt	Total collection amounts ever owed	Excluded: Post-Loan
tot_cur_bal	Total current balance of all accounts	Excluded: Post-Loan
total_rev_hi_lim	Description not found	Excluded: Unknown
acc_open_past_24mths	Number of trades opened in past 24 months	Excluded: Post-Loan
avg_cur_bal	Average current balance of all accounts	Included
bc_open_to_buy	Total open to buy on revolving bankcards	Included
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts	Excluded
chargeoff_within_12_mths	Number of charge-offs within 12 months	Excluded: Post-Loan
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent	Included
mo_sin_old_il_acct	Months since oldest bank instalment account opened	Excluded
mo_sin_old_rev_tl_op	Months since oldest revolving account opened	Included
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened	Included
mo_sin_rcnt_tl	Months since most recent account opened	Included
mort_acc	Number of mortgage accounts	Included
mths_since_recent_bc	Months since most recent bankcard account opened	Included
mths_since_recent_inq	Months since most recent inquiry	Excluded
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due	Excluded
num_actv_bc_tl	Number of currently active bankcard accounts	Included
num_actv_rev_tl	Number of currently active revolving trades	Included
num_bc_sats	Number of satisfactory bankcard accounts	Included
num_bc_tl	Number of bankcard accounts	Included
num_il_tl	Number of instalment accounts	Excluded
num_op_rev_tl	Number of open revolving accounts	Included
num_rev_accts	Number of revolving accounts	Excluded
num_rev_tl_bal_gt_0	Number of revolving trades with balance >0	Excluded
num_sats	Number of satisfactory accounts	Excluded
num_tl_120dpd_2m	Number of accounts currently 120 days past due (updated in past 2 months)	Excluded: Post-Loan
num_tl_30dpd	Number of accounts currently 30 days past due (updated in past 2 months)	Excluded: Post-Loan
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months	Excluded: Post-Loan
num_tl_op_past_12m	Number of accounts opened in past 12 months	Excluded: Post-Loan
pct_tl_nvr_dlq	Percent of trades never delinquent	Excluded
percent_bc_gt_75	Percentage of all bankcard accounts > 75% of limit	Included
pub_rec_bankruptcies	Number of public record bankruptcies	Included
tax_liens	Number of tax liens	Excluded
tot_hi_cred_lim	Total high credit/credit limit	Included
total_bal_ex_mort	Total credit balance minus mortgage	Excluded
total_bc_limit	Total bankcard limit	Excluded
total_il_high_credit_limit	Total instalment limit	Excluded
hardship_flag	Description not found	Unknown
disbursement_method	Description not found	Unknown

debt_settlement_flag	Description not found	Unknown
issue_d_yr	Loan issue date (year)	Excluded: Post-Loan
earliest_cr_line_yr	The earliest credit line reported for the borrower (year)	Excluded: Correlation

Table A2. Hyperparameter grids for model tuning.

Model	Hyperparameters	Values
Random Forest	max_depth	[10, 20]
	min_samples_split	[1, 4]
	min_samples_leaf	[2, 5]
	n_estimators	[100, 200, 500]
Decision Tree	max_depth	[15, 20, 25]
	max_leaf_nodes	[20, 50]
	min_impurity_decrease	[0, 0.01]
	min_samples_split	[2, 5]
	min_samples_leaf	[1, 2, 4]
	criterion	[‘gini′, ‘entropy′]
	class_weight	[‘balanced′, None]
SVM	C	[1, 10]
	kernel	[‘linear′, ‘rbf′, ‘poly′]
	gamma	[‘scale′, ‘auto′, 1]
	degree	[2, 3]
XGBoost	colsample_bytree	[0.7, 0.9]
	max_depth	[3, 6, 10, 20]
	learning_rate	[0.1, 0.2]
	n_estimators	[200, 300, 500]
	subsample	[1, 2]
	reg_alpha	[1]
	reg_lambda	[1.5]
ADABoost	n_estimators	[100, 200, 300]
	learning_rate	[0.15, 0.2]
MLP	activation	[‘relu′, ‘tanh′]
	batch_size	[‘auto′]
	early_stopping	[True]
	hidden_layer_sizes	[(100, 100, 100), (150, 150, 150)]
	alpha	[0.001, 0.002]
	learning_rate	[‘adaptive′, ‘constant′]
	solver	[‘adam′, ‘sgd′]

Table A3. Feature contribution.

