Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 13, No. 3, September 2025, pp. 703∼721
ISSN: 2089-3272, DOI: 10.52549/ijeei.v13i3.6667

Exploratory Analysis of the Impact of Data Balancing on the Classifier's Performance in Predicting Creditworthiness Reliability

Md. Mahedi Hassan[1], Arif Hossen[2], Yeasin Arafat[3], Md Nurunnabi sarker[4], Md Hossain Jamil[5], Ayesha Siddika[6]
[1]Computer Science and Engineering, World University of Bangladesh, Dhaka-1230, Bangladesh
[2]Business Analytics, International American University, Los Angeles, USA
[3]IT Management, Westcliff University, California, USA
[4]Data Analysis, Westcliff University, California, USA
[5]Information Technology, Humphreys University, California, USA
[6]Software Engineering, Daffodil International University, Dhaka-1216, Bangladesh

Article Info

Article history:

Received May 13, 2025
Revised Aug 30, 2025
Accepted Sep 13, 2025

Keywords:

Creditworthiness Prediction
Loan Eligibility
Machine learning
Algorithm comparison
Imbalance data handling
XAI

ABSTRACT

This study examines the application of machine learning algorithms for creditworthiness prediction within the banking sector and addresses the issue of class imbalance through sampling methodologies. The research indicates that using the Stacking Ensemble algorithm with random oversampling can predict creditworthiness with an impressive 93% accuracy. The method consistently achieves excellent precision, recall, and F1-score values, indicating that it can produce accurate predictions while maintaining a balanced evaluation. Random oversampling helps models improve their predictive accuracy and reduce class imbalance. The research findings underscore the feasibility of this technique for financial institutions, facilitating informed lending decisions and improving credit risk assessment methodologies. This research enhances the field by identifying the most effective machine learning methods for accurate creditworthiness evaluation. Using XAI tools like Shapash provides financial organizations with valuable insights into assessing loan risks and enhancing their lending operations.

Corresponding Author:

Md. Mahedi Hassan
Department of Computer Science and Engineering
World University of Bangladesh
Dhaka-1230, Bangladesh
Email: mahedi7171@gmail.com

1. INTRODUCTION

Loan eligibility is crucial in business and banking. Personal and commercial loan applicants must be carefully assessed by lenders. This examination formerly used subjective, time-consuming manual techniques. The emergence of machine learning algorithms and data analysis has driven interest in automated creditworthiness assessment. This research compares creditworthiness-predicting machine learning techniques. Loan defaults affect both established banks and the startup Internet finance industry. Borrowers who default hurt banks and the economy, perhaps causing an economic crisis. [1] To determine loan eligibility, the loan eligibility prediction task evaluates income, credit history, employment, and other characteristics. Traditional rule-based techniques cannot capture complex variable relationships. Logistic regression is used to predict

loan defaulters [2]. An ensemble model using two or more classifiers is proposed to improve loan acceptance predictions [3]. In contrast, machine learning methods can use such data to make accurate predictions. Through parsing of historical loan transaction data, these algorithms can identify patterns and associations that lending approval decisions can rely on. The effectiveness of machine learning and deep learning models for stock market trend [4] forecasting has led to their use in various other financial fields with the aim of automating and improving processes. The success of these advanced models has enabled financial operations automation and innovation, which could enhance efficiency and accuracy in manual jobs [5]. This study aims to improve the accuracy of loan eligibility prediction, reduce decision time, and minimize workload. Lenders can even use machine-learning algorithms to help them make better and more calculated decisions on who gets approved for loans, thus reducing the risk factor. Prediction of loan eligibility is one way to achieve this, allowing all individuals to have access to funds on a fair and consistent basis, which are prerequisites for financial inclusion and economic development [6]. Performance variations across the datasets highlight that a model that is robust and can perform well on both large and small sample data is crucial. The goal is to develop a robust model that works in varied data settings. This entails achieving high accuracy in larger datasets and maintaining the model's usefulness with small data samples. The goal is to develop a model that overcomes data-set size problems and predicts loan eligibility on different data scales. This method strengthens a model's robustness and generalisability for use in real-world settings with diverse and changing datasets. Building an advanced ensemble machine learning model that exceeds existing performance requirements is the thesis's goal. Advanced strategies like hyperparameter tweaking boost the model's prediction power by optimising it further. A brief review with Shapash provides a comprehensive and localized explanation of the most relevant and influential variables in the model decisions, enabling a more accurate interpretation of loan eligibility predictions.

To develop a loan qualification system and to mitigate the problems existing in the study, the following objectives are set.

- To study and compare the performance of various machine learning algorithms for loan eligibility prediction, and to investigate the applicability of sampling methods on this problem.
- To construct an ensemble model capable of delivering superior performance across datasets of varying sizes by optimize the hyperparameters.
- To offer insights and recommendations on the experimental results that help financial institutions choose the most appropriate algorithm for predicting loan eligibility with global and local explanations, considering XAI tools.

Numerous studies have been conducted to determine the likelihood of someone repaying a loan, employing various algorithms and methods. In the following paragraphs, we talk about several recent studies that look into this topic.

The logistic regression model by Mohammad Ahmad Sheikh et al. [7] is essential for predictive analytics loan defaulter prediction. This technique enables for loan default risk assessment, making it easy to choose the right customers for loan offering. If the bank has a solid model to predict which client loans to accept and which to reject, it can reduce loan default risk. Based on the original data set, the maximum accuracy is 0.811. [8] Understanding the internal dependent and independent variables requires univariate, bivariate, and multivariate studies.

A significant amount of research has been conducted to predict creditworthiness using various algorithms and methods. Ashwini S. Kadam et al. proposed the Naive Bayes model, which is superior to other models in predicting loans [9].

In a thorough investigation [10], Yong Shic et al. explored the complex domain of customer churn prediction in commercial banks. At the same time, Kwofie et al. also examined the effectiveness of logistic regression in estimating the likelihood of default using data from a microfinance organization [11].

The study suggests using Random Forest and Decision Trees to predict if someone is eligible for a loan based on certain traits. The reported accuracies for these methods are 73

Singh et al. claimed in [12] that the accurate prediction of the dataset is ascertained by three machine learning algorithms: Decision Tree, XGBoost and Random Forest. Iain Brown et al. did a thorough comparison of several methods used to analyze credit score datasets in [13]. We use logistic regression, random forests, gradient boosting, neural networks, and least square SVM methods. A study [14] demonstrates that the

probability of an individual's loan approval can be forecasted using four machine learning algorithms: Random Forest, SVM, Logistic Regression, and XGBoost.

In another study[15], Odegua achieved a 79% accuracy rate by employing the XGBoost algorithm on a banking dataset. Kwofie et al. [16] utilized logistic regression on microfinance company data to forecast defaults. The researchers utilized 90 sampled beneficiaries to construct a logistic regression model and 30 beneficiaries to predict loan defaults. The analysis employed age, marital status, gender, education, business experience, and initial capital as predictive variables. Based on the model, marital status, business years, and starting capital were identified as statistically significant factors. The logistic regression model exhibited minimal explained variability in the response variable, suggesting its inadequacy in properly predicting defaults based on the chosen predictors.

Ashwini S. Kadam et al. developed the Naive Bayes model, which is superior to previous models in predicting loans [17].

Amruta S. Aphale et al. [18] automate bank risk assessment by utilizing client creditworthiness. The suggested model predicts the likelihood of a client repaying a loan by examining their behavior. The experiment showed that all algorithms, except for Nearest Centroid and Gaussian Naive Bayes, do well in terms of accuracy and other performance measures. These algorithms were right 76% to 80% of the time. They built a linear regression model that could estimate the likelihood of a person repaying a loan based on the most significant factors.

M. Srinivasa Rao et al. provided [19] methodologies for assessing credit risk using customer datasets. The proposed approach considers all factors influencing an individual's loan status and delivers precise outcomes for credit extension or denial. They developed a loan risk analysis system that utilizes five algorithms to integrate the models from the five techniques mentioned. Naive Bayes has the highest accuracy rate (75%) of all the models.

Despite numerous advances, it remains unknown how to properly handle class imbalance and apply an ensemble model effectively in credit scoring [20]. By applying feature selection and oversampling techniques (SMOTE), it is also demonstrated that the predictive score improves with the stacking ensemble model. For example, Rofik et al. combined stacking and SMOTE, which achieved 83% accuracy on a well-known credit dataset [21].

