Ensemble based Machine Learning Algorithm for Loan Default Risk Prediction

Abisola Akinjole , Olamilekan Shobayo * , Jumoke Popoola , Obinna Okoyeigbo , Bayode Ogunleye

Posted Date: 11 September 2024

doi: 10.20944/preprints202409.0923.v1

Keywords: Credit default prediction; deep learning; ensemble learning; machine leaning

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 and to also address the issue of data imbalance. In this research, we conducted a comparative analysis of random forest, decision tree, SVM (Support Vector Machine), XGBoost (eXtreme Gradient Boosting), ADABoost (ADAptive Boosting) and multilayered perceptron with three-hidden layers, to predict credit default using loan data from LendingClub. Additionally, we also combined the model predictions using voting and stacking ensemble methods to enhance the models' performance. Furthermore, various sampling techniques was explored to handle the issue of class imbalance observed in the dataset, with the result showing that the balanced data performs better than the imbalanced data. 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 leaning

1. Introduction

Numerous financial institutions, like banks and lending platforms have relied on the interest and fees from loans as a source of revenue [1], so, to maintain their financial strength and profitability, banks and lending platforms must ensure that the loan payments are made and that borrowers do not default on their payments. Although the economic breakdown in the late 2000s, specifically the financial crisis in 2008, was caused by many factors, it can also highlight the impact of lending to individuals or businesses who are unable to repay their debts [2,3], which is why predicting credit default risk is important, as it can help lenders avoid having large losses, mitigate financial crises and further maintain public trust in the banking system [4]. Therefore, credit default risk is the likelihood that a borrower will fail to fulfil their payment obligations [5-7].

Hence, given the need to accurately predict credit default risk, this study aims to compare various machine learning models for this prediction. This study further combines these models with boosting classifiers, such as eXtreme Gradient Boosting (XGBoost) and ADAptive Boosting (ADABoost), to determine if the combined or ensembled model will perform better than the individual models. By evaluating how accurate the models are at predicting defaults, this project seeks to determine the most effective technique to accurately assess credit default risk using dataset from LendingClub [8], a financial lending company, which includes detailed information on every

 doi:10.20944/preprints202409.0923.v

loan issued from 2007 to 2018, including the comprehensive list of borrowers' characteristics, such as their annual income, amount borrowed, loan purpose, debt-to-income ratio, credit history, credit score and other relevant variables.

Furthermore, literature synthesis evidenced that a common problem in credit default prediction is the issue of class imbalance. Therefore, this paper handled class imbalance by testing various techniques, including over-sampling techniques such as Random Over-Sampling (ROS), Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN), under-sampling techniques such as Random Under-Sampling (RUS) and Tomek Links, as well as a combination of sampling techniques such as SMOTE Tomek and SMOTE with the Edited Nearest Neighbours (SMOTE + ENN).

The methodology followed by [9] is the closest to the methodology used in this work and we were able to achieve higher result compared to the accuracy obtained by the study [9], as well as in comparison to these related studies [10-12]. 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), finally, the conclusion and recommendations (Section 5).

2. Related Works

Authors in [13] showed that **loan** default rate and profitability are highly correlated and thus, models that can be used to accurately predict **loan** default is required, which is why machine learning techniques have been taken advantage of, as they have significantly improved the performance of predictability in various financial applications [14-17]. In the context of credit default prediction, the data used contains various borrowers' characteristics as inputs and the target variable, therefore, this study does not consider the use of unsupervised learning algorithms for the prediction of default.

Several supervised machine learning models such as logistic regression, random forest, decision tree, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Extreme Gradient Boosting (XGBoost) and Adaptive Boosting (ADABoost) have been used for credit default prediction. However, very few studies have thoroughly addressed the issue of class imbalance which limits the generalisation of the models. For example, the study of [14] compared SVM and logistic regression models to predict credit default, using data from the portfolio of a Portuguese bank. Their study achieved good results using SVM, however, the size of the dataset (1992 non-defaulting customers and 1008 defaulting customers) used may bring about some limitations. Similarly, authors in [1] made use of random forest and decision tree for their prediction. They showed random forest performed better than the decision tree with 80% accuracy. However, it is worth stating that their study evaluated the models mainly with accuracy. Unfortunately, the evaluation metric, accuracy is not sufficient for evaluation in the presence of class imbalance as the models are biased to the majority class, which in this case is the non-defaulters.

Similar to random forest used in [1], there are some machine learning models that are derived from the combination of predictions from multiple models using techniques like boosting, which is an ensemble technique that combines weak learners to create stronger algorithms [18]. For example, in [17], boosting classifiers, Light Gradient Boosting Machine (LightGBM) and XGBoost were used for the prediction of loan default using LendingClub data, from July 2007 to June 2017. This study had an interesting approach to cleaning the data, as in this study, two separate cleaning processes, multi-observational and multi-dimensional methods were used to identify and correct inconsistencies, observing that multi-observational was the superior method. With an accuracy of 80.1% and an error rate of 19.9%, the authors noted that LightGBM outperformed the XGBoost classifier in the prediction of loan defaults.

The prevalence of class imbalance in credit data was observed by [19], which is an issue that occurs when the classes in the dataset are not represented equally. In loan dataset, the non-default loans are usually more than the defaulted loans, and if not handled properly, it can cause the model to perform poorly on the minority class. [19] proposed XGBoost classifier to build credit risk assessment models and made use of cluster-based under-sampling to process the imbalanced data.

 doi:10.20944/preprints202409.0923.v

Accuracy and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used as validation metrics, as the proposed model was compared with other models including logistic regression and SVM, with XGBoost outperforming the other models with an accuracy of 90.0% against 69.7% and 76.9% accuracy scores for logistic regression and SVM respectively, and AUC values of 0.94 against 0.77 and 0.87. Although this study achieved an impressive result with the proposed model, the dataset size might pose a limitation, as 6,271 records were used in this research. 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. Furthermore, the study of [20] made use of a deep learning model to predict consumer loan default using a dataset with 1,000 observations gotten from the response to a questionnaire created by the authors. This study used Keras, a neural network library which 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, [20] employed stratified random sampling.

The assessment and prediction of lending risk using MLP with three-hidden layers was presented by [21] with the LendingClub dataset used for the model development and evaluation. The authors classified the output variable into three categories using TensorFlow: safe loans, risky loans and bad loans, with majority of the data belonging to safe loan. The class imbalance issue was handled using Synthetic Minority Oversampling Technique (SMOTE). Furthermore, accuracy served as the measure of the model's performance when compared with other models. The deep learning model with an accuracy of 93.2% outperformed other models including logistic regression (77.1%), decision tree (50.5%), linear SVM (78.9%), ADABoost (85.2%) and MLP with one-hidden layer (62.8%). The other performance metrics used were sensitivity (75.6%) and specificity (72.2%). In this study, no under-sampling or hybrid method was used to handle class imbalance.