Prediction Results					
ID:	865040	1555274	449256	501313	508884
Status:	Default	Default	Default	Non-Default	Non-Default
Feature Contribution					
int_rate	−0.9235	0.1532	−0.1252	0.6710	0.7091
term	−0.3001	−0.3291	−0.2929	−0.4548	−0.5768
fico_range_low	0.1290	0.0504	−0.3363	0.1065	−0.1803
dti	0.4284	−0.4859	−0.3480	0.0451	0.4179
loan_amnt_dti	0.0517	0.0811	−0.0047	0.0663	0.0750
annual_inc_installment:	0.0011	−0.2472	−0.2640	0.0040	−0.1360

bc_open_to_buy	0.0246	−0.0556	−0.0025	0.0364	0.0064
avg_cur_bal	−0.0456	−0.0184	0.0624	−0.0297	−0.0748
tot_hi_cred_lim	−0.0271	0.0088	0.0685	0.0064	−0.1719
percent_bc_gt_75	1.0439	1.2103	1.3094	0.5351	0.1701
num_actv_rev_tl	−0.2787	−0.1123	−0.3886	−0.1407	−0.0883
loan_amnt	−0.0648	−0.2035	0.2745	−0.0134	−0.2907
total_bc_limit	0.0304	0.1285	−0.0521	−0.1556	0.0642
mort_acc	−0.0928	−0.1118	−0.1172	−1.1850	−0.5271
home_ownership_MORTGAGE	0.0250	0.0895	0.1417	−0.1235	−0.1263
verification_status_Verified	−0.0960	−0.0439	−0.0346	−0.0851	0.1597
home_ownership_RENT	0.0818	0.1043	−0.1973	−0.1385	−0.2050
annual_inc	−0.0110	−0.0312	−0.0881	−0.0814	−0.0116
total_rev_hi_lim	0.0697	0.1249	−0.0004	−0.1864	−0.0205
mo_sin_rcnt_tl	0.1624	0.1267	0.1272	0.0026	0.0109
mths_since_recent_bc	0.1239	−0.5703	0.2025	−0.4710	−0.0046
mo_sin_rcnt_rev_tl_op	0.1539	0.0383	0.0950	0.0473	−0.0932
annual_inc_binned_Low	0.0463	0.0340	−0.0008	−0.0401	0.0505
mo_sin_old_rev_tl_op	0.0286	0.0686	0.0593	−0.1941	−0.1120
num_actv_bc_tl	−0.2203	−0.3038	−0.2853	−0.0558	−0.2246
purpose_credit_card	0.0523	0.0309	0.0200	0.0375	0.0565
purpose_debt_consolidation	−0.0543	−0.0278	−0.1285	−0.1087	−0.0390
num_op_rev_tl	0.2198	0.2315	0.1212	0.1651	0.0660
open_acc	0.0202	0.2411	0.0883	0.0907	0.2031
pub_rec	−0.0108	0.0214	0.0047	0.0118	−0.0085
purpose_small_business	−0.0019	−0.0033	−0.0046	−0.0037	−0.0027
verification_status_Source Verified	−0.0145	−0.2025	0.0056	−0.0941	−0.1220
pub_rec_bankruptcies	−0.0099	−0.0014	−0.0009	−0.0039	−0.0064
delinq_2yrs	0.1062	−0.0984	−0.0384	−0.0562	0.1431
revol_bal	0.0037	−0.0216	0.0848	0.0135	0.0657
region_West	−0.0075	0.0516	−0.2981	−0.0250	−0.0322
purpose_home_improvement	−0.0029	−0.0005	−0.0054	−0.0081	−0.0018
num_bc_sats	0.4341	0.6630	0.4305	0.4034	0.0938
revol_bal_binned_Very Low	0.0175	0.0072	−0.0849	−0.0346	0.0011
pct_tl_nvr_dlq	0.0440	0.0280	−0.0540	0.0231	−0.0063
num_bc_tl	−0.0849	−0.0105	−0.1792	−0.0441	−0.0190
annual_inc_binned_Very Low	0.0005	−0.0012	0.0002	−0.0001	−0.0011
application_type_Joint App	−0.0005	−0.0019	−0.0053	−0.0698	−0.0009
total_acc	0.0227	0.0935	0.0834	−0.0461	0.0854
revol_bal_binned_Medium	−0.0020	−0.0014	−0.0032	−0.0011	−0.0032
region_SouthEast	−0.0004	−0.0194	−0.0257	0.0106	−0.0222
annual_inc_binned_Medium	−0.0210	0.0002	−0.0368	−0.0064	0.0117
annual_inc_dti	0.0061	−0.0343	−0.0108	0.0032	0.0064