Other studies also show that ensemble methods, such as Random Forest, AdaBoost, XGBoost, and two-stage models, can help forecast creditworthiness; however, they have issues with uneven data distribution and making the models more complex to understand. For instance, Uddin et al. showed that ensemble models improve the prediction of loan acceptance, but the accuracy rates are still below 90%, which could be improved even further [22].

Additionally, some newly developed techniques for class imbalance learning, including asymmetric adjusted activation function [23] also demonstrate that the treatment of minority class representation is of paramount importance when dealing with credit scoring tasks and statistics relevantly show the overfitting problem that models tend to favor majority good customers [24].

Several previous studies have shown good results with their proposed models. However, an ensemble model that uses different classifiers is a better and more accurate option, which could outperform other individual models. Consequently, the principal aim of this research is to develop resilient models for assessing creditworthiness, utilizing eleven machine learning techniques. This contribution also features next-level explainability with XAI tools, which isn't very common in existing research.

The succeeding sections of this work are organized as follows. Section 2 describes how our proposed method operates and outlines the experimental setup. In this section, we provide a comprehensive explanation of the method we employed to address the research challenge. This document provides a comprehensive account of the methods, procedures, and tools employed in our research. Section 3 presents the findings from our experiments. In Section 4, the report concludes with a summary of our findings and an examination of their importance. We also provide an overview of relevant topics for future research and development in this sector, highlighting various approaches to further research and improvement.

2. METHOD

This section covers over the preprocessing strategies used to clean and prepare the data for analysis. It will look at solutions for managing unbalanced data, specifically the issue of unevenly distributed classes in the

target variable. Furthermore, the chapter will present information about several machine-learning techniques. It will also go into detail about the hyperparameter tuning procedure and the feature importance analysis technique.

2.1. Dataset Description

The Dream Housing Finance company collected the data in the dataset to automate the process of determining who is eligible for a loan [25]. The idea is to identify groups of customers who are eligible for a loan based on the information provided in an online application form. The dataset includes these variables:

Table 1. A short summary of the Dataset

Feature	Description
Loan_ID	A unique code assigned to each loan application
Gender	Whether the applicant identifies as male or female
Married	Indicates if the applicant is married or single
Dependents	The number of people financially dependent on the applicant
Education	The applicant's level of education, either graduate or undergraduate
Self_Employed	Shows if the applicant runs their own business or not
ApplicantIncome	The applicant's yearly income
CoapplicantIncome	The annual income of the co-applicant, if there is one
LoanAmount	The amount of money the applicant is asking to borrow (in thousands)
Loan_Amount_Term	How long the applicant has to repay the loan, in months
Credit_History	Whether the applicant's credit record meets the lender's standards (1 for yes, 0 for no)
Property_Area	The type of area where the property is located: urban, semi-urban, or rural
Loan_Status	The final decision on the loan application, approved or not approved

These variables capture various attributes and information about the loan applicants, their financial situation, and the properties associated with the loan applications. The dataset contains a total of 614 entries. However, some columns have missing values (nonnull count is less than 614). The data types of the columns include float64 (for numerical values), int64 (for integer values), and object (for categorical values).

2.2. Overview of the Methodology

Various machine learning methods have been applied in the banking sector to assess creditworthiness. A relevant dataset was obtained from Kaggle and analyzed using Python within the Anaconda Jupyter Notebook environment. The methodology included data collection, preprocessing, data splitting, algorithm training and testing, performance evaluation, and prediction formulation. Performance metrics such as precision, recall, accuracy, and F1 score were used to address data bias. The results were visualized to support more effective credit risk assessment by banks. A 10-fold cross-validation procedure was implemented to ensure model stability across multiple datasets. Grid Search and Random Search improved model performance by tweaking hyperparameters. To improve input feature quality, feature engineering methodologies and data pretreatment methods like missing value handling and categorical variable encoding were examined. Ensemble methods were compared to find the best ones. Shapash study illuminated feature importance in model decision-making. Potential dataset discrepancies were reduced and ethical concerns addressed. This strategy organises our study and helps us predict banking creditworthiness. Metrics assess algorithm performance. These indicators reveal several facets of the model's creditworthiness prediction.

2.3. Imbalanced Data Handling Techniques

2.3.1. Random Over Sampler

The Random Over Sampler is a machine learning technique that addresses class imbalances in datasets. It operates by randomly replicating examples from the minority class until the classes are balanced. This would result in making the instances more uniformly spread across different clusters, which assists the classifier by having a better spread of instances.

2.3.2. SMOTE

SMOTE (Synthetic Minority Oversampling Technique) is a method for addressing imbalanced data in classification tasks. It works by interpolating to create new examples of the minority class that fall between the existing ones. This helps balance the number of examples in each class, making the classifier more effective in both classes.

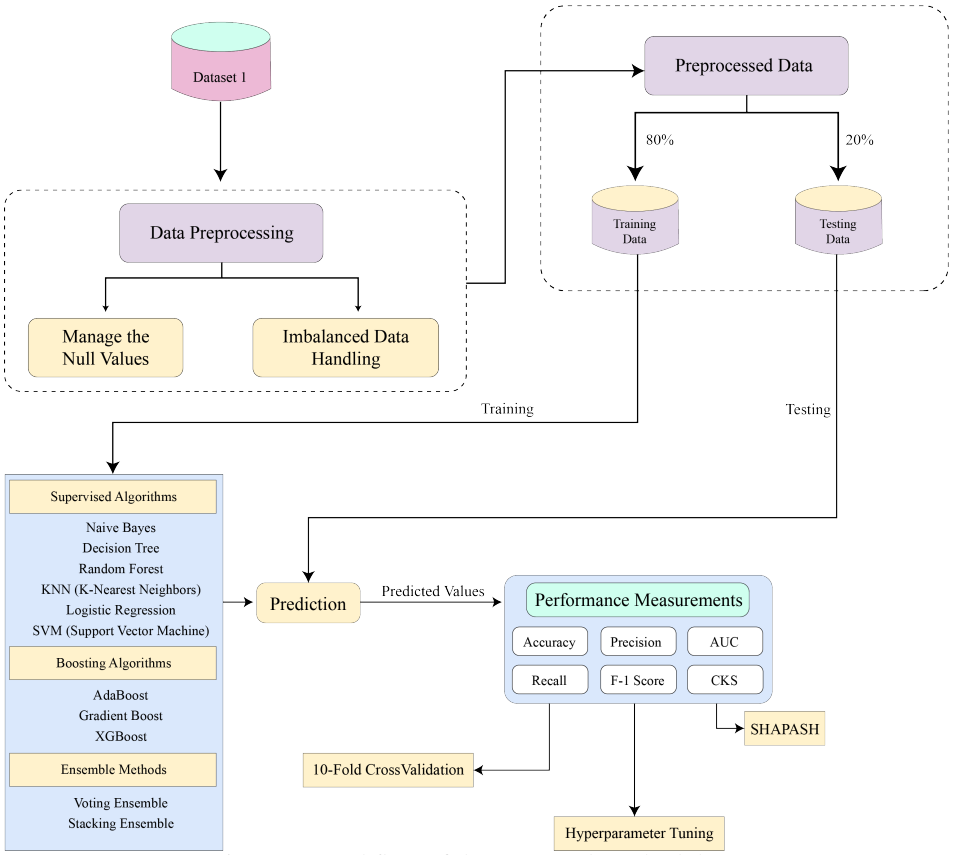

Figure 1. Workflow of the Proposed Methodology

2.3.3. NearMiss

Another way to cope with uneven data is to use NearMiss. To make this work, you pick instances from the majority class that are ”close” to examples from the minority class. It is crucial to choose examples from the majority class, and they must also be fairly representative of other data points in the minority classes to even out the distribution ratio.

2.3.4. SMOTETomek

SMOTE-Tomek is a combination of the SMOTE and Tomek connection technologies. Tomek linkages are examples of things that are close to one another yet belong to different classifications. The SMOTE-Tomek method eliminates Tomek links to further purify the data and improve the classifier's performance.

2.4. Algorithms description

2.4.1. Naive Bayes

Naive Bayes is a type of classifier that predicts outcomes using Bayes' theorem. It's called ”naive” because it assumes that every feature is independent from the others—a simplification that makes calculations much faster. Despite its simple assumptions, Naive Bayes works surprisingly well for sorting information into categories, especially when dealing with large amounts of data or text. It's particularly popular for tasks like detecting spam emails, classifying documents, and analyzing the sentiment of text [26].

