big data and cognitive computing

MDPI

Article

Enhancing Supervised Model Performance in Credit Risk Classification Using Sampling Strategies and Feature Ranking

Niwan Wattanakitrungroj [1,*], Pimchanok Wijitkajee [1], Saichon Jaiyen [1], Sunisa Sathapornvajana [1] and Sasiporn Tongman [2,*]

[1] School of Information Technology, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand; pimchanok.wiji@kmutt.ac.th (P.W.); saichon.jai@kmutt.ac.th (S.J.); sunisa.sat@kmutt.ac.th (S.S.)

[2] Department of Biotechnology, Faculty of Science and Technology, Thammasat University, Khlong Luang 12120, Pathum Thani, Thailand

* Correspondence: watta.niwan@gmail.com (N.W.); tongman.sas@gmail.com (S.T.)

Abstract: For the financial health of lenders and institutions, one important risk assessment called credit risk is about correctly deciding whether or not a borrower will fail to repay a loan. It not only helps in the approval or denial of loan applications but also aids in managing the non-performing loan (NPL) trend. In this study, a dataset provided by the LendingClub company based in San Francisco, CA, USA, from 2007 to 2020 consisting of 2,925,492 records and 141 attributes was experimented with. The loan status was categorized as "Good" or "Risk". To yield highly effective results of credit risk prediction, experiments on credit risk prediction were performed using three widely adopted supervised machine learning techniques: logistic regression, random forest, and gradient boosting. In addition, to solve the imbalanced data problem, three sampling algorithms, including under-sampling, over-sampling, and combined sampling, were employed. The results show that the gradient boosting technique achieves nearly perfect *Accuracy*, *Precision*, *Recall*, and *F1 score* values, which are better than 99.92%, but its *MCC* values are greater than 99.77%. Three imbalanced data handling approaches can enhance the model performance of models trained by three algorithms. Moreover, the experiment of reducing the number of features based on mutual information calculation revealed slightly decreasing performance for 50 data features with *Accuracy* values greater than 99.86%. For 25 data features, which is the smallest size, the random forest supervised model yielded 99.15% *Accuracy*. Both sampling strategies and feature selection help to improve the supervised model for accurately predicting credit risk, which may be beneficial in the lending business.

Keywords: credit risk classification; machine learning; imbalance handling; feature ranking

Citation: Wattanakitrungroj, N.; Wijitkajee, P.; Jaiyen, S.; Sathapornvajana, S.; Tongman, S. Enhancing Supervised Model Performance in Credit Risk Classification Using Sampling Strategies and Feature Ranking. *Big Data Cogn. Comput.* **2024**, *8*, 28. https://doi.org/10.3390/bdcc8030028

Academic Editor: Domenico Talia

Received: 23 January 2024
Revised: 18 February 2024
Accepted: 1 March 2024
Published: 6 March 2024

1. Introduction

Machine learning techniques have several benefits in various applications, especially in the form of predicting a trend or outcome. Hence, machine learning models can accurately assess credit default probabilities and improve credit risk prediction [1]. Focusing on financial services like personal loans, accurately predicting the risk of non-performing loans (NPLs) in peer-to-peer (P2P) lending is one crucial thing for lenders such as P2P lending platforms. When borrowers fail to repay (or default on) their loans, it brings about an NPL for the lenders. Generally, an NPL is a major task to overcome in order to reach the stability and profitability of not only financial institutions [2] but also P2P platforms. So, risk assessment measures, diversification strategies, and collection processes are always performed to minimize the NPL issue. These P2P platforms, which are widely used in many countries, are involved with higher risk than traditional lending, because they depend on individuals [3]. However, there are many advantages superior to banking credit, i.e., lenders' and borrowers' direct interaction, detailed credit scoring [4], and the opportunity to gather and analyze large numbers of data which can be used to assess trustability and

reduce risks [5]. Therefore, several previous research works have been studied to build an efficient model to predict the risk of lending [6–8]. Still, there are several challenges, including selecting important features, coping with imbalanced data, handling data quality, and experimenting with in-depth model evaluations. Based on lending datasets, they often contain imbalanced data, i.e., a higher proportion of good loans than risky ones. This possibly leads to the model's prediction bias. In addition, resolving missing values in the data needs mindful consideration of whether to impute, remove, or ignore them. In addition, selecting a relevant and informative input feature set is one substantial step for avoiding model overfitting or underfitting. Apart from that, to ensure the efficient model's real-world performance, one way is to validate it on many lending datasets. In summary, covering these above challenges may be able to result in successfully developing a reliable lending risk prediction model.