Authors in [15] used Artificial Neural Network (ANN), random forest, XGBoost, and Gradient Boosting Regression Tree (GBRT). To address the issue of class imbalance, SMOTE was employed. In terms of the prediction models, GBRT constructs an ensemble of weak prediction trees to form a stronger predictor, while random forest obtains predictions by averaging the predictions from multiple individually trained decision trees. On the other hand, ANN is based on a mathematical process that can process nonlinear relationships between the independent variables and dependent variable. [15] showed that random forest model performed better than the other models when using metrics such as accuracy, kappa, precision, recall and F1-score to evaluate the performance of the models. The study of [16] used logistic regression and MLP models to predict credit default. Gini coefficient was used for feature selection, it measures the separation capability of the model. Subsequently, they combine the models with two ensemble techniques, the first method was averaging the probabilities obtained from both models to get the final predictions (bagging), while the second method was to input the probabilities into logistic regression (meta-model) to produce a final probability value. Bagging ensemble model performed better than all the other models, with the performance of each model evaluated using AUC, Gini index, KS, accuracy, error ratio, Positive Predictive Value (PPV), and Negative Predictive Value (NPV).

Authors in [9] used diverse oversampling and under sampling techniques and thereafter used two ensemble methods, bagging and stacking, as well as K-Nearest Neighbour (KNN), random forest, Logistic Model Tress (LMT) and Gradient Boosted Decision Trees (GBDT) model. Moreover, three datasets – Taiwan clients credit dataset with 30,000 observations and 6,636 defaults, South-German clients credit dataset with 1,000 observations and 300 defaults, and lastly, Belgium clients 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), SMOTE, k-means SMOTE, borderline SMOTE, SMOTE Tomek and random oversampling method were tested. [9] noted that the oversampling techniques performed better than the under-sampling techniques and the GBDT

 doi:10.20944/preprints202409.0923.v

method with SMOTE performed better than the other models using accuracy, precision, recall, F-measure, ROC curve and G-means. Although, this current study uses similar methodology as [9], this current study is different, as it identifies the best method to make use of at each stage and further ensembles the boosting classifiers with other machine learning models, as well as with MLP model (three-hidden layers).

In [11], SMOTE was applied to balance the data used to build a smart application for loan approval prediction, the data used was from Kaggle repository, and contained 806 observations and 12 features, which was used to train logistic regression, decision tree, random forest, SVM, 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, taking two approaches, firstly combining the predictions from all the models, and also combining three of the best performing models. [11] observed that the deep learning models were less effective when dealing with loan dataset compared to the traditional machine learning models, with the second approach outperforming the other models. Although [11] handled class imbalance, this current study test other sampling techniques, used more data for the prediction, additionally other techniques were explored to improve the models' performance similar to [12] and [22], as feature selection techniques were used to optimise the models for credit default risk predictions. [12] 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, [22] 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.

Finally, few studies performed hyperparameter tuning using GridSearchCV. For example, reference [10] used GridSearchCV to get the parameters to build ANN, logistic regression, random forest, 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 get 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 [10] identifying XGBoost as the best performing model. This study highlighted the effectiveness of GridSearchCV in model optimisation.

In conclusion, accurately detecting credit defaults remains a concern to financial institutions, especially the role it plays in reducing financial losses [23], and while previous studies have applied various machine learning algorithms to accurately predict credit defaults, the problem of class imbalance and generalisation remains. Furthermore, the combination 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 was from LendingClub [24], 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 [25], therefore, the target is defined as:

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

 doi:10.20944/preprints202409.0923.v

To balance the system's efficiency and have a representative of the data, 30% of the data was sampled using stratified sampling method [25], which resulted to a sample size of 403,593 and 152 variables. This approach ensured that there was sufficient data without overwhelming the system.

The framework of our approach 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.

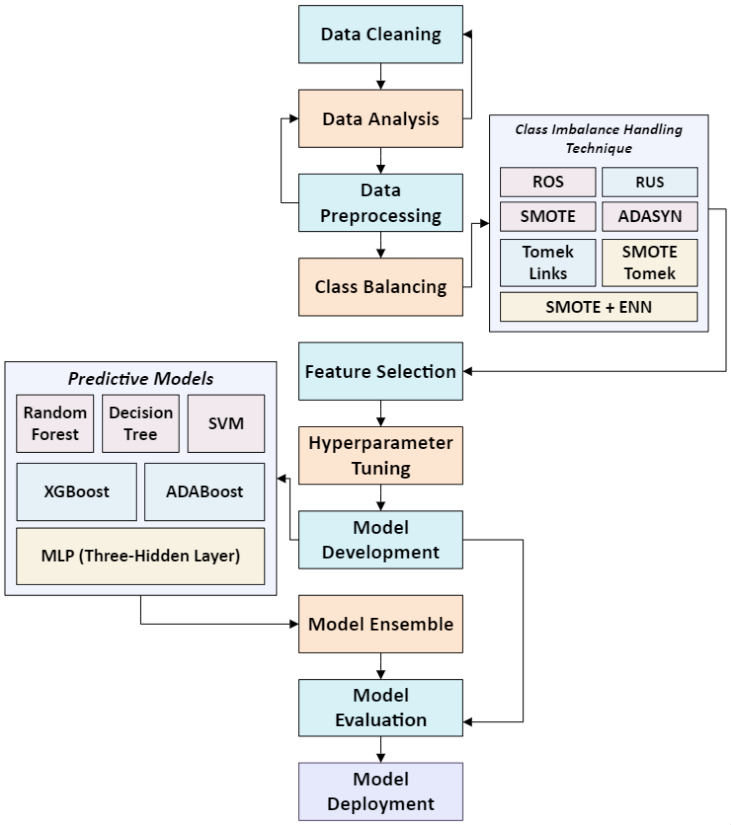

Figure 1. Research Framework.

3.1. Data Cleaning

The next stage involved data cleaning and preprocessing, which prepared the data for proper analysis, ensuring the quality and reliability of the data. The approach used was similar to [17], 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 large number of missing values that were deemed as not useful for the analysis were removed, 'emp_title' with 142,402 unique values, 'title' with 21,976 unique values that were similar to '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:

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

where,

ith central point (m_i) is defined as:

 doi:10.20944/preprints202409.0923.v

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

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 [27,28], to preserve data integrity.

Pearson correlation coefficient (r) is a filter method that measures the relationship between variables [29], and 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 [30]. It can be calculated as:

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

3.2. Data Analysis

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 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 $9,522,972, which is a possible error, which would likely affect the debt-to-income ratio 'dti', which showed a maximum DTI of 999.00%. The errors and outliers were handled in the data pre-processing stage.

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

3.3. Data Pre-Processing

This stage is very important in getting the data ready for model development, here, observed errors are removed, outliers are treated, features are binned and combined to capture more information, categorical data are one-hot encoded and transformed to numerical data, finally, the values are normalised [31,32]. At each stage different techniques were tested, using XGBoost, to identify the best technique to utilise, similar to [10]. This model was selected because it is simple yet powerful and is known for its ability to generalise well to other models, moreover, they are efficient, which helped to save time [33], this approach was done to improve the performance of the models. Furthermore, after the observed errors in features like 'annual_inc' and 'dti' were removed, and categorical features shown in Table 2 were one-hot encoded, the outliers observed in the numerical features were handled, and the data was normalised.

 doi:10.20944/preprints202409.0923.v

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 to maybe 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. The below methods were tested:

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

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

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 different range can affect some models, this was handled by scaling the features using the following normalisation techniques:

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

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

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:

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

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

$$n=\frac{n_i-n_{median}}{IQR} \tag{8}$$

3.3.3. Evaluation Metrics

 doi:10.20944/preprints202409.0923.v

To assess the effectiveness of the models including the model used in testing (XGBoost), various metrics were used. 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:

$$\text{Accuracy}= \frac{TP+TN}{TP+TN+FT+FN} \tag{9}$$

Precision which measures the proportion of the actual defaults among all default predictions:

$$\text{Precision}= \frac{TP}{TP+FP} \tag{10}$$

Recall which is also known as sensitivity or True Positive Rate (TPR), measures the proportion of the actual defaults that are correctly identified, calculated as:

$$\text{Recall(Sensitivity)}= \frac{TP}{TP+FN} \tag{10}$$

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. ROC curve plots the TPR against the False Positive Rate (FRP).