2.4.2. Decision Tree

Decision trees are commonly used when you need to predict outcomes for both categories and numbers. They have the advantage of training much faster than neural networks, making them a practical choice for quick predictions and situations where speed matters [27]. Decision trees are called non-parametric, meaning they don't make assumptions about how the data is distributed. They can handle data with lots of features, often leading to more accurate results. Decision trees work especially well when paired with techniques like SMOTE. One challenge, however, is deciding which feature to split on at each step. Two popular ways to

make this decision are Information Gain and the Gini Index. As a tree splits the training data, the disorder—or entropy—changes, and Information Gain measures how much the entropy decreases. Equation:

$$Gain(S, A) = Entropy(S) - \sum_{v}^{A} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$

2.4.3. Random Forest

A random forest is an ensemble method made up of many decision trees working together. Each tree is built using slightly different data and random choices, so they each learn something a little different. When it's time to make a prediction, all the trees "vote" and the most common answer wins. Generally, the more trees in the forest, the better the accuracy [28]. Some forests are built with special techniques like bootstrapping or boosting, which can make them even stronger or give them unique advantages compared to other approaches.

2.4.4. KNN

KNN, or k-nearest neighbors, is a straightforward method used for both classification and regression. To make a prediction about a new data point, KNN looks at the 'k' closest labeled examples in the dataset and chooses the most common label or average value. The basic idea is that similar things tend to be found near each other, so their results are likely to be similar too [29]. In real-world research, data often comes from many sources and isn't always complete—missing values are common. Choosing how to fill in these gaps (imputation) is important for model accuracy. In Python's scikit-learn, the KNN imputer fills in missing values by looking at the closest complete data points, using a distance metric called Euclidean distance. When calculating this distance, it ignores missing values and compares only the available information. The algorithm's equation is:

$$D_{xy} = \sqrt{\text{weight} \cdot \text{squared distance from present coordinates}} \tag{1}$$

where,

$$\text{Weight} = \frac{\text{total number of coordinates}}{\text{number of present coordinates}} \tag{2}$$

2.4.5. Logistic Regression

Logistic regression is a type of classification method that can be used in supervised learning models to guess the probability of a target variable. LR is like linear regression in that it tries to guess the target's probability based on an input feature. This approach has been used for boom classification jobs in water quality [30], but it may also be changed to work with multiclass classification problems. Regarding the binary classification problem, there is a well-known method known as Logistic Regression(LR). The logistic equation, or commonly called the sigmoid function, is among the reasons why LR is so popular. The sigmoid function maps the output of that number to a value between 0 and 1 with its signature S-curve.

$$y = \frac{1}{1 + e^{-value}}$$

2.4.6. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a type of supervised learning algorithm used for classification and regression purposes. It seeks to discover the best hyperplane that maximizes the margin between classes [31]. The SVM algorithm seeks a hyperplane such that for a given dataset, it is defined as follows:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3}$$

where $\mathbf{w}$ is the weight and $\mathbf{x}$ is an input while b being a scalar bias. The SVM aims to achieve the maximum margin $2/|\mathbf{w}|$, but, of course, with all data points of each class on the right side of the hyperplane. SVM can be generalized, using kernel functions to map the input space into higher dimensions, where a separation in the linear case becomes possible.

2.4.7. AdaBoost

AdaBoost is an intelligent method for creating more accurate predictions by combining weaker models, known as weak learners, and boosting them to form a stronger model. The idea is based on giving more emphasis to examples that are difficult to get right: whenever a weak learning algorithm makes an error, AdaBoost increases the importance of that example for learning the next round, [32]. This eventually enables the final model to focus on the problematic examples and make more accurate predictions. The key you're aiming for is simply continual improvement by avoiding the same mistakes.

2.4.8. Gradient Boosting

An alternative approach to learning a combination of many simple models (usually decision trees) to produce a much more accurate prediction [33]. You can use it to predict both categorical and numerical values. The next model in the series attempts to correct the errors of its predecessors. Put them all together and you have a powerful prediction instrument.

In other words, each step in the gradient boosting process is trying to adjust for the errors of its predecessor. The cycle repeats, as the next mini-model learns from the mistakes of those before it. Once all the models are combined, the aggregate prediction is far more accurate than any individual model by itself.

2.4.9. XGBoost

XGBoost Classifier excels in classification. It forms a powerful team of decision trees that focus on regions the previous ones neglected. XGBoost enhances decision trees differently from extreme gradient boosting. Teamwork and accuracy and efficiency skills yield outstanding results [34]. Model performance is measured and improved using metrics including accuracy, precision, recall, and F1 score. This study classifies water quality precisely using XGBoost, a fast supervised learning algorithm. Regularised learning features refine final weights and reduce overfitting, motivating its use. Equation of this algorithm,

$$\Omega(\theta) = \sum_{i=1}^{n} d(y_i, \hat{y}_i) + \sum_{k=1}^{k} \beta(f_k)$$

2.4.10. Voting Ensemble

The Voting Ensemble algorithm improves performance by combining machine learning model predictions. Hard and soft voting are the primary types. The majority vote of the classifiers determines the final prediction in hard voting, while the average of projected probabilities is used in soft voting [35].

For a given set of classifiers $\{h_1, h_2, \ldots, h_n\}$ and an input sample $\mathbf{x}$, the hard voting prediction $\hat{y}$ is given by:

$$\hat{y} = \text{mode}(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_n(\mathbf{x})) \tag{4}$$

In soft voting, the prediction $\hat{y}$ is obtained by averaging the predicted probabilities $\hat{P}(y = c|\mathbf{x})$ for each class c and selecting the class with the highest average probability:

$$\hat{y} = \arg\max_{c} \frac{1}{n} \sum_{i=1}^{n} P_i(y = c|\mathbf{x}) \tag{5}$$

Voting Ensemble algorithms are effective because they leverage the strengths of multiple models, leading to improved accuracy and robustness compared to individual models.

2.4.11. Stacking Ensemble

Stacking Ensemble uses Random Forest (RF), XGBoost (XGB), AdaBoost (ADA), Gradient Boosting (GB), and Decision Tree. A list of base models is used as the estimator's parameter to create this ensemble using the Stacking Classifier class. These base models forecast independently and merge during stacking. The final creditworthiness prediction is made by training a Random Forest Classifier with the aggregated forecasts [36].

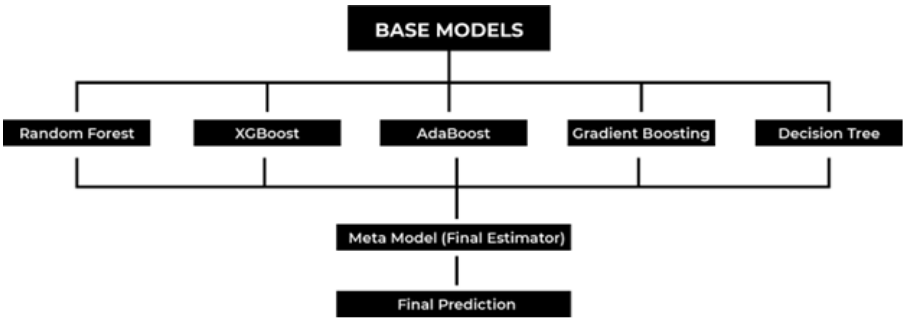

Figure 2. Stacking ensemble structure.

2.5. XAI tools

Shapash Shapash is a Python package that helps you understand and explain models. SHAP (Shapley Additive exPlanations) is better with automated, customizable, and interactive ML model explanations. With Shapash, people of all skill levels can understand model predictions and how features affect them. SHAPash's easy-to-use and interactive graphs of SHAP values help users examine how each feature affects the model's predictions. Force, summary, and dependence graphs illustrate how features are related, how they interact, and what they may indicate. Users can comprehend their models' behavior using Shapash's model comparison, sensitivity analysis, and global feature relevance evaluation. It can work with tree-based, linear, and ensemble ML models; therefore, it can be used in a wide range of fields and applications [37]. Shapash is utilized by data science projects to analyze, debug, and validate models due to its simple interface and excellent visualization tools.