In this research, to overcome the challenges associated with building a machine learning model for this lending risk prediction problem, various approaches were implemented and contributed. Firstly, exploratory data analysis (EDA) to explore and clean the data was conducted, which aimed to adjust data quality before initiating the model creation process. Secondly, logistic regression (LG), random forest (RF), and gradient boosting (GB), which are supervised machine learning approaches, were used for model building experiments. Thirdly, over-sampling, under-sampling, and combined sampling techniques to mitigate the imbalanced data problem were comparatively employed. Lastly, an experiment on reducing feature number according to its importance computed by mutual information was also performed.

The remaining sections of this paper are organized as follows. A brief literature review about the machine learning approaches and imbalanced data handling techniques utilized in this study is provided in Section 2. The methodology such as material data description, data preparation, experimental setup, and performance evaluations is outlined in Section 3. In Section 4, the results and discussions are reported. Finally, in Section 5, the conclusion and future works are summed up.

2. Related Works

2.1. Literature Review

Lately, various machine learning algorithms have been applied in the lending risk assessment problem [9], for example, logistic regression, variance in decision trees [10], neural networks and deep learning [11], as well as ensemble approaches [12–14]. One important issue is the imbalanced data problem. Commonly, the number of good credit customers is much greater than that of bad ones. This problem needs to be mitigated, since many machine learning algorithms cannot well handle it, leading to biased predictive models. Consequently, many wrong predictions bring about lenders' financial losses. Therefore, variously proposed techniques to handle imbalanced data have been offered by researchers. Some examples are as follows. Ref. [15] offered the under-sampling method in their resampling ensemble model called REMDD for imbalanced credit risk evaluation in P2P lending. In the work [16], the ADASYN (adaptive synthetic sampling approach) [17] was adopted for reducing the class imbalance problem. Meanwhile, ref. [18] proposed quite balanced datasets, yielded by employing the under-sampling technique for creating models to predict the default risk of P2P lending. Focusing on datasets previously used in this research domain, the LendingClub dataset is one famously public dataset. It is from a lending platform in the United States. There are several LendingClub dataset versions which have been used in many works, as exemplified in Table 1.

Table 1. Research paper examples of various versions of LendingClub data.

Research	LendingClub Data	Imbalance Solving	ML	Best Performance
[19]	Year: 2013–2015 Samples: 656,724 Features: 115 Classes: 2 ({good}; {bad})	-	Random forest Decision tree Bagging	*Accuracy*: 0.885
[20]	Year: 2012–2013 Samples: 164,620 Features: 34 Classes: 2 ({Charged Off, Late (31–120 days), Default}; {Fully Paid})	-	BPSOSVM + Extremely randomized tree	*Accuracy*: 0.64 *Precision*: 062 *Recall*: 0.65 *F1 score*: 0.61
[9]	Year: 2016–2017 Samples: 877,956 Features: 151 Classes: 2 ({Fully Paid}; {Charged Off})	Under-sampling Over-sampling Hybrid	Logistic regression Random forest MLP	*Accuracy*: 0.64 *AUC*: 0.71 *TPR*: 0.66 *TNR*: 0.64
[16]	Year: 2019 Samples: 128,262 Features: 150 Classes: no details	ADASYN	Fusion model (logistic regression, random forest, and CatBoost)	*Accuracy*: 0.994 *Recall*: 0.99 *F1 score*: 0.99
[14]	Year: 2007–2015 Samples: 9578 Features: 14 Classes: 2 ({not.fully.paid}; {fully.paid})	SMOTE	LGBFS + StackingXGBoos	*Accuracy*: 0.9143 *Recall*: 0.9151 *F1 score*: 0.9165
[14]	Year: 2012–2018 Samples: 2,875,146 Features: 18 Classes: 2 loan_status $\in \{0, 1\}$	SMOTE	LGBFS + StackingXGBoos	*Accuracy*: 0.99982 *Recall*: 0.9999 *F1 score*: 0.9999
[21]	Year: 2007–2016 Samples: 396, 030 Features: 27 Classes: 2 ({Fully Paid}; {Charged Off})	SMOTE	LGB-XGB-Stacking	*Accuracy*: 0.8940 *Recall*: 0.7131 *AUC*: 0.7975