References

1. Madaan, M.; Kumar, A.; Keshri, C.; Jain, R.; Nagrath, P. Loan default prediction using decision trees and random forest: A comparative study. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; Volume 1022, p. 012042. https://doi.org/10.1088/1757-899x/1022/1/012042.
2. Ivashina, V.; Scharfstein, D. Bank lending during the financial crisis of 2008. *J. Financial Econ.* **2010**, *97*, 319–338. https://doi.org/10.1016/j.jfineco.2009.12.001.
3. Brunnermeier, M.K. Deciphering the Liquidity and Credit Crunch 2007–2008. *J. Econ. Perspect.* **2009**, *23*, 77–100, https://doi.org/10.1257/jep.23.1.77.

4. Acharya, V.; Philippon, T.; Richardson, M.; Roubini, N. *The Financial Crisis of 2007–2009: Causes and Remedies*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
5. Switzer, L.N.; Wang, J. Default Risk Estimation, Bank Credit Risk, and Corporate Governance. In *Financial Markets, Institutions & Instruments*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 22, pp. 91–112. https://doi.org/10.1111/fmii.12005.
6. Chen, N.; Ribeiro, B.; Chen, A. Financial credit risk assessment: A recent review. *Artif. Intell. Rev.* **2015**, *45*, 1–23. https://doi.org/10.1007/s10462-015-9434-x.
7. Duffie, D. *Measuring Corporate Default Risk*; Oxford University Press: New York, NY, USA, 2011.
8. Thomas, L.C. A survey of credit and behavioural scoring: Forecasting financial risk of lending to consumers. *Int. J. Forecast.* **2000**, *16*, 149–172. https://doi.org/10.1016/s0169-2070(00)00034-0.
9. Nwulu, N.I.; Oroja, S.; Ilkan, M. A Comparative Analysis of Machine Learning Techniques for Credit Scoring. *Inf. Int. Interdiscip. J.* **2012**, *15*, 4129–4145.
10. Radović, O.; Marinković, S.; Radojičić, J. Credit scoring with an ensemble deep learning classification methods—Comparison with tradicional methods. *Facta Univ. Series: Econ. Organ.* **2021**, *18*, 29–43. https://doi.org/10.22190/fueo201028001r.
11. Lessmann, S.; Baesens, B.; Seow, H.-V.; Thomas, L.C. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *Eur. J. Oper. Res.* **2015**, *247*, 124–136. https://doi.org/10.1016/j.ejor.2015.05.030.
12. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 160. https://doi.org/10.1007/s42979-021-00592-x.
13. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. https://doi.org/10.1023/a:1010933404324.
14. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. https://doi.org/10.1038/nature14539.
15. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. https://doi.org/10.1613/jair.953.
16. Guyon, I.; Elisseeff, A. An introduction to feature selection. In *Feature Extraction*; Springer: Berlin/Heidelberg, Germany, **2003**; pp. 1–25. https://doi.org/10.1007/978-3-540-35488-8_1.
17. Rizopoulos, D. Max Kuhn and Kjell Johnson. Applied Predictive Modeling. New York, Springer. *Biometrics* **2018**, *74*, 383. https://doi.org/10.1111/biom.12855.