2.6. Performance Metrics

You can use these approaches to quantify performance, including accuracy, precision, recall, F1 score, specificity, AUC, Cohen's Kappa, and others. Accuracy is the number of correctly categorized data points divided by the total number of observations. Precision indicates how accurately the model forecasts the positive case. Recall shows us how many of the good things that happened were remembered as good things. The F-1 score is the harmonic mean of Precision and Recall. Specificity indicates how accurately the model predicts the negative instance. The Area Under the ROC Curve (AUC) indicates how effectively a classifier distinguishes between classes. Cohen's Kappa quantifies the degree of agreement between two raters, taking into account the possibility that they might agree by chance.

3. RESULTS AND DISCUSSION

3.1. Performance of the Models on Dataset

Table 2. Performance of the different models using SMOTE

Algorithm	Accuracy	Precision	Recall	F1 Score	AUC	CKS
Naive Bayes	0.75	0.81	0.75	0.73	0.81	0.5
Decision Tree	0.76	0.76	0.76	0.76	0.76	0.52
RF	0.84	0.85	0.84	0.84	0.91	0.69
KNN	0.62	0.62	0.62	0.62	N/A	0.23
LR	0.74	0.77	0.74	0.73	0.83	0.48
SVM	0.5	0.5	0.5	0.5	0.52	0.38
Ada Boost	0.8	0.81	0.8	0.8	0.89	0.6
Gradient Boosting	0.83	0.84	0.83	0.83	0.92	0.66
XG boost	0.84	0.85	0.84	0.84	0.91	0.69
Voting Classifier	0.82	0.83	0.82	0.82	0.91	0.65
Stacking Classifier	0.82	0.83	0.82	0.82	0.9	0.65

Table 2 shows the performance metrics of machine learning models applied to dataset DS1 after using SMOTE to handle imbalanced data. The models include Naive Bayes, Decision Tree, Random Forest, K-Nearest Neighbours (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Ada Boost, Gradient Boosting, XG Boost, Voting Classifier, and Stacking Classifier. Model performance is measured by accuracy,

precision, recall, F1 score, AUC, and Cohen's Kappa Score. In particular, Random Forest, Gradient Boosting, XG Boost, Ada Boost, and ensemble approaches like Voting Classifier and Stacking Classifier perform well on the dataset and task.

Table 3. Performance of the different models using NearMiss

Algorithm	Accuracy	Precision	Recall	F1 Score	AUC	CKS
Naive Bayes	0.79	0.82	0.79	0.79	0.83	0.59
Decision Tree	0.71	0.72	0.71	0.71	0.71	0.42
RF	0.79	0.84	0.79	0.79	0.89	0.59
KNN	0.62	0.64	0.62	0.62	N/A	0.26
LR	0.75	0.8	0.75	0.74	0.79	0.51
SVM	0.58	0.6	0.58	0.58	0.65	0.18
Ada Boost	0.79	0.81	0.79	0.79	0.87	0.59
Gradient Boosting	0.75	0.79	0.75	0.75	0.85	0.51
XG boost	0.76	0.78	0.76	0.76	0.85	0.53
Voting Classifier	0.78	0.82	0.78	0.77	0.84	0.56
Stacking Classifier	0.75	0.8	0.75	0.74	0.78	0.51

Table 3 shows NearMiss sample performance of various machine learning algorithms, with rows representing algorithms and columns indicating accuracy, precision, recall, and F1 score metrics. Random Forest (RF) has the best accuracy (0.81) and precision (0.85), indicating reliable classification and favourable predictions. By utilising model strengths, ensemble approaches including AdaBoost, Gradient Boosting, XGBoost, Voting Classifier, and Stacking Classifier achieve competitive accuracy of 0.75 to 0.79. However, Support Vector Machine (SVM) performs worse across all metrics, suggesting classification issues. The NearMiss and SMOTE methods yield different performance measures. Compared to SMOTE, NearMiss has inferior accuracy, precision, recall, and F1 score. In this investigation, NearMiss may not improve algorithm performance as much as SMOTE. The different methods of NearMiss (undersampling) and SMOTE (oversampling) to class imbalance lead to these variances. NearMiss may reduce majority class samples, affecting the model's capacity to learn from the majority class. Increasing minority class representation through oversampling appears to ameliorate class imbalance better, as seen by SMOTE's superior performance metrics.

Table 4. Performance of the different models using SMOTETomek

Algorithm	Accuracy	Precision	Recall	F1 Score	AUC	CKS
Naive Bayes	0.75	0.79	0.75	0.74	0.78	0.48
Decision Tree	0.83	0.84	0.83	0.83	0.83	0.66
RF	0.86	0.86	0.86	0.86	0.92	0.72
KNN	0.62	0.62	0.62	0.62	N/A	0.24
LR	0.77	0.8	0.77	0.76	0.8	0.53
SVM	0.48	0.48	0.48	0.48	0.51	-0.03
Ada Boost	0.84	0.84	0.84	0.84	0.86	0.68
Gradient Boosting	0.86	0.87	0.86	0.86	0.91	0.72
XG boost	0.82	0.83	0.82	0.82	0.91	0.65
Voting Classifier	0.87	0.87	0.87	0.87	0.91	0.74
Stacking Classifier	0.87	0.87	0.87	0.87	0.94	0.74

SMOTETomek data handling performance metrics for machine learning algorithms are shown in Table 4. Random Forest (RF) predicts class labels best (0.85, 0.86). XG Boost follows with 0.85/0.86 accuracy and precision. SVM has the lowest accuracy (0.61) and precision (0.62), showing classification difficulties. Voting and Stacking Classifier ensemble techniques often outperform, showing that mixed models can compete. RF, XG Boost, and Naive Bayes have strong recall and F1 Scores, indicating they can collect positive cases and balance precision and recall. KNN and SVM have lower recall and F1 Score, suggesting they may struggle to detect positive cases. Voting Classifier accuracy was 0.83 and Stacking Classifier accuracy was 0.81, proving ensemble methods work. Finally, RF, XG Boost, and Naive Bayes work well, but ensemble techniques increase model performance using SMOTETomek data processing. In conclusion, Random Forest, XG Boost, and Naive Bayes have good classification accuracy, precision, recall, and F1 Score. Poor KNN and SVM. Voting and Stacking Classifier ensemble techniques consistently generate competitive results, showcasing SMOTETomek's multi-model strengths.

The Table 5 summarises RandomOverSampler's performance metrics for several methods on the dataset. The Stacking Classifier and RF (Random Forest) algorithms have the highest accuracy, precision,

Table 5. Performance of the different models using Random Over Sampler

Algorithm	Accuracy	Precision	Recall	F1 Score	AUC	CKS
Naive Bayes	0.69	0.77	0.69	0.66	0.76	0.34
Decision Tree	0.84	0.84	0.84	0.42	0.84	0.67
RF	0.9	0.9	0.9	0.9	0.95	0.77
KNN	0.59	0.59	0.59	0.59	-	0.34
LR	0.68	0.72	0.68	0.67	0.77	0.38
SVM	0.55	0.55	0.55	0.55	0.79	0.38
Ada Boost	0.73	0.74	0.73	0.73	0.81	0.46
Gradient Boosting	0.77	0.78	0.77	0.76	0.86	0.53
XG boost	0.88	0.88	0.88	0.88	0.94	0.71
Voting Classifier	0.9	0.9	0.9	0.9	0.95	0.78
Stacking Classifier	0.93	0.93	0.93	0.93	0.95	0.78

Table 6. Hyperparameters Tuning of Algorithms

Algorithms	Hyperparameter
Random Forest	n_estimators: [100, 200]
XGBoost	n_estimators: [100, 200]
AdaBoost	n_estimators: [50, 100]
Gradient Boosting	n_estimators: [50, 100]
Final Estimator	n_estimators: [100, 200] max_depth: [None, 10, 20]

recall, and F1 scores, 0.93 and 0.9, respectively. These algorithms accurately detect positive instances and minimize false positives. The Decision Tree algorithm has a lower F1 score of 0.42, indicating poorer dataset classification. The SVM algorithm scores lower on all metrics, indicating poor performance compared to other algorithms. Ensemble approaches like Voting and Stacking Classifiers perform well. Ensemble approaches increase accuracy, precision, recall, and F1 scores by integrating various algorithms to maximize model strengths. This shows how ensemble methods improve performance and predictions.