3.3.4. Identifying Data Pre-processing Techniques

Recall, precision, and accuracy are metrics used in the selection process. The result can be seen in Table A1. The 'winsorize' method was identified as the technique to use in handling the outliers, and the 'robust scaler' was used for the data normalisation.

3.4. Addressing Class Imbalance

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 like SMOTE and ADASYN are used frequently, this research requires that the best technique to make use of is identified, therefore, different techniques were tested as shown below:

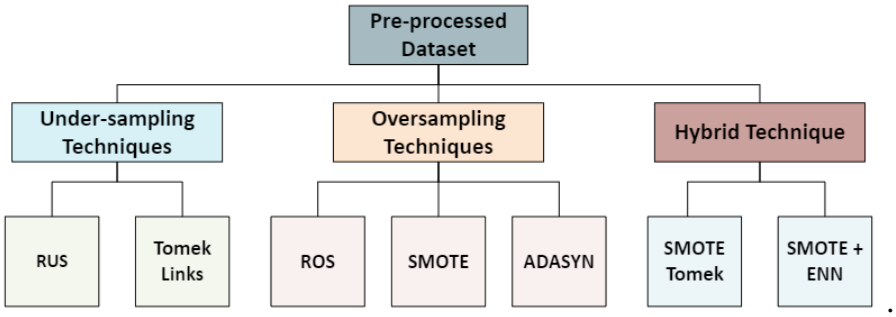

Figure 2. Class Imbalance Handling Techniques.

The techniques evaluated are:

Random Over-Sampling (ROS): It works by randomly adding data samples from the minority class to the dataset until the whole data is balanced [33,34]. It can be represented as

$$\text{New } S_{minority} = S_{minority} \cup \{S_{minority} \text{ duplicated until } |S_{minority}| = N_{majority}\} \tag{11}$$

where,

$S_{minority}$ = minority class samples.

 doi:10.20944/preprints202409.0923.v

$N_{majority}$ = Number of majority class samples.

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

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 data is balance [34]. It can be shown as:

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

SMOTE: It works by generating synthetic data through interpolating between the existing minority class data samples and their nearest neighbour, thereby adding new data point without adding duplicates [9,36]. It generates synthetic data (x_{new}) with:

$$x_{new} = x_i + \lambda * (x_{nn} - x_i) \quad (13)$$

where,

x_i = minority class.

x_{nn} = one of the nearest neighbours.

λ = a random value between [0, 1].

ADASYN: It works in a similar way to SMOTE, but it focuses on generating more synthetic samples for the harder to classify minority class, the number of the synthetic sampled i, (G_i) is calculated as:

$$G_i = d_i * G \quad (14)$$

where,

d_i = ratio of the majority neighbours.

G = total synthetic samples needed.

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 [37,36]. 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.

SMOTE-Tomek: It is a combination of SMOTE and Tomek Links, firstly, SMOTE is used to generate the synthetic data samples for the minority class, then Tomek Links are removed to clean up the boundaries between classes, thereby improving the quality of the synthetic data [38].

SMOTE+ENN: It is a hybrid technique that improves the quality of the synthetic data created by SMOTE, as Edited Nearest Neighbour (ENN) is used to remove instances of misclassification of the nearest neighbour [35-40].

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 AUC. After the data has been balanced, the next step involved testing various splits to determine the best split to use, with 20%, 25%, 30%, 35% and 40% test ratios evaluated. Subsequently, the selected split 80:20 was used in selecting the appropriate features for the model development in the next stage.

3.5. Feature Selection

Feature selection is a crucial step in model development, and the goal here is to get features that can be used to simple and efficient models, as deploying a model with 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 [9,22]. 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 [41]. This method aims to find the features that gives the best performance using a scoring metric (scorer) and given that the focus is to correctly predict defaults, recall was used. Additionally, the 'step' parameter was set to 1, which indicates that

 doi:10.20944/preprints202409.0923.v

one feature is removed per iteration, moreover, redundant features were also removed, thereby ensuring that the best features are selected for the model development process.

3.6. Model Development Process

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

3.6.1. Predictive Models

Decision Tree, which has a tree structure that works by recursively splitting the data into subsets of the tree based on a decision rule [1]. It selects the best feature to split based on criteria like Gini index – the impurity of a node and the values closer to 0 are the purer nodes, it is calculated and is calculated as:

$$\text{Gini}= 1-\sum_{i=1}^{c}(p_i)^2 \tag{15}$$

where,

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

c = number of classes.

Random Forest, an ensemble learning method that combines the predictions gotten from training multiple decision trees to get the final predictions [37,42]. The final predictions are made using majority voting. Since it is a combination of decision trees, it uses the Gini as well for splitting.

SVM, finds the optimal hyperplane that separates the data into different classes [23]. It uses kernel functions to handle non-linear separation by mapping input features into high-dimensional spaces. The hyperplane:

$$h(x_i)=\text{sign}(w\cdot x_i+b) \tag{16}$$

XGBoost, builds an ensemble of weak learners in an iterative manner in order to improve on the models' performance [17]. It uses gradient boosting with specific loss functions l and regularisation terms $\Omega(f)$:

$$L^{(t)}=\sum_{i=1}^{n} l\left(y_i,\hat{y}_i^{(t-1)}+f_t(X_i)\right)+\Omega(f_t) \tag{17}$$

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.

ADABoost, focuses on creating strong classifiers by combining multiple weak classifiers [43]. It trains weaker learners on the errors made by the previous ones, and when there is a misclassification, it assigns more weight to them. Final predictions are calculated by:

$$H(x)=\text{sign}\left[\sum_{t=1}^{T}\alpha_t h_t(x)\right] \tag{18}$$

where,

T = weak classifiers.

α_t= weight for the weak classifier.

 doi:10.20944/preprints202409.0923.v

$h_t(x)$ = predictions for the weak classifier.
$sign$ = determines the final prediction

MLP, is a feedforward type of ANN that consist 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. Each connection has a weight, and MLP uses backpropagation to adjust the weights based on the error in the output, and gradually increases the predictions. Furthermore, this model is capable of learning complex patterns in data.

3.6.2. Hyperparameter Tuning

This process can help in getting the best performance from each model, and similar to [41], GridSearchCv was used to get the parameters used for the model development. Some of the parameters used for GridSearchCV are shown in Table A2, the values were predominantly chosen to strike a balance between reducing overfitting and increasing the performance of the models, for instance, the max_depth limits the depth of the tree, therefore, the values chosen may capture complexity and make the model efficient. Additionally, for reg_alpha and reg_lambda, which are Lasso (L1) and Ridge (L2) regularisation terms, they allow XGBoost control sparsity as well as the magnitude of the model's weights. The best parameters identified and used for the model development are shown in Table 3:

Table 3. Hyperparameter Tuning.