In research communities, the lending prediction problem is currently active. One major challenge of this problem is how to effectively solve imbalanced data for machine learning model training. Due to a lot of attributes in the lending dataset, one challenge is how to effectively reduce data dimension. Recently, random forest classifiers combining with either the feature selection method [22] or the imbalanced data handling technique [23] showed good predictive results. Apart from that, the LendingClub dataset is still a widely used public dataset in numerous research studies. This presents several challenges such as its big volume, rapidly increasing volume size, high data dimensionality, missing data occurrence, and massive imbalanced data. To create efficient models for loan status prediction and eventually decrease risks within the lending system, rigorous data exploration should be performed to cope with such a messy dataset.

2.2. Machine Learning Approaches

In this study, three machine learning approaches, i.e., logistic regression, random forest, and gradient boosting, were applied to create models for predicting loan statuses. A brief overview of each algorithm is provided below.

2.2.1. Logistic Regression (LR)

Logistic regression [24] is based on a statistical approach primarily designed for solving binary classification problems, where the output has two categorical classes. The probability of an input belonging to a specific class using the logistic function (sigmoid function) is calculated via Equation (1).

$$h_\theta(x^{(i)}) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n)}} \tag{1}$$

where $x_0, x_1, \ldots, x_n$ are the values of input features and the associated parameters updated during the learning process are represented as $\theta_0, \theta_1, \ldots, \theta_n$. These parameters are then returned as the model for prediction. The output is in the range of 0 to 1, indicating the probability of the input belonging to the positive class. Logistic regression is simple to interpret and efficient, especially in situations where the relationship between the features and the binary output is assumed to be linear. Logistic regression is currently popular for building predictive analyses in financial research [25–28].

2.2.2. Random Forest (RF)

The random forest algorithm [29] is an ensemble learning method widely used for classification and regression tasks. The data are separated into M subsets for creating M decision trees with several parameters involved in the creation of decision trees such as the maximum depth of the tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node. A forest of decision trees is constructed during the training phase. Each decision tree is created using a subset of the training data and a random subset of features at each split, presenting diversity among the trees. Random forest is based on the bagging technique, in which multiple subsets of the training data (with replacement) are used to train individual trees, thereby reducing overfitting and improving generalization. Additionally, random feature selection at each split ensures that the trees are less correlated, resulting in a more robust ensemble. For classification tasks, the voting process involves counting the votes for each class from all the decision trees, and the class with the most votes is chosen as the final prediction. Mathematically, if M is the number of trees in the random forest and V_{ij} is the vote count for class j by tree i, the final predicted class y_{pred} is determined via Equation (2).

$$y_{\text{pred}} = \text{argmax}_j \left(\sum_{i=1}^{M} V_{ij} \right) \tag{2}$$

where y_{pred} is the predicted class and argmax_j returns the class j that maximizes the sum of votes across all trees. Random forest has been widely applied to various problems in the lending domain [30–32].

2.2.3. Gradient Boosting (GB)

Gradient boosting [33] is a machine learning algorithm that operates by sequentially improving the performance of weak learners, typically decision trees, to create a strong predictive model. The algorithm works in an iterative manner, adding new weak learners to correct the errors made by the existing ensemble. An initial prediction for each class is often set by assigning balanced probabilities to each class. Subsequently, the pseudo-residuals for data input i and class j, denoted as r_{ij}, are calculated via Equation (3).

$$r_{ij} = y_{ij} - F_{m-1}(x_i) \tag{3}$$

where y_{ij} is the true class label for data input i and class j. $F_{m-1}(x_i)$ is the predicted class probability for data input i from the model at iteration $m - 1$. The pseudo-residuals, r_{ij},

represent the disparity between the true class labels y_{ij} and the current predicted class probabilities. The iterative update of the class probabilities is derived via Equation (4):

$$F_m(x) = F_{m-1}(x) + \eta \sum_{i=1}^{N} \gamma_m h_m(x_i) \quad (4)$$

where η is the learning rate, controlling the contribution of each weak learner to the overall model. $h_m(x_i)$ represents the prediction made by the weak learner for the data input i at iteration m. γ_m is the weight assigned to the output of the weak learner at iteration m. This weight is determined during the training process and is chosen to minimize the overall loss of the model. The final prediction for a given input in a classification task is determined by selecting the class with the highest cumulative probability of all weak learners. Mathematically, the predicted output is expressed as:

$$\hat{y}_i = \text{argmax}_j F_M(x_i) \quad (5)$$

where $\hat{y}_i$ is the predicted class for data input i. $F_M(x_i)$ is the cumulative sum of contributions from all weak learners up to the final iteration (M) for data input i. Gradient boosting has recently gained popularity for risk prediction in the financial domain [10,34–37].