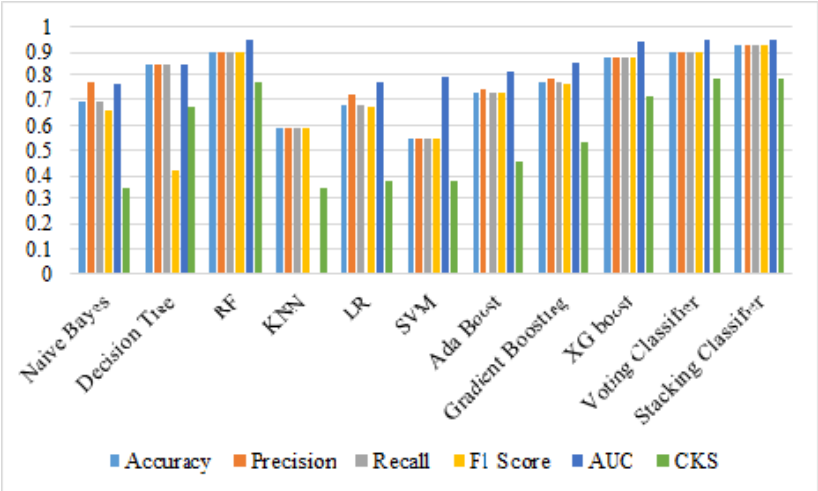

Figure 3. Comparison of models performance using Random Over Sampler Data Balancing

3.2. 10-Fold Cross Validation

In Table 7, which represents the 10-fold cross-validation results for the Stacking Classifier on Dataset, we observe a consistent and commendable performance. The F1 scores range from 0.88 to 0.98, showcasing a robust ability of the model to balance precision and recall across different folds. The corresponding accuracy scores vary from 0.89 to 0.96, with an average accuracy of 0.93. This suggests a high level of accuracy in predicting the target variable across different subsets of the dataset.

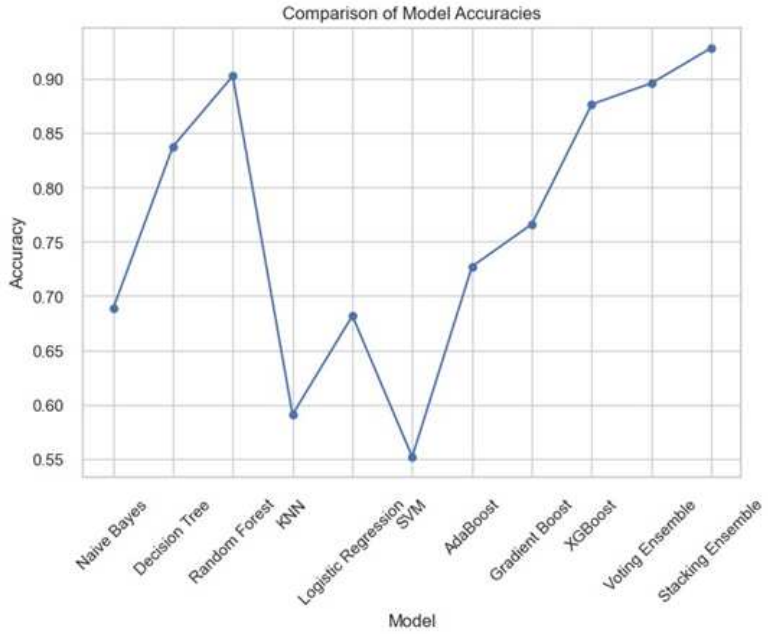

Figure 4. Comparison of models output using Random Over Sampler

Table 7. Result for 10-Fold Cross Validation of Stacking Ensemble Model

Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Avg. Accuracy
Stacking Classifier	0.88	0.89	0.9	0.95	0.98	0.93	0.91	0.89	0.93	0.96	0.93

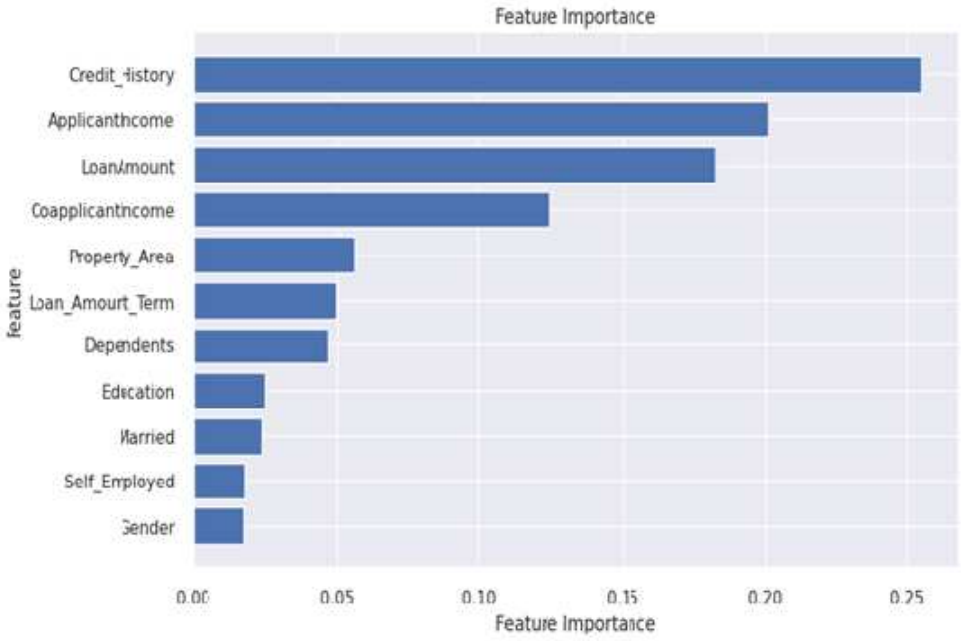

Figure 5. Importance of global characteristics of the prediction

3.3. XAI Analysis

3.3.1. Global Explainability

Figure 5 illustrates the feature importance graph, which displays the relative importance of each feature in relation to the others. In this case, the response variable is one, and the graph illustrates how much each

feature adds to the mean absolute value. CREDIT_HISTORY has the greatest value, which means it is the most important factor in predicting the response variable. This means that when the CREDIT_HISTORY feature is set to one, it is strongly related to the answer variable. Another essential thing in predicting the target variable is that APPLICANTINCOME has a value that is close to CREDIT_HISTORY. CREDIT_HISTORY and APPLICANTINCOME, on the other hand, have higher values than LOANAMOUNT, COAPPLICANTINCOME, PROPERTY_AREA, LOAN_AMOUNT_TERM, DEPENDENTS, and EDUCATION. This means that the relationship between these features and the response variable is not very strong. This means that those factors don't significantly contribute to the response variable in the dataset examined.

3.3.2. Local Explainability

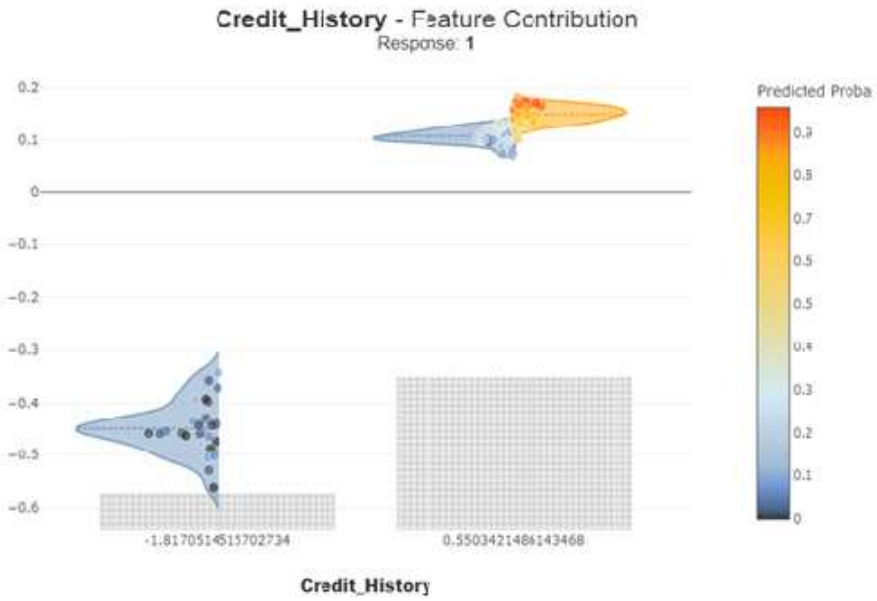

Figure 6. Contribution of the Feature CREDIT_HISTORY in the model

Figure 6 illustrates that a contribution plot is a graph that displays the importance or utility of each feature in a statistical or machine learning model. In this scenario, the plot illustrates how a CREDIT_HISTORY can either help or hurt things. In our methodology, a scenario where an elevated expected outcome correlates with an increase in the CREDIT_HISTORY value is identified as a positive contribution. On the other hand, a plot with a negative contribution indicates that CREDIT_HISTORY is a significant component of the model and has a substantial impact on its decisions.