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. Additionally, the second method uses the stacking method for the combination, here, a meta-model is used to get the final prediction, the meta-model learns how best to aggregate the predictions to make the final prediction. The ensemble methods used in this work and how they are combined is shown in Figures 3 and 4 respectively.

 doi:10.20944/preprints202409.0923.v

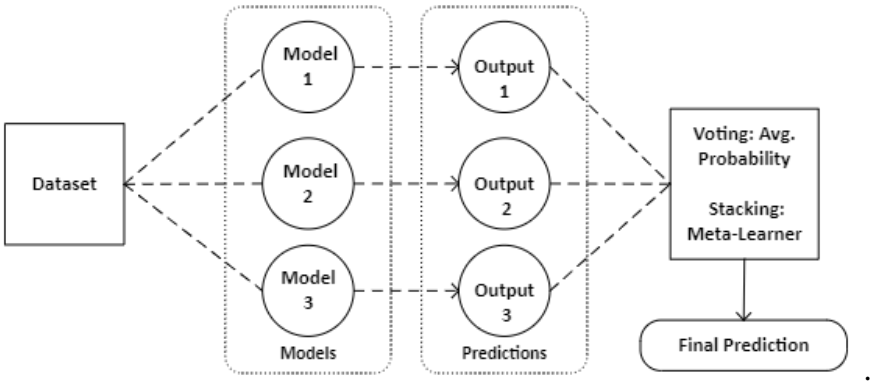

Figure 3. Ensemble Methods.

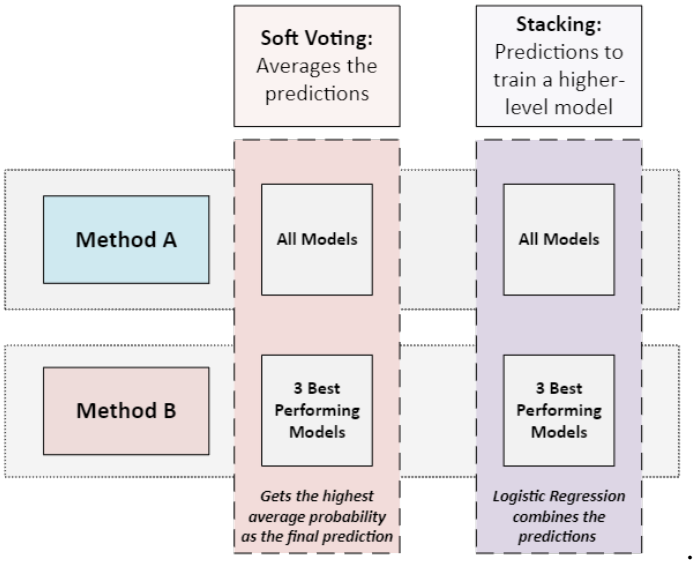

Figure 4. Methods used in combining the model predictions.

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 4 shows the results of employing the sampling techniques, while Figure 5 shows the results from using RFECV.

Table 4. 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**

 doi:10.20944/preprints202409.0923.v

As presented in Table 4, ROS performed better than RUS, which coincides with the observation made by [9], that the oversampling technique always performed better than the under-sampling technique, and while ADASYN, SMOTE and SMOTE-Tomek showed impressive results, SMOTE+ENN showed the most impressive performance across all the metrics, hence, by combining both SMOTE and ENN, the data is not only being balanced, the noise or ambiguous data samples that may affect the model's performance are also being removed [40]. Additionally, with a recall of 92.02%, it showed that the model captures the minority class correctly, which is sometimes more important than getting a high accuracy. Furthermore, given the result, SMOTE+ENN was used to balance the dataset.

Additionally, Figure 5 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 shows the variability and stability of the model's performance across the folds, with 48 features identified as the optimal number of features to get the optimal recall score of 92.16%.

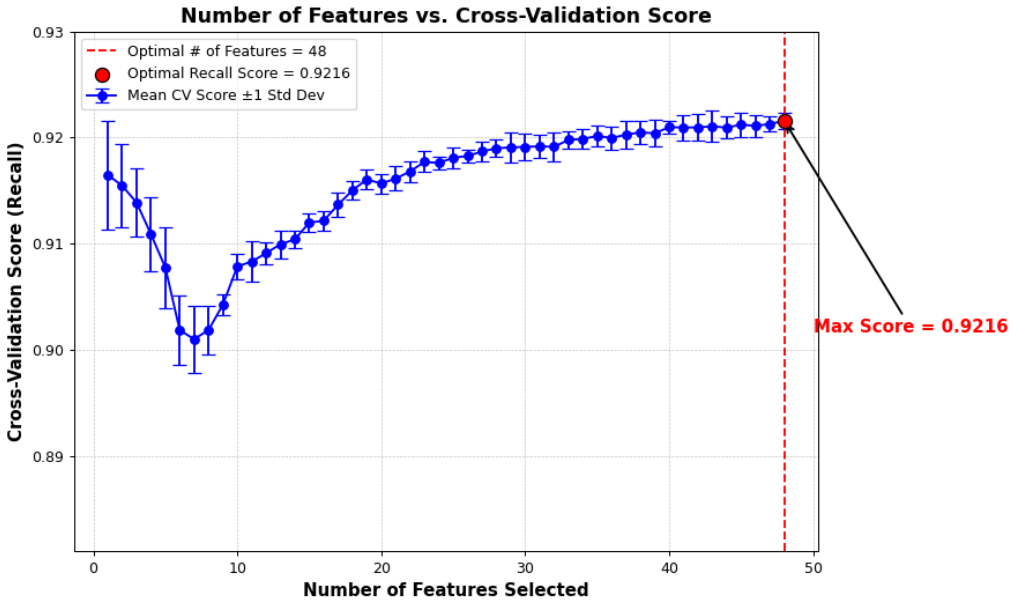

Figure 4. Features Selected.

Additionally, the results for the feature importance are shown in Figure 5 below, with the interest rate, credit score and the loan term, identified as the most important features in the prediction of credit default. Based on these findings and the best parameters listed in Table 3 (Section 3.6.3), the models were subsequently developed.