2.3. Resampling Imbalanced Data

Three widely used approaches to handle imbalanced data were applied, including the over-sampling, under-sampling, and combined sampling approaches. Their details are explained as follows.

2.3.1. Over-Sampling Approach

In the first approach, SMOTE (Synthetic Minority Over-sampling Technique) [38] was employed for generating synthetic data of the minority class to create a more balanced dataset. The basic idea behind SMOTE is to create synthetic data by interpolating between the existing data of the minority class. Let x_i be a data point from the minority class and x_{zi} be one of its k nearest neighbors (i.e., it is selected randomly). Also, let λ be a random number between 0 and 1. The synthetic data point x_{new} is created using the formula:

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i) \quad (6)$$

This equation represents a linear interpolation between the original minority class data point x_i and one of its k nearest neighbors x_{zi}. The parameter λ determines the amount of interpolation, and it is randomly chosen for each synthetic data point. In summary, the steps of SMOTE are as follows.

(1) Select a minority class data point x_i.
(2) Find its k nearest neighbors (e.g., x_{zi}).
(3) Randomly select one of the neighbors x_{zi}.
(4) Generate a random number λ between 0 and 1.
(5) Use the formula to create a synthetic instance x_{new}.
(6) Repeat steps (1)–(5) for the desired number of synthetic data points.

This process helps balance the class distribution by creating synthetic data points along the line segments connecting existing minority class data points, consequently solving the class imbalance issue in the dataset.

2.3.2. Under-Sampling Approach

Under-sampling for handling imbalanced data problems involves reducing the size of the majority class to balance it with the minority class. In this approach, the data are randomly selected from the majority class to achieve a more balanced class distribution [39]. Unlike SMOTE, which involves creating synthetic data, the random under-sampling simply

removes examples from the majority class randomly. We assume x_i is a data point from the majority class, N is the total number of data points in the majority class, and N_{new} is the desired number of data points after under-sampling. The basic idea is to randomly select N_{new} data points from the majority class without replacement. The processes of random under-sampling are as follows.

(1) Calculate the sampling ratio: ratio $= \frac{N_{new}}{N}$.
(2) For each data point x_i in the majority class:
    (2.1) With probability ratio, keep x_i.
    (2.2) With probability $1 -$ ratio, discard x_i.

This process is repeated until N_{new} data points are selected, achieving the desired sample class distribution.

2.3.3. Combined Sampling Approach

For the combined sampling approach, we use SMOTEENN, which is a combination of over-sampling using SMOTE and under-sampling using edited nearest neighbors (ENN) [40]. The goal is to address imbalanced data by first generating synthetic data points with SMOTE and then cleaning the dataset using edited nearest neighbors to remove potentially noisy examples. After applying SMOTE to generate synthetic data points, edited nearest neighbors is used to remove data points that are considered noisy or misclassified.

(1) Identify data points in the dataset that are misclassified.
(2) For each misclassified data point, check its k nearest neighbors.
    (2.1) If the majority of the neighbors have a different class label, remove the misclassified data point.

This process helps to improve the overall quality of the dataset by eliminating noisy points introduced during the over-sampling process.