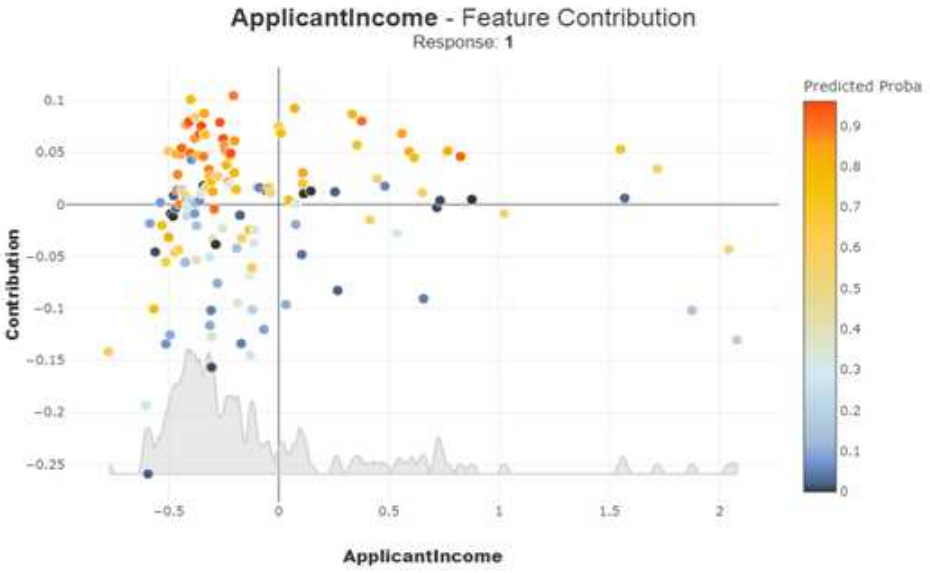

Figure 7. Contribution of the Feature APPLICANTINCOME in the model

Figure 7 shows how the APPLICANTINCOME feature adds to the model by showing its contribution plot. The length of the subgroup is 2000 (90%). Plotting shows that APPLICANTINCOME has a big positive effect on the model and a smaller negative effect.

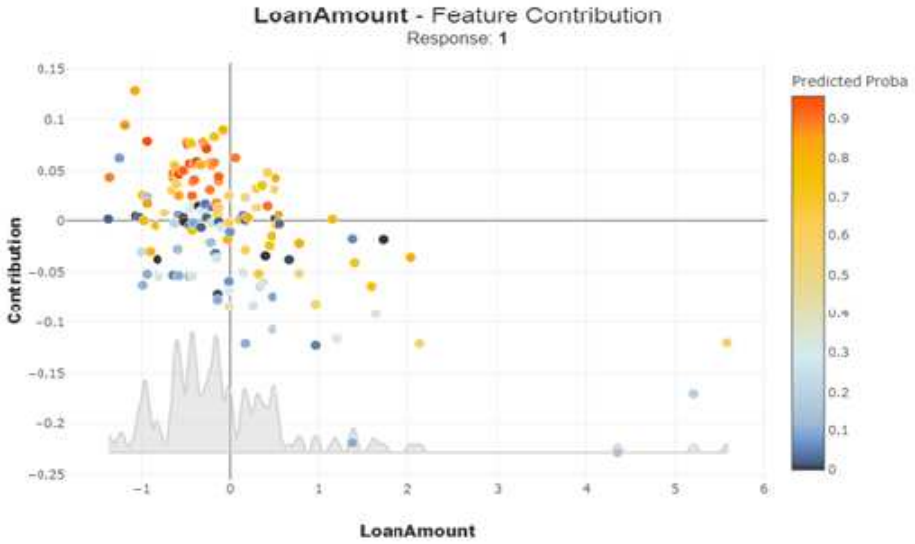

Figure 8. Contribution of the Feature LOANAMOUNT in the model

Figure 8 shows the contribution plot, which illustrates how the loan amount affects a group of 2000 cases (90%). The plot shows that LOANAMOUNT has a significantly positive effect on the model, outweighing any negative effect, which means that LOANAMOUNT is crucial in determining the model's behavior.

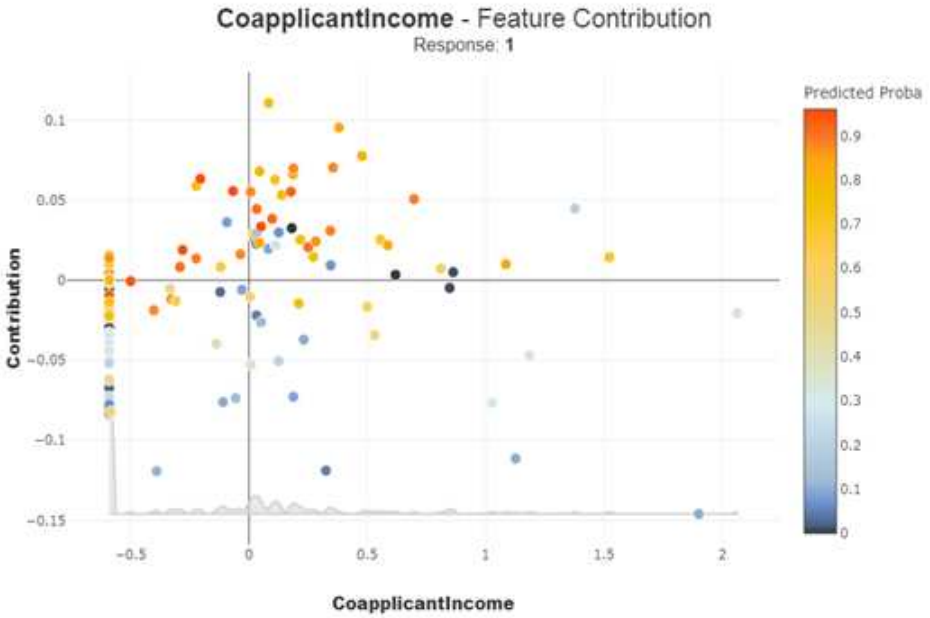

Figure 9. Contribution of the Feature COAPPLICANTINCOME in the model

With a subset length of 2000 (90%), the contribution of COAPPLICANTINCOME is depicted in the contribution plot in Figure 9. COAPPLICANTINCOME has a varied effect on the model, as evidenced by the plot, contributing both positively and negatively. The data indicate that although COAPPLICANTINCOME may be predictively relevant, its influence may be less substantial compared to other variables that have either a greater positive contribution or a lesser negative one.