 doi:10.20944/preprints202409.0923.v

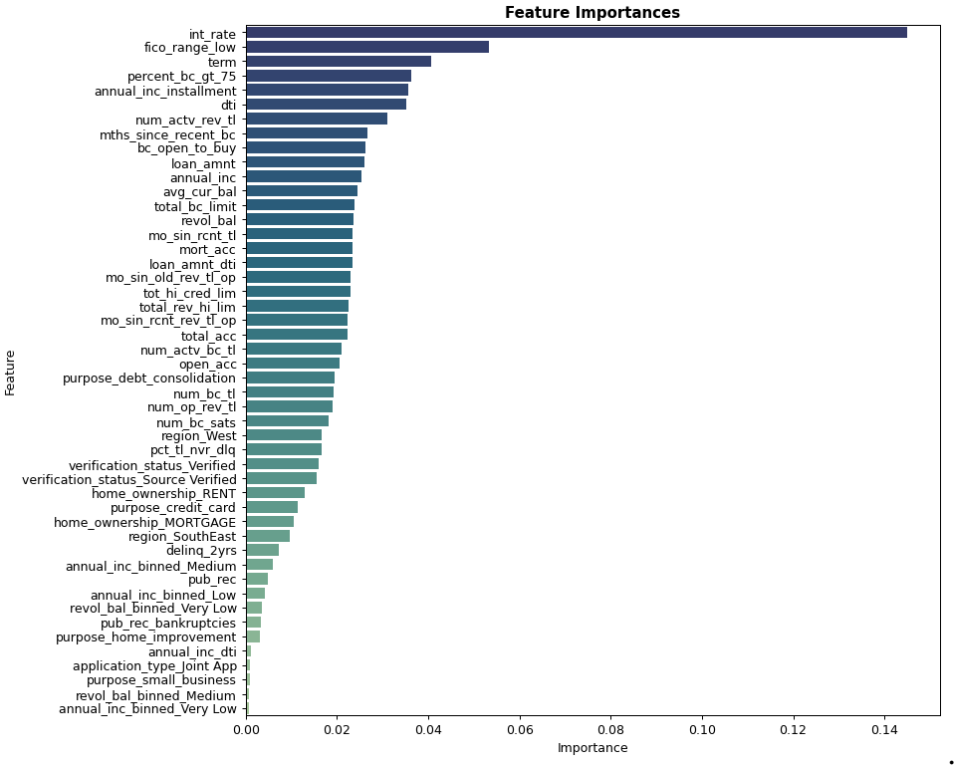

Figure 5. Feature Importance.

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 5:

Table 5. 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 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 recall, which indicates that the models can effectively identify default cases are 96.56% (Random Forest) and 93.30% (XGBoost), with high AUC scores of 95.89% and 97.29%, which suggests that the models are able to effectively distinguish between the defaulters and the non-defaulters. Similarly, with a recall of 92.48%, MLP had a solid performance, however, SVM had the lowest score, despite having a high precision value of 94.76%, which may suggest that SVM is not able to address the complexity of the credit default data.

 doi:10.20944/preprints202409.0923.v

The ROC curve in Figure 6 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.

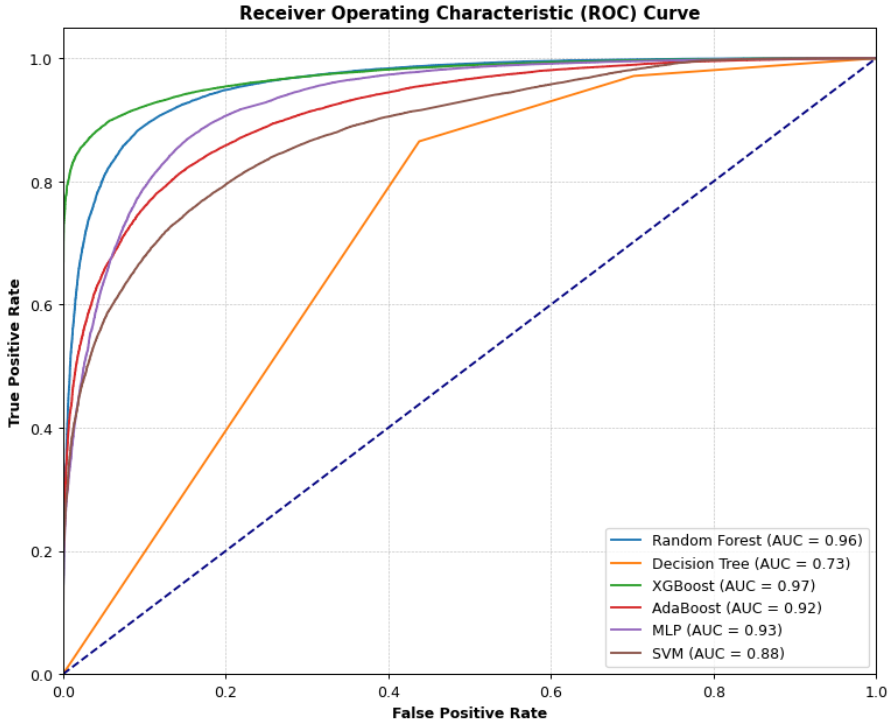

Figure 6. ROC Curve (Individual Models).

For the individual models XGBoost with an accuracy of 91.56%, precision of 94.78% and AUC of 97.26% achieved the best results, which shows how effective the model is at handling complex credit default data, with the performance being 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:

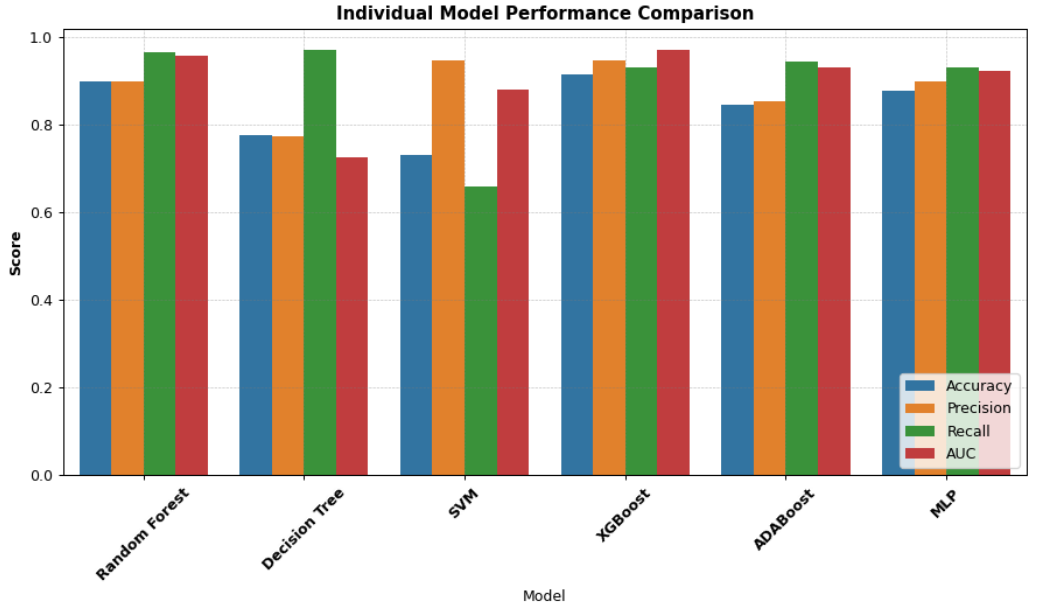

Figure 7. Comparative Result (Individual Model).

 doi:10.20944/preprints202409.0923.v

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 below:

Table 6. 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. The method A, which combines the predictions from all the models achieved the highest result overall (Staking A), with an accuracy of 93.69%, precision of 95.59 and AUC of 97.81%, which suggests that combining the models with the stacking technique led to an improvement in the performance.

Although XGBoost and the ensemble methods – Voting A, B and Stacking A, B performed well, as seen in Figure 8, however, Stacking A's performance is impressive, as it is identified as the best model with the ability to effectively separate the classes. This 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.