3. Materials and Methodology

3.1. Data Description and Preprocessing

In this work, financial data provided by the LendingClub Company from 2007 to 2020Q3 [41] were used. The data consist of 2,925,493 records which are divided into various loan statuses as in Table 2. The loan statuses are categorized as "Good" or "Risk" users. The "Fully Paid" status is categorized as "Good" users, whereas "Charged Off", "In Grace Period", "Late (16–30 days)", "Late (31–120 days)", and "Default" are grouped as "Risk" users. The "Current" status is not explicitly categorized as "Good" or "Risk" since it represents the current state of ongoing payments. "Issued" is also not classified as it may refer to loans that are approved but not yet active. "Does not meet the credit policy" users were excluded in this study. For the experiment, the data contain 1,497,783 samples labeled as "Good" and 391,882 samples labeled as "Risk", totaling 1,889,665 samples. This is a two-class dataset with an imbalance ratio (IR) equal to 3.82, which shows a slight class imbalance, as displayed in Figure 1. IR is the majority class size divided by the minority class size. A high IR value may affect model performance in some machine learning algorithms, i.e., the majority class is more correctly predicted than the minority class due to imbalanced training data causing model bias. However, the selected algorithms used in this paper like random forest and gradient boosting are quite robust for mild class imbalance. It is more helpful to solve imbalanced data before model training because of the increasing chance of model performance improvement. In addition, the imbalanced data handing methods used in this work are explained in Section 2.3.

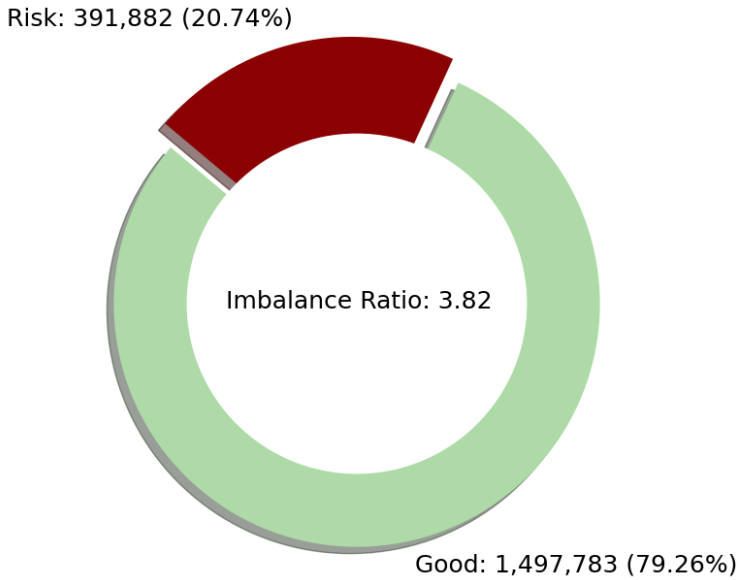


Figure 1. Class imbalance of our experimental dataset from LendingClub dataset.

Table 2. Dataset from LendingClub company from 2007 to 2020Q3 and loan status distribution.

Loan Status	Count	Label
"Fully Paid"	1,497,783	"Good"
"Charged Off"	362,548	"Risk"
"In Grace Period"	10,028	"Risk"
"Late (16–30 days)"	2719	"Risk"
"Late (31–120 days)"	16,154	"Risk"
"Default"	433	"Risk"
"Current"	1,031,016	-
"Issued"	2062	-
"Does not meet the credit policy. Status: Fully Paid"	1988	-
"Does not meet the credit policy. Status: Charged Off"	761	-
Total	2,925,493	

There are 141 attributes in the original data which contain many missing values, as illustrated in Figure 2, with high percentages. Some columns need to be dropped and transformed before training the models. The data were preprocessed through the following steps.

(1) Drop column "id" because it typically serves as a unique identifier for each row, and including it as a feature could lead the model to incorrectly learn patterns that are specific to certain ids rather than generalizing well to new data.
(2) Drop "url" because it might not provide meaningful information for your model, or its content might be better represented in a different format.
(3) Drop columns "pymnt_plan" and "policy_code" because every record in the "pymnt_plan" column has the value "n" and every record in the "policy_code" column has the value 1. These columns contain constant values, resulting in the model being unable to differentiate between different data inputs.
(4) Drop columns that have missing values exceeding 50%. The selected dataset now comprises 101 columns, including 100 features and the loan status.
(5) In the "int_rate" and "revol_util" columns, convert the percentage values from string format to float.
(6) For categorical data, fill the missing values with the mode and transform them into numerical values.
(7) For real value data, fill the missing values with the mean of the existing values.