Figure 10. Contribution of the Feature PROPERTY_AREA in the model

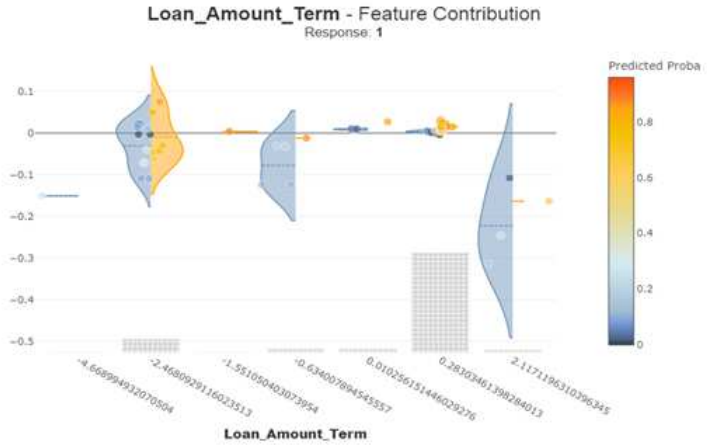

Figure 11. Contribution of the Feature LOAN_AMOUNT_TERM in the model

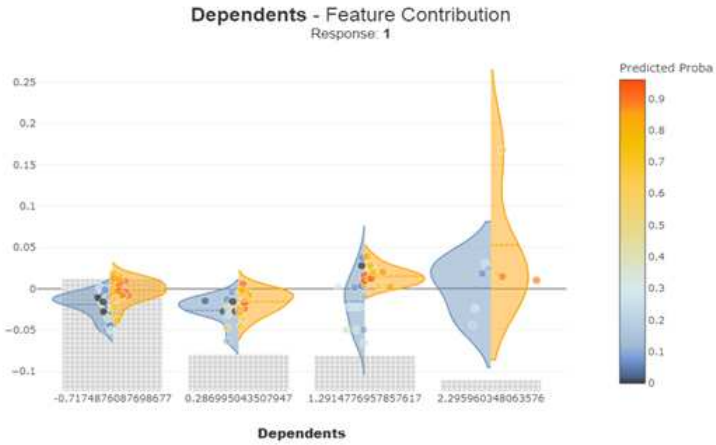

Figure 12. Contribution of the Feature DEPENDENTS in the model

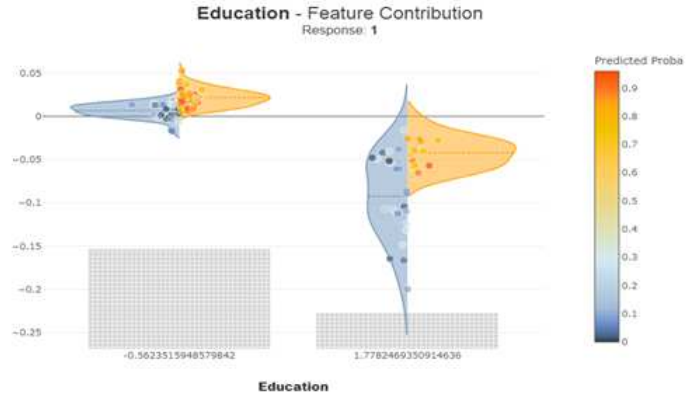

Figure 13. Contribution of the Feature EDUCATION in the model

The contribution plot in Figures 10, 11, 12, and 13 shows that the PROPERTY_AREA, LOAN_AMOUNT_TERM, DEPENDENTS, and EDUCATION characteristics are just as important in the model, using

a subset length of 2000 (90%). Plot analysis indicates that the PROPERTY_AREA, LOAN_AMOUNT_TERM, DEPENDENTS, and EDUCATION features exert a more favorable influence than an unfavorable one, exhibiting a significantly smaller adverse impact.

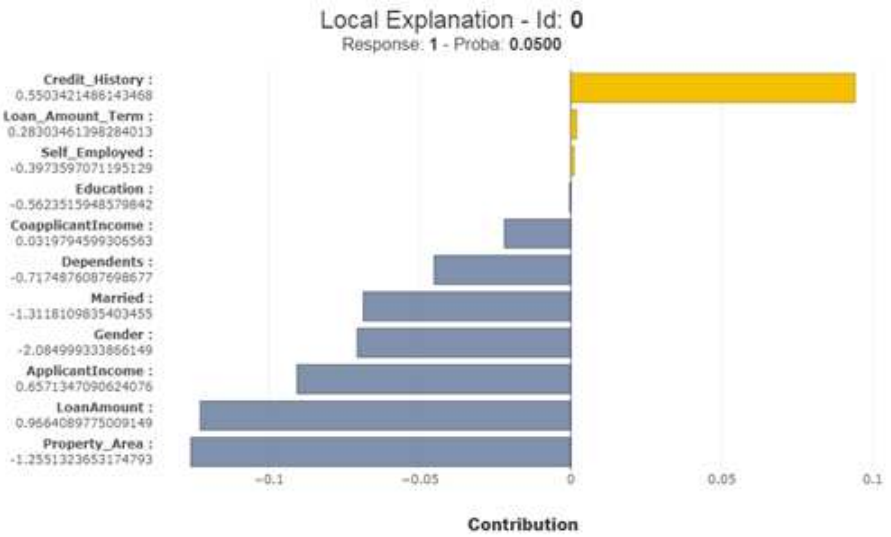

Figure 14. Local explanation of a random id: 0

Figure 14 displays a local explanation for a person with an ID of 900 and a probability value of 0.0500. CREDIT_HISTORY, APPLICANTINCOME, LOANAMOUNT, COAPPLICANTINCOME, and PROPERTY_AREA all have positive values. CREDI_HISTORY has the greatest value of all of them. But DEPENDENTS has a value that is less than zero.

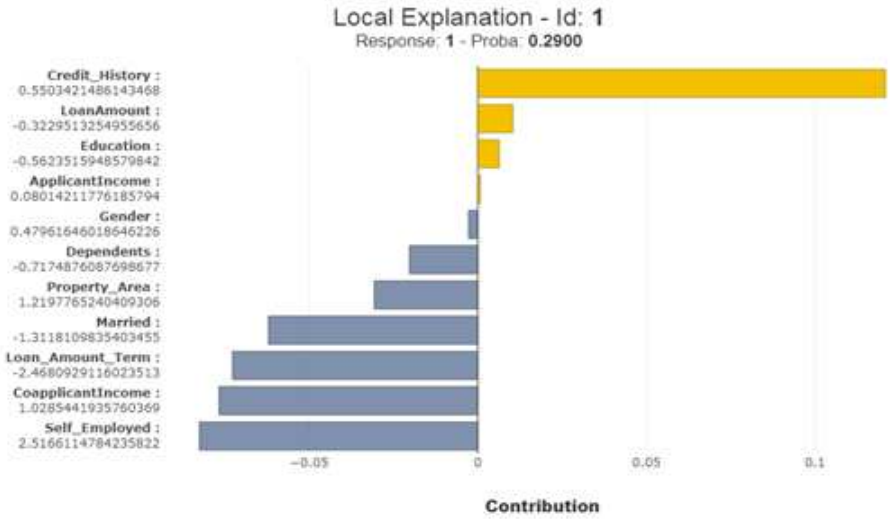

Figure 15. Local explanation of a random id: 1

An individualized explanation for person with ID 900 with the predicted probability of 0.29 is given in Figure 15. In this scenario, CREDIT_HISTORY APPLICANTINCOME LOANAMOUNT COAPPLICANTINCOME and PROPERTY_AREA pushes the probability higher with highest being CREDIT_HISTORY. On the other side if there is more DEPENDENTS then also it increases the probability. The comparison plot in Figure 16 shows the contribution levels for each feature. It was generated by randomly choosing 15 unique IDs. The lines for CREDIT_HISTORY, APPLICANTINCOME, LOANAMOUNT, and PROPERTY_AREA

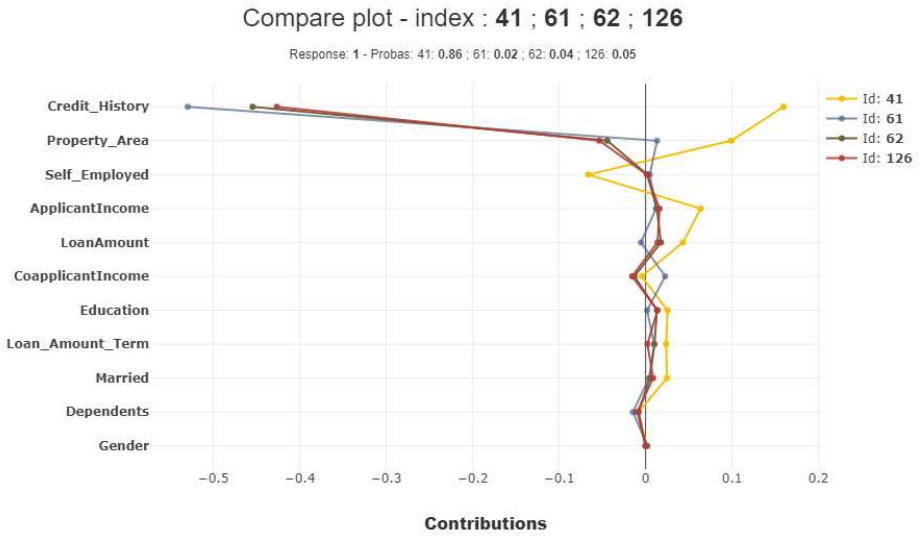

Figure 16. Comparison plot for contribution values of each feature

are more spread out than the lines for the other attributes. The lines for the other features become shorter, indicating that they are less important than the first features.