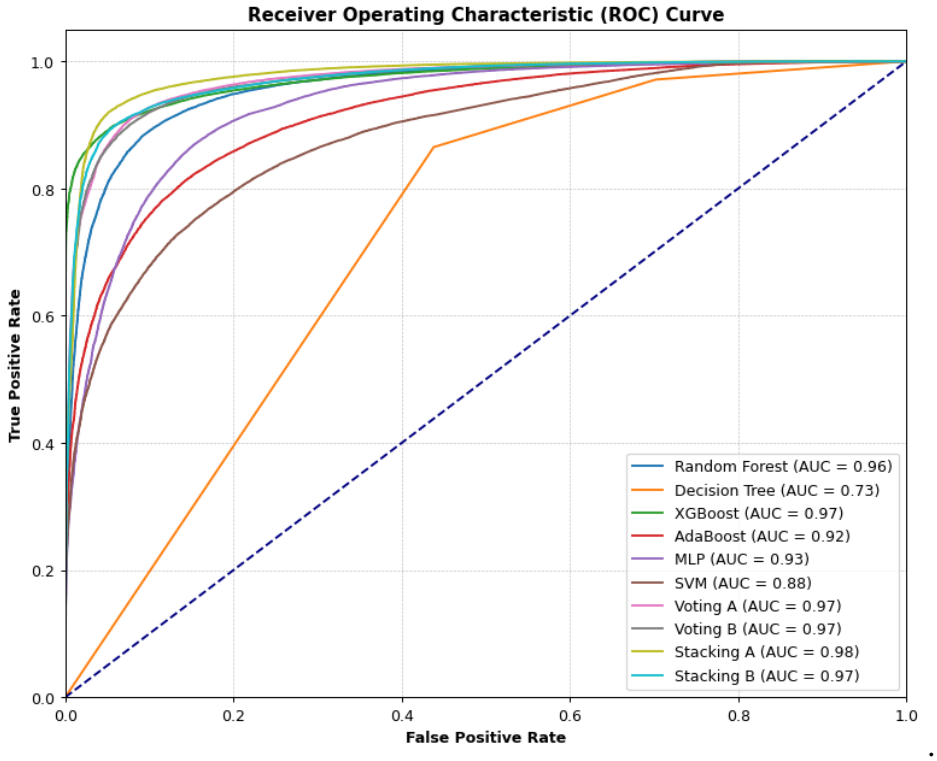

Figure 8. ROC Curve (All Models).

The comparative result can be seen 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.

 doi:10.20944/preprints202409.0923.v

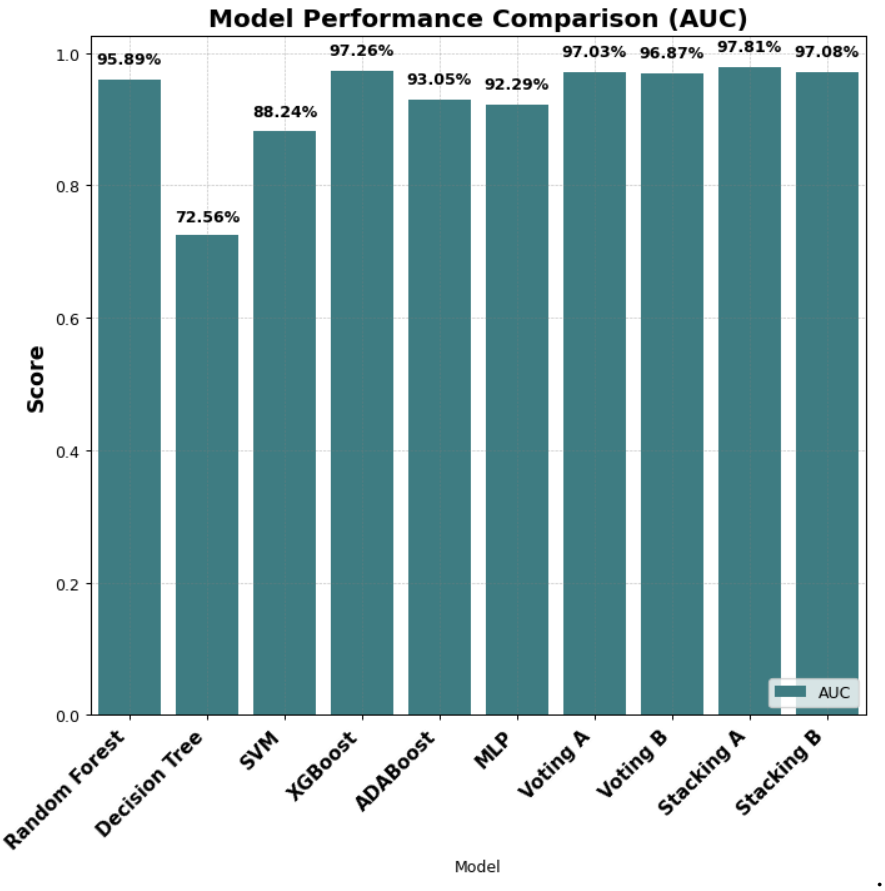

Figure 9. Comparative Result (All Models).

4.2. Comparison with Baseline Results

The performance of the proposed model is compared with other related works, which served as the baseline for further evaluation. The result can be seen below:

Table 5. Baseline Comparison.

Reference	Dataset	Total Record	Model	Result
Proposed Method	**LendingClub**	**394,073**	**Stacking**	**93.7%**
[9]	Taiwan Credit-Client	30,000	GBDT*	88.7%
	South German Credit-Client	1,000		83.5%
	Belgium Credit-Client	285,299		86.3%
[10]	LendingClub	282,763	XGBoost	87.9%
[11]	Data from Kaggle (Chatterjee, 2021)	614	Voting	87.3%
[12]	Auto-Finance	25,383	Deep Learning-Optimised Stacking	92.7%

*Gradient Boosted Decision Tree (GBDT)

The proposed method displays a higher accuracy when compared with the other results, [9] achieved accuracy scores ranging from 83.5% to 88.7% using GBDT, with results varying across the different datasets, additionally, [10] achieved an accuracy of 87.9% with the XGBoost model on LendingClub dataset, moreover, [12] achieved an accuracy of 92.7% with Deep Learning-Optimised

 doi:10.20944/preprints202409.0923.v

Stacking, an ensemble model. However, the proposed result performed better with a score of 93.7%, showing how well the model is at predicting credit default.

Finally, SHAP (SHapley Additive exPlanations) with XGBoost model was 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 pushes the prediction towards default, while positive values, contributes to the predictions being higher, for instance, positive 'fico_range_low' means that higher credit scores move the prediction away from being classed as a default. With the proposed model developed, tested and evaluated, this research shows how well the methodology used worked, as testing the techniques to identify the most suitable one contributed to the overall performance of the models, additionally, combining the predictions from wearker classifiers contributed as well.

5. Conclusion and Recommendations

In this study, various techniques have been explored to identify the best techniques to use, as well as to handle the issue of class imbalance. Additionally, diverse machine learning and deep learning models have been explored to predict the likelihood of loan default – Random Forest, Decision Tree, Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (ADABoost) and Multi-Layered Perceptron (MLP) with three-hidden layer.