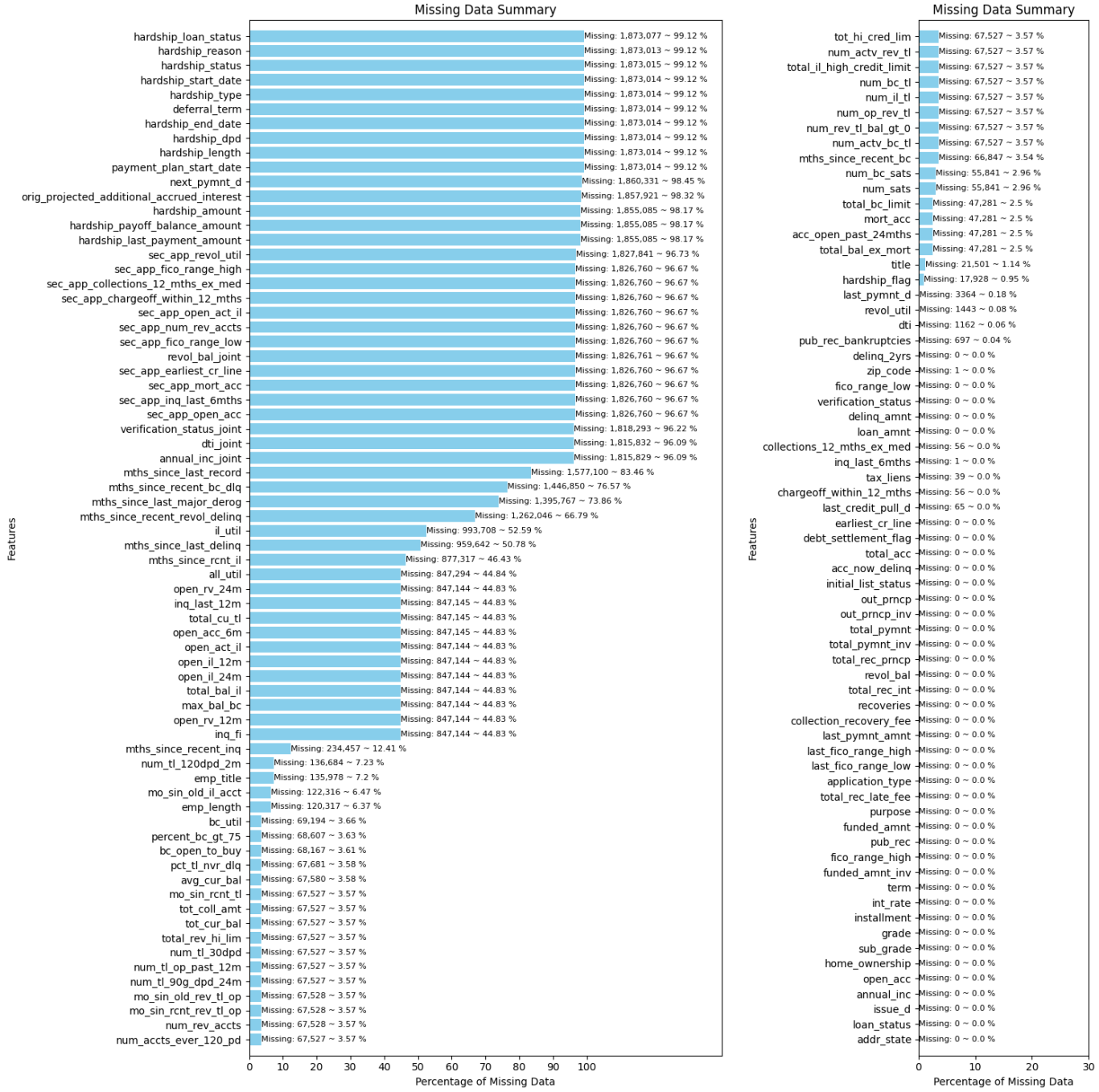


Figure 2. The summary of missing values on each attribute excluding the "id", "url", "pymnt_plan", and "policy_code" attributes.

Now, the dataset comprises 100 features. Each feature was explored in the relationship with its target variable (class label) to rank the importance of features. Mutual information (MI) can identify informative features on both linear and non-linear relationships between features and target variables. In feature selection, a feature with a higher mutual information value is considered as more important and typically selected into a training feature set. The importance of these features can be represented by mutual information, as defined in Equation (7).