4. CONCLUSION

This research demonstrates the ability to predict of creditworthiness with several machine learning techniques. The goal is to find the best way to accomplish things by looking at algorithms like Decision Tree, Random Forest, Naive Bayes, KNN, SVM, Logistic Regression, AdaBoost, Gradient Boost, XGBoost, Voting Ensemble, and Stacking Ensemble. After a lot of research, it was determined that the Stacking Ensemble method, when used with random oversampling, worked better than previous algorithms, with an amazing accuracy rate of 93

Future study could focus on real-time implementation as two key areas of interest. Although this paper has offered useful insights into the prediction of creditworthiness using machine learning algorithms, there are various opportunities for further research and enhancement. One could explore advanced sampling approaches like Borderline-SMOTE or ADASYN to better handle class imbalance and potentially enhance algorithm performance. In addition, conducting additional optimisation and refinement of ensemble models, such as fine-tuning hyperparameters and investigating other ensemble configurations, has the potential to improve their performance. Deploying and integrating the created models into real-time creditworthiness prediction systems would guarantee scalability, efficiency, and accuracy in a production context. Ultimately, these endeavours can result in enhanced precision and dependability of forecasting models, empowering financial organisations to make more astute judgements and streamline their lending procedures with greater effectiveness.

REFERENCES

[1] O. Netzer, A. Lemaire, and M. Herzenstein, "When words sweat: Identifying signals for loan default in the text of loan applications," *Journal of Marketing Research*, vol. 56, no. 6, pp. 960–980, 2019.

[2] J. Galindo and P. Tamayo, "Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications," *Computational economics*, vol. 15, pp. 107–143, 2000.

[3] P. Pławiak, M. Abdar, and U. R. Acharya, "Application of new deep genetic cascade ensemble of svm classifiers to predict the australian credit scoring," *Applied Soft Computing*, vol. 84, p. 105740, 2019.

[4] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied soft computing*, vol. 93, p. 106384, 2020.

[5] J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," *Journal of big Data*, vol. 7, pp. 1–33, 2020.

[6] D. Karlan and J. Morduch, "Access to finance," in *Handbook of development economics*. Elsevier, 2010, vol. 5, pp. 4703–4784.

[7] M. A. Sheikh, A. K. Goel, and T. Kumar, "An approach for prediction of loan approval using machine learning algorithm," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 490–494.

[8] X. F. Jency, V. Sumathi, and J. S. Sri, "An exploratory data analysis for loan prediction based on nature of the clients," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 4, pp. 17–23, 2018.

[9] E. Balaji, D. Brindha, and R. Balakrishnan, "Supervised machine learning based gait classification system for early detection and stage classification of parkinson's disease," *Applied Soft Computing*, vol. 94, p. 106494, 2020.

[10] B. He, Y. Shi, Q. Wan, and X. Zhao, "Prediction of customer attrition of commercial banks based on svm model," *Procedia computer science*, vol. 31, pp. 423–430, 2014.

[11] L. Simieli, L. T. B. Gobbi, D. Orcioli-Silva, V. S. Beretta, P. C. R. Santos, A. M. Baptista, and F. A. Barbieri, "The variability of the steps preceding obstacle avoidance (approach phase) is dependent on the height of the obstacle in people with parkinson's disease," *Plos one*, vol. 12, no. 9, p. e0184134, 2017.

[12] V. Singh, A. Yadav, R. Awasthi, and G. N. Partheeban, "Prediction of modernized loan approval system based on machine learning approach," in *2021 International Conference on Intelligent Technologies (CONIT)*. IEEE, 2021, pp. 1–4.

[13] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert systems with applications*, vol. 39, no. 3, pp. 3446–3453, 2012.

[14] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[15] R. Odegua, "Predicting bank loan default with extreme gradient boosting," *arXiv preprint arXiv:2002.02011*, 2020.

[16] C. Kwofie, C. Owusu-Ansah, and C. Boadi, "Predicting the probability of loan-default: An application of binary logistic regression," *Research Journal of Mathematics and Statistics*, vol. 7, no. 4, pp. 46–52, 2015.

[17] A. S. Kadam, S. R. Nikam, A. A. Aher, G. V. Shelke, and A. S. Chandgude, "Prediction for loan approval using machine learning algorithm," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 04, 2021.

[18] A. S. Aphale and S. R. Shinde, "Predict loan approval in banking system machine learning approach for cooperative banks loan approval," *International Journal of Engineering Trends and Applications (IJETA)*, vol. 9, no. 8, 2020.

[19] M. Srinivasa Rao, C. Sekhar, and D. Bhattacharyya, "Comparative analysis of machine learning models on loan risk analysis," in *Machine Intelligence and Soft Computing: Proceedings of ICMISC 2020*. Springer, 2021, pp. 81–90.

[20] Q. Xing, C. Yu, S. Huang, Q. Zheng, X. Mu, and M. Sun, "Enhanced credit score prediction using ensemble deep learning model," *arXiv preprint arXiv:2410.00256*, 2024.

[21] R. Rofik, R. Aulia, K. Musaadah, S. S. F. Ardyani, and A. A. Hakim, "The optimization of credit scoring model using stacking ensemble learning and oversampling techniques," *Journal of Information System Exploration and Research*, vol. 2, no. 1, 2024.

[22] N. Uddin, M. K. U. Ahamed, M. A. Uddin, M. M. Islam, M. 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, 2023.

[23] Y. Chen, R. Calabrese, and B. Martin-Barragan, "Interpretable machine learning for imbalanced credit scoring datasets," *European Journal of Operational Research*, vol. 312, no. 1, pp. 357–372, 2024.

[24] X. Li, H. Zheng, K. Tao, and M. Mao, "Implementation of an asymmetric adjusted activation function for class imbalance credit scoring," *arXiv preprint arXiv:2501.12285*, 2025.

[25] Srikanth, "Home loan approval prediction," https://www.kaggle.com/code/srikanth917/home-loan-approval-prediction, 2024, accessed: September 12, 2025.

[26] D. Berrar, "Bayes' theorem and naive bayes classifier," 2019.

[27] Z. A. Ali, Z. H. Abduljabbar, H. A. Taher, A. B. Sallow, and S. M. Almufti, "Exploring the power of extreme gradient boosting algorithm in machine learning: A review," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, 2023.

[28] M. Mahedi Hassan, M. Fazle Rabbi, M. Hasan, and B. Roy, "An ensemble machine learning approach to classify parkinson's disease from voice signal," in *International Conference on Big Data, IoT and Machine Learning*. Springer, 2023, pp. 575–590.

[29] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1774–1785, 2017.

[30] M. Hasan, M. M. Islam, S. W. Sajid, and M. M. Hassan, "The impact of data balancing on the classifier's performance in predicting cesarean childbirth," in *2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*. IEEE, 2022, pp. 1–4.

[31] S. Amarappa and S. Sathyanarayana, "Data classification using support vector machine (svm), a simplified approach," *Int. J. Electron. Comput. Sci. Eng*, vol. 3, pp. 435–445, 2014.

[32] M. Hasan, U. Das, R. K. Datta, and M. Z. Abedin, "Model development for predicting the crude oil price: Comparative evaluation of ensemble and machine learning methods," in *Novel financial applications of machine learning and deep learning: Algorithms, product modeling, and applications*. Springer, 2023, pp. 167–179.

[33] P. Mahajan, S. Uddin, F. Hajati, and M. A. Moni, "Ensemble learning for disease prediction: A review," in *Healthcare*, vol. 11, no. 12. MDPI, 2023, p. 1808.

[34] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[35] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *Ieee Access*, vol. 7, pp. 82 512–82 521, 2019.

[36] A. Morshed-Bozorgdel, M. Kadkhodazadeh, M. Valikhan Anaraki, and S. Farzin, "A novel framework based on the stacking ensemble machine learning (seml) method: application in wind speed modeling," *Atmosphere*, vol. 13, no. 5, p. 758, 2022.

[37] M. M. Hassan, M. F. Abrar, and M. Hasan, "An explainable ai-driven machine learning framework for cybersecurity anomaly detection," in *Cyber Security and Business Intelligence*. Routledge, 2023, pp. 197–219.