Voting and stacking ensemble techniques have been employed to optimise the models, with the stacking method combining the predictions from all the models identified as the best-performing model. The proposed model is capable of precisely gauging default risk, with a recall of 95.5%. Moreover, when comparing the results from this research with those from the baseline results as shown in Section 4.2, this study was able to achieve higher accuracy, 93.7%. Furthermore, this research can help with future work, as the performance of diverse techniques and models has been explored and documented.

The techniques and findings obtained from this research 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. 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

 doi:10.20944/preprints202409.0923.v

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

*Selected Techniques

Table A2. Hyperparameter Tuning-Some Parameters Used.

Model	params
Random Forest	{'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100} {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200} {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200} {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 500} {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, '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} {'class_weight': None, 'criterion': 'gini', 'max_depth': 15, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2} {'class_weight': None, 'criterion': 'gini', 'max_depth': 15, 'max_leaf_nodes': 20, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2} {'class_weight': None, 'criterion': 'gini', 'max_depth': 15, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 2, 'min_samples_split': 2} {'class_weight': None, 'criterion': 'gini', 'max_depth': 15, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 5}
XGBOOST	{'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 200, 'reg_alpha': 1, 'reg_lambda': 1.5, 'subsample': 1.0} {'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 300, 'reg_alpha': 1, 'reg_lambda': 1.5, 'subsample': 1.0} {'colsample_bytree': 0.9, 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 200, 'reg_alpha': 1, 'reg_lambda': 1.5, 'subsample': 1.0} {'colsample_bytree': 0.9, 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 300, 'reg_alpha': 1, 'reg_lambda': 1.5, 'subsample': 1.0} {'colsample_bytree': 0.9, 'learning_rate': 0.2, 'max_depth': 20, 'n_estimators': 200, 'reg_alpha': 1, 'reg_lambda': 1.5, 'subsample': 1.0}
ADABoost	{'learning_rate': 0.15, 'n_estimators': 100} {'learning_rate': 0.15, 'n_estimators': 200}

 doi:10.20944/preprints202409.0923.v

MLP	{'learning_rate': 0.2, 'n_estimators': 200} {'learning_rate': 0.15, 'n_estimators': 300} {'learning_rate': 0.2, 'n_estimators': 300} {'activation': 'tanh', 'alpha': 0.001, 'batch_size': 'auto', 'early_stopping': True, 'hidden_layer_sizes': (100, 100, 100), 'learning_rate': 'constant', 'solver': 'adam'} {'activation': 'relu', 'alpha': 0.002, 'batch_size': 'auto', 'early_stopping': True, 'hidden_layer_sizes': (100, 100, 100), 'learning_rate': 'constant', 'solver': 'adam'} {'activation': 'tanh', 'alpha': 0.002, 'batch_size': 'auto', 'early_stopping': True, 'hidden_layer_sizes': (100, 100, 100), 'learning_rate': 'constant', 'solver': 'adam'} {'activation': 'relu', 'alpha': 0.001, 'batch_size': 'auto', 'early_stopping': True, 'hidden_layer_sizes': (150, 150, 150), 'learning_rate': 'constant', 'solver': 'adam'} {'activation': 'relu', 'alpha': 0.001, 'batch_size': 'auto', 'early_stopping': True, 'hidden_layer_sizes': (150, 150, 150), 'learning_rate': 'adaptive', 'solver': 'sgd'}
SVM	{'C': 1, 'degree': 2, 'gamma': 1, 'kernel': 'rbf'} {'C': 1, 'degree': 2, 'gamma': 'scale', 'kernel': 'rbf'} {'C': 1, 'degree': 2, 'gamma': 'scale', 'kernel': 'linear'} {'C': 10, 'degree': 2, 'gamma': 1, 'kernel': 'rbf'} {'C': 10, 'degree': 2, 'gamma': 'scale', 'kernel': 'rbf'}

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

 doi:10.20944/preprints202409.0923.v

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

 doi:10.20944/preprints202409.0923.v

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. M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study," IOP Conference Series: Materials Science and Engineering, vol. 1022, no. 1, p. 012042, Jan. 2021, doi: https://doi.org/10.1088/1757-899x/1022/1/012042.
2. V. Ivashina and D. Scharfstein, "Bank lending during the financial crisis of 2008," Journal of Financial Economics, vol. 97, no. 3, pp. 319–338, 2010, doi: https://doi.org/10.1016/j.jfineco.2009.12.001.
3. M. K. Brunnermeier, "Deciphering the Liquidity and Credit Crunch 2007–2008," Journal of Economic Perspectives, vol. 23, no. 1, pp. 77–100, Jan. 2009, doi: https://doi.org/10.1257/jep.23.1.77.
4. V. Acharya, T. Philippon, M. Richardson, and N. Roubini, "The financial crisis of 2007-2009: Causes and remedies," Hoboken, NJ: John Wiley & Sons, 2009.
5. L. N. Switzer and J. Wang, "Default Risk Estimation, Bank Credit Risk, and Corporate Governance," Financial Markets, Institutions & Instruments, vol. 22, no. 2, pp. 91–112, Apr. 2013, doi: https://doi.org/10.1111/fmii.12005.
6. N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," Artificial Intelligence Review, vol. 45, no. 1, pp. 1–23, Oct. 2015, doi: https://doi.org/10.1007/s10462-015-9434-x.
7. D. Duffie, Measuring corporate default risk. Oxford University Press, 2011.
8. LendingClub, "Peer to Peer Lending & Alternative Investing," Lendingclub.com, 2019. https://www.lendingclub.com/
9. T. M. Alam et al., "An investigation of credit card default prediction in the imbalanced datasets," Ieee Access, vol. 8, pp. 201173–201198, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3033784.
10. A.-H. Chang, L.-K. Yang, R.-H. Tsaih, and S.-K. Lin, "Machine learning and artificial neural networks to construct P2P lending credit-scoring model: A case using Lending Club data," Quantitative Finance and Economics, vol. 6, no. 2, pp. 303–325, 2022, doi: https://doi.org/10.3934/qfe.2022013.
11. N. Uddin, Md. K. Uddin Ahamed, M. A. Uddin, Md. M. Islam, Md. A. Talukder, and S. Aryal, "An ensemble machine learning based bank loan approval predictions system with a smart application," International Journal of Cognitive Computing in Engineering, vol. 4, pp. 327–339, Jun. 2023, doi: https://doi.org/10.1016/j.ijcce.2023.09.001.
12. Y. Wang, M. Wang, P. Yong, and J. Chen, "Joint loan risk prediction based on deep learning-optimized stacking model," Engineering reports, vol. 6, no. 4, Aug. 2023, doi: https://doi.org/10.1002/eng2.12748.
13. E. B. Ntiamoah, E. Oteng, B. Opoku, and A. Siaw, "Loan default rate and its impact on profitability in financial institutions," Research Journal of Finance and Accounting, vol. 5, no. 14, Art. no. 14, 2014.
14. K. Amzile and M. Habachi, "Assessment of Support Vector Machine performance for default prediction and credit rating," Banks and Bank Systems, vol. 17, no. 1, pp. 161–175, Apr. 2022, doi: https://doi.org/10.21511/bbs.17(1).2022.14.
15. J. Xu, Z. Lu, and Y. Xie, "Loan default prediction of Chinese P2P market: a machine learning methodology," Scientific Reports, vol. 11, no. 1, Sep. 2021, doi: https://doi.org/10.1038/s41598-021-98361-6.
16. A. Dzik-Walczak and M. Heba, "An implementation of ensemble methods, logistic regression, and neural network for default prediction in Peer-to-Peer lending," Zbornik radova Ekonomskog fakulteta u Rijeci, vol. 39, no. 1, pp. 163–197, Jun. 2021, doi: https://doi.org/10.18045/zbefri.2021.1.163.