18. Zhang, C.; Ma, Y. *Ensemble Machine Learning: Methods and Applications*; Springer Verlag: Heidelberg, Germany, 2012; pp. 1–329. https://doi.org/10.1007/9781441993267.
19. Ntiamoah, E.B.; Oteng, E.; Opoku, B.; Siaw, A. Loan default rate and its impact on profitability in financial institutions. *Res. J. Financ. Account.* **2014**, *5*, 67–72.
20. Amzile, K.; Habachi, M. Assessment of Support Vector Machine performance for default prediction and credit rating. *Banks Bank Syst.* **2022**, *17*, 161–175. https://doi.org/10.21511/bbs.17(1).2022.14.
21. Xu, J.; Lu, Z.; Xie, Y. Loan default prediction of Chinese P2P market: A machine learning methodology. *Sci. Rep.* **2021**, *11*, 18759. https://doi.org/10.1038/s41598-021-98361-6.
22. Dzik-Walczak, A.; Heba, M. An implementation of ensemble methods, logistic regression, and neural network for default prediction in Peer-to-Peer lending. *Zb. Rad. Èkon. Fak. U Rijeci-Proceedings Rij. Fac. Econ.* **2021**, *39*, 163–197. https://doi.org/10.18045/zbefri.2021.1.163.
23. Bühlmann, P. Bagging, Boosting and Ensemble Methods. In *Handbook of Computational Statistics*; Springer Nature: Berlin, Germany, 2011; pp. 985–1022. https://doi.org/10.1007/978-3-642-21551-3_33.
24. Bühlmann, P.; Hothorn, T. Boosting Algorithms: Regularization, Prediction and Model Fitting. *Stat. Sci.* **2007**, *22*, 477–505. https://doi.org/10.1214/07-sts242.
25. Dietterich, T.G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Mach. Learn.* **2000**, *40*, 139–157. https://doi.org/10.1023/a:1007607513941.
26. Alzubi, J.; Nayyar, A.; Kumar, A. Machine Learning from Theory to Algorithms: An Overview. *J. Physics Conf. Ser.* **2018**, *1142*, 12012. https://doi.org/10.1088/1742-6596/1142/1/012012.
27. Mahesh, B. Machine Learning Algorithms—A Review. *Int. J. Sci. Res. (IJSR)* **2018**, *9*, 381–386. https://doi.org/10.21275/ART20203995.
28. Ma, X.; Sha, J.; Wang, D.; Yu, Y.; Yang, Q.; Niu, X. Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. *Electron. Commer. Res. Appl.* **2018**, *31*, 24–39. https://doi.org/10.1016/j.elerap.2018.08.002.
29. Chang, Y.-C.; Chang, K.-H.; Wu, G.-J. Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Appl. Soft Comput.* **2018**, *73*, 914–920. https://doi.org/10.1016/j.asoc.2018.09.029.
30. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. https://doi.org/10.1007/s42979-021-00815-1.
31. Sharifani, K.; Amini, M. Machine learning and deep learning: A review of methods and applications. *World Inf. Technol. Eng. J.* **2023**, *10*, 3897–3904.
32. Duan, J. Financial system modeling using deep neural networks (DNNs) for effective risk assessment and prediction. *J. Frankl. Inst.* **2019**, *356*, 4716–4731. https://doi.org/10.1016/j.jfranklin.2019.01.046.
33. Jumaa, M.; Saqib, M.; Attar, A. Improving Credit Risk Assessment through Deep Learning-based Consumer Loan Default Prediction Model. *Int. J. Finance Bank. Stud.* **2023**, *12*, 85–92. https://doi.org/10.20525/ijfbs.v12i1.2579.