17. X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "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," Electronic Commerce Research and Applications, vol. 31, pp. 24–39, Sep. 2018, doi: https://doi.org/10.1016/j.elerap.2018.08.002.
18. K. Nordhausen, "Ensemble Methods: Foundations and Algorithms by Zhi-Hua Zhou," International Statistical Review, vol. 81, no. 3, pp. 470–470, Nov. 2013, doi: https://doi.org/10.1111/insr.12042_10.
19. Y.-C. Chang, K.-H. Chang, and G.-J. Wu, "Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions," Applied Soft Computing, vol. 73, pp. 914–920, Dec. 2018, doi: https://doi.org/10.1016/j.asoc.2018.09.029.
20. M. Jumaa, M. Saqib, and A. Attar, "Improving Credit Risk Assessment through Deep Learning-based Consumer Loan Default Prediction Model," International Journal of Finance & Banking Studies, vol. 12, no. 1, pp. 85–92, Jun. 2023, doi: https://doi.org/10.20525/ijfbs.v12i1.2579.
21. J. Duan, "Financial system modeling using deep neural networks (DNNs) for effective risk assessment and prediction," Journal of the Franklin Institute, vol. 356, no. 8, pp. 4716–4731, May 2019, doi: https://doi.org/10.1016/j.jfranklin.2019.01.046.
22. X. Li, D. Ergu, D. Zhang, D. Qiu, Y. Cai, and B. Ma, "Prediction of loan default based on multi-model fusion," Procedia Computer Science, vol. 199, pp. 757–764, 2022, doi: https://doi.org/10.1016/j.procs.2022.01.094.
23. F. E. Moula, C. Guotai, and M. Z. Abedin, "Credit default prediction modeling: an application of support vector machine," Risk Management, vol. 19, no. 2, pp. 158–187, Feb. 2017, doi: https://doi.org/10.1057/s41283-017-0016-x.
24. LendingClub, "Developers | LendingClub's API," Lendingclub.com, 2024. https://www.lendingclub.com/developers/api-overview (accessed Aug. 29, 2024).
25. LendingClub, "What Do the Different Note Statuses Mean?," www.lendingclub.com, 2023. https://www.lendingclub.com/help/investing-faq/what-do-the-different-note-statuses-mean
26. A. S. Acharya, A. Prakash, P. Saxena, and A. Nigam, "Sampling: Why and How of it?," Indian Journal of Medical Specialities, vol. 4, no. 2, pp. 330–333, 2013, doi: https://doi.org/10.7713/ijms.2013.0032.
27. M. K. Cain, Z. Zhang, and K.-H. Yuan, "Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation," Behavior Research Methods, vol. 49, no. 5, pp. 1716–1735, Oct. 2016, doi: https://doi.org/10.3758/s13428-016-0814-1.
28. S. Alam, M. S. Ayub, S. Arora, and M. A. Khan, "An investigation of the imputation techniques for missing values in ordinal data enhancing clustering and classification analysis validity," Decision Analytics Journal, vol. 9, p. 100341, Dec. 2023, doi: https://doi.org/10.1016/j.dajour.2023.100341.
29. P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: Appropriate Use and Interpretation," Anesthesia & Analgesia, vol. 126, no. 5, pp. 1763–1768, 2019, doi: https://doi.org/10.1213/ane.0000000000002864.
30. M. Tsagris and N. Pandis, "Multicollinearity," American Journal of Orthodontics and Dentofacial Orthopedics, vol. 159, no. 5, pp. 695–696, May 2021, doi: https://doi.org/10.1016/j.ajodo.2021.02.005.
31. S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," Big Data Analytics, vol. 1, no. 1, Nov. 2016, doi: https://doi.org/10.1186/s41044-016-0014-0.
32. S.-A. N. Alexandropoulos, S. B. Kotsiantis, and M. N. Vrahatis, "Data preprocessing in predictive data mining," The Knowledge Engineering Review, vol. 34, no. 1, 2019, doi: https://doi.org/10.1017/s026988891800036x.
33. A. A. Khan, O. Chaudhari, and R. Chandra, "A Review of Ensemble Learning and Data Augmentation Models for Class Imbalanced problems: Combination, Implementation and Evaluation," Expert Systems with Applications, vol. 244, p. 122778, Dec. 2023, doi: https://doi.org/10.1016/j.eswa.2023.122778.
34. F. M. Megahed, Y.-J. Chen, A. Megahed, Y. Ong, N. Altman, and M. Krzywinski, "The class imbalance problem," Nature Methods, vol. 18, no. 11, pp. 1270–1272, Oct. 2021, doi: https://doi.org/10.1038/s41592-021-01302-4.
35. A. Namvar, M. Siami, F. Rabhi, and M. Naderpour, "Credit risk prediction in an imbalanced social lending environment," International Journal of Computational Intelligence Systems, vol. 11, no. 1, pp. 925–935, Mar. 2018, doi: https://doi.org/10.2991/ijcis.11.1.70.
36. E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset," Sensors, vol. 22, no. 9, p. 3246, Jan. 2022, doi: https://doi.org/10.3390/s22093246.
37. Y.-R. Chen, J.-S. Leu, S.-A. Huang, J.-T. Wang, and J.-I. Takada, "Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets," IEEE Access, vol. 9, pp. 73103–73109, 2021, doi: https://doi.org/10.1109/access.2021.3079701.
38. M. Enes, S. Aras, and İ. Deveci Kocakoç, "Investigating the Effect of Class Balancing Methods on the Performance of Machine Learning Techniques: Credit Risk Application," İzmir Yönetim Dergisi, vol. 5, no. 1, pp. 55–70, Jul. 2024, doi: https://doi.org/10.56203/iyd.1436742.

39. Y. Zhu, Y. Hu, Q. Liu, H. Liu, C. Ma, and J. Yin, "A Hybrid Approach for Predicting Corporate Financial Risk: Integrating SMOTE-ENN and NGBoost," IEEE Access, vol. 11, pp. 111106–111125, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3323198.
40. M. Muntasir Nishat et al., "A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers with SMOTE-ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset," Scientific Programming, vol. 2022, pp. 1–17, Mar. 2022, doi: https://doi.org/10.1155/2022/3649406.
41. N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," Journal of Information Security and Applications, vol. 55, p. 102596, Dec. 2020, doi: https://doi.org/10.1016/j.jisa.2020.102596.
42. M. Aria, C. Cuccurullo, and A. Gnasso, "A comparison among interpretative proposals for Random Forests," Machine Learning with Applications, vol. 6, p. 100094, Dec. 2021, doi: https://doi.org/10.1016/j.mlwa.2021.100094.
43. P. Bühlmann and T. Hothorn, "Boosting Algorithms: Regularization, Prediction and Model Fitting," Statistical Science, vol. 22, no. 4, pp. 477–505, Nov. 2007, doi: https://doi.org/10.1214/07-sts242.