34. Nordhausen, K. Ensemble Methods: Foundations and Algorithms by Zhi-Hua Zhou. *Int. Stat. Rev.* **2013**, *81*, 470. https://doi.org/10.1111/insr.12042_10.
35. Seni, G.; Elder, J.F. Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. *Synth. Lect. Data Min. Knowl. Discov.* **2010**, *2*, 1–126. https://doi.org/10.2200/s00240ed1v01y200912dmk002.
36. Opitz, D.; Maclin, R. Popular Ensemble Methods: An Empirical Study. *J. Artif. Intell. Res.* **1999**, *11*, 169–198.
37. Yu, L.; Yue, W.; Wang, S.; Lai, K. Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Syst. Appl.* **2010**, *37*, 1351–1360. https://doi.org/10.1016/j.eswa.2009.06.083.
38. Alam, T.M.; Shaukat, K.; Hameed, I.A.; Luo, S.; Sarwar, M.U.; Shabbir, S.; Li, J.; Khushi, M. An Investigation of Credit Card Default Prediction in the Imbalanced Datasets. *IEEE Access* **2020**, *8*, 201173–201198. https://doi.org/10.1109/ACCESS.2020.3033784.
39. Uddin, N.; Ahamed, K.U.; Uddin, A.; Islam, M.; Talukder, A.; Aryal, S. An ensemble machine learning based bank loan approval predictions system with a smart application. *Int. J. Cogn. Comput. Eng.* **2023**, *4*, 327–339. https://doi.org/10.1016/j.ijcce.2023.09.001.
40. Wang, Y.; Wang, M.; Pan, Y.; Chen, J. Joint loan risk prediction based on deep learning-optimized stacking model. *Eng. Rep.* **2023**, *6*, e12748. https://doi.org/10.1002/eng2.12748.
41. Li, X.; Ergu, D.; Zhang, D.; Qiu, D.; Cai, Y.; Ma, B. Prediction of loan default based on multi-model fusion. *Procedia Comput. Sci.* **2022**, *199*, 757–764. https://doi.org/10.1016/j.procs.2022.01.094.
42. Chang, A.-H.; Yang, L.-K.; Tsaih, R.-H.; Lin, S.-K. Machine learning and artificial neural networks to construct P2P lending credit-scoring model: A case using Lending Club data. *Math. Biosci. Eng.* **2022**, *6*, 303–325. https://doi.org/10.3934/qfe.2022013.
43. Moula, F.E.; Guotai, C.; Abedin, M.Z. Credit default prediction modeling: An application of support vector machine. *Risk Manag.* **2017**, *19*, 158–187. https://doi.org/10.1057/s41283-017-0016-x.
44. Acharya, A.S.; Prakash, A.; Saxena, P.; Nigam, A. Sampling: Why and how of it? *Indian J. Med. Spéc.* **2013**, *4*, 330–333. https://doi.org/10.7713/ijms.2013.0032.
45. Guo, W.; Zhou, Z.Z. A comparative study of combining tree-based feature selection methods and classifiers in personal loan default prediction. *J. Forecast.* **2022**, *41*, 1248–1313. https://doi.org/10.1002/for.2856.
46. Cain, M.K.; Zhang, Z.; Yuan, K.-H. Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation. *Behav. Res. Methods* **2016**, *49*, 1716–1735. https://doi.org/10.3758/s13428-016-0814-1.
47. Schober, P.; Boer, C.; Schwarte, L.A. Correlation Coefficients: Appropriate Use and Interpretation. *Anesth. Analg.* **2019**, *126*, 1763–1768. https://doi.org/10.1213/ane.0000000000002864.
48. Tsagris, M.; Pandis, N. Multicollinearity. *Am. J. Orthod. Dentofac. Orthop.* **2021**, *159*, 695–696. https://doi.org/10.1016/j.ajodo.2021.02.005.
49. Watson, R. Quantitative research. *Nurs. Stand.* **2015**, *29*, 44–48. https://doi.org/10.7748/ns.29.31.44.e8681.
50. García, S.; Ramírez-Gallego, S.; Luengo, J.; Benítez, J.M.; Herrera, F. Big data preprocessing: Methods and prospects. *Big Data Anal.* **2016**, *1*, 9. https://doi.org/10.1186/s41044-016-0014-0.
51. Alexandropoulos, S.-A.N.; Kotsiantis, S.B.; Vrahatis, M.N. Data preprocessing in predictive data mining. *Knowl. Eng. Rev.* **2019**, *34*, e1. https://doi.org/10.1017/s026988891800036x.
52. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2023**, *244*, 122778. https://doi.org/10.1016/j.eswa.2023.122778.
53. Baesens, B.; Van Vlasselaer, V.; Verbeke, W. *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
54. Dash, C.S.K.; Behera, A.K.; Dehuri, S.; Ghosh, A. An outliers detection and elimination framework in classification task of data mining. *Decis. Anal. J.* **2023**, *6*, 100164. https://doi.org/10.1016/j.dajour.2023.100164.
55. Ramsauer, A.; Baumann, P.M.; Lex, C. The Influence of Data Preparation on Outlier Detection in Driveability Data. *SN Comput. Sci.* **2021**, *2*, 222. https://doi.org/10.1007/s42979-021-00607-7.
56. Milli, M.E.F.; Aras, S.; Kocakoç, I.D. Investigating the Effect of Class Balancing Methods on the Performance of Machine Learning Techniques: Credit Risk Application. *İzmir Yönetim Derg.* **2024**, *5*, 55–70. https://doi.org/10.56203/iyd.1436742.
57. Megahed, F.M.; Chen, Y.-J.; Megahed, A.; Ong, Y.; Altman, N.; Krzywinski, M. The class imbalance problem. *Nat. Methods* **2021**, *18*, 1270–1272. https://doi.org/10.1038/s41592-021-01302-4.
58. Namvar, A.; Siami, M.; Rabhi, F.; Naderpour, M. Credit risk prediction in an imbalanced social lending environment. *Int. J. Comput. Intell. Syst.* **2018**, *11*, 925–935. https://doi.org/10.2991/ijcis.11.1.70.
59. Swana, E.F.; Doorsamy, W.; Bokoro, P. Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset. *Sensors* **2022**, *22*, 3246. https://doi.org/10.3390/s22093246.
60. Chen, Y.-R.; Leu, J.-S.; Huang, S.-A.; Wang, J.-T.; Takada, J.-I. Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets. *IEEE Access* **2021**, *9*, 73103–73109. https://doi.org/10.1109/access.2021.3079701.
61. Zhu, Y.; Hu, Y.; Liu, Q.; Liu, H.; Ma, C.; Yin, J. A Hybrid Approach for Predicting Corporate Financial Risk: Integrating SMOTE-ENN and NGBoost. *IEEE Access* **2023**, *11*, 111106–111125. https://doi.org/10.1109/access.2023.3323198.
62. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature Selection: A data perspective. *ACM Comput. Surv.* **2018**, *50*, 1–45. https://doi.org/10.1145/3136625.
63. Rtayli, N.; Enneya, N. Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *J. Inf. Secur. Appl.* **2020**, *55*, 102596. https://doi.org/10.1016/j.jisa.2020.102596.

64. Aria, M.; Cuccurullo, C.; Gnasso, A. A comparison among interpretative proposals for Random Forests. *Mach. Learn. Appl.* **2021**, *6*, 100094. https://doi.org/10.1016/j.mlwa.2021.100094.
65. Schapire, R. *The Boosting Approach to Machine Learning An Overview*; Springer Nature: Berlin, Germany, 2003. Available online: https://www.cs.princeton.edu/courses/archive/spring07/cos424/papers/boosting-survey.pdf (accessed on 19 April 2019).
66. Goodfellow, I. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
67. Rojarath, A.; Songpan, W. Probability-Weighted Voting Ensemble Learning for Classification ModelProbability-Weighted Voting Ensemble Learning for Classification Model. *J. Adv. Inf. Technol.* **2020**, *11*, 217–227. https://doi.org/10.12720/jait.11.4.217-227.
68. Nishat, M.M.; Faisal, F.; Ratul, I.J.; Al-Monsur, A.; Ar-Rafi, A.M.; Nasrullah, S.M.; Reza, T.; Khan, R.H. A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers with SMOTE-ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset. *Sci. Program.* **2022**, *2022*, 3649406. https://doi.org/10.1155/2022/3649406.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.