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \cdot \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \quad (7)$$

where $p(y)$, $p(x,y)$, and $p(x)$ represent the probabilities associated with the target variable Y and the joint and marginal distributions of features X and Y, respectively. The mutual information values for all features are presented in Figure 3. These were used to investigate the impact of feature selection on the performance of the models. The correlation matrix for the first 25 features with the highest mutual information and the class label is depicted in Figure 4. Each cell in the table shows the correlation between two variables. It is often used to understand the relationships between different variables in a dataset. The values range from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation.

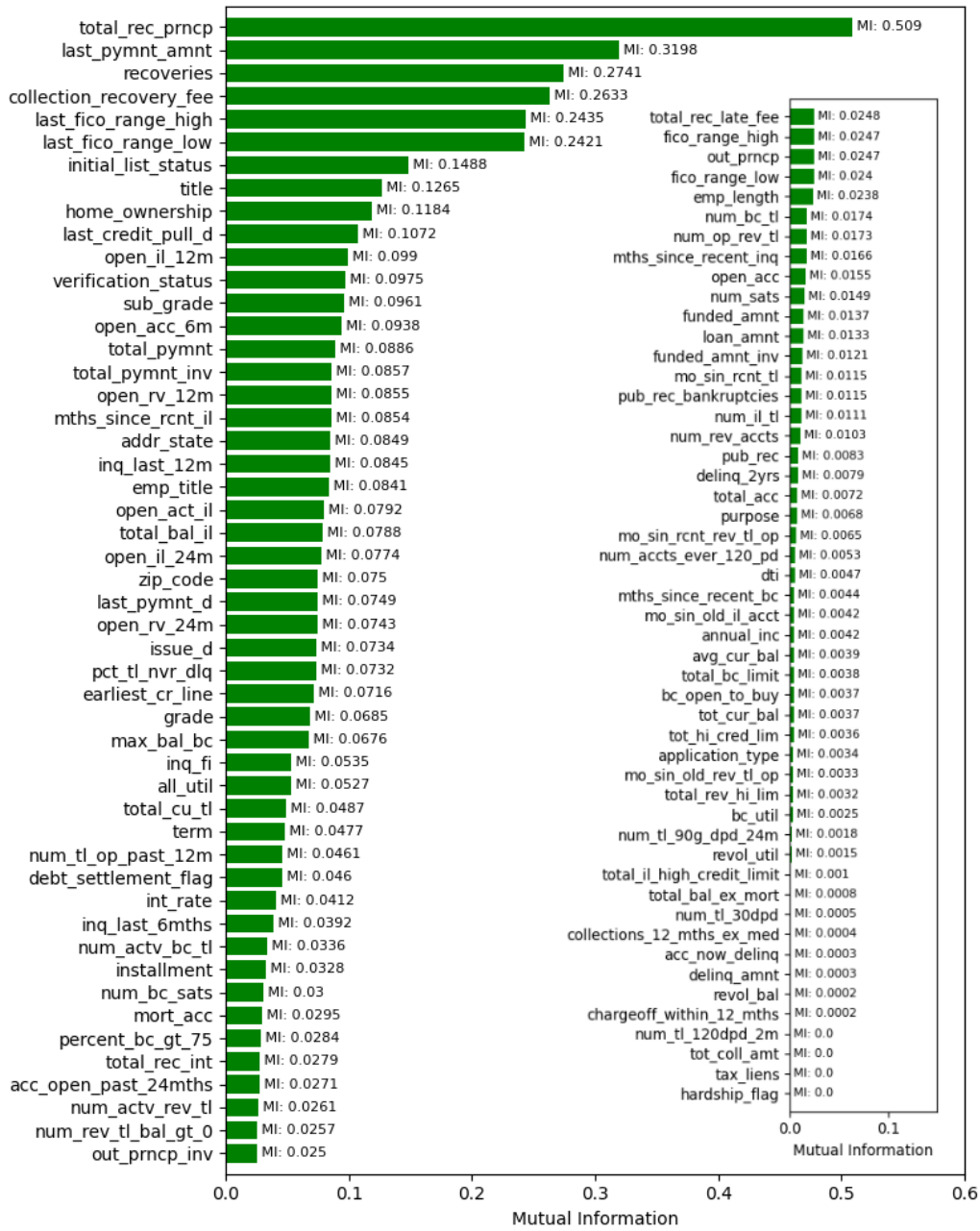


Figure 3. A summary of mutual information (MI) across the 100 features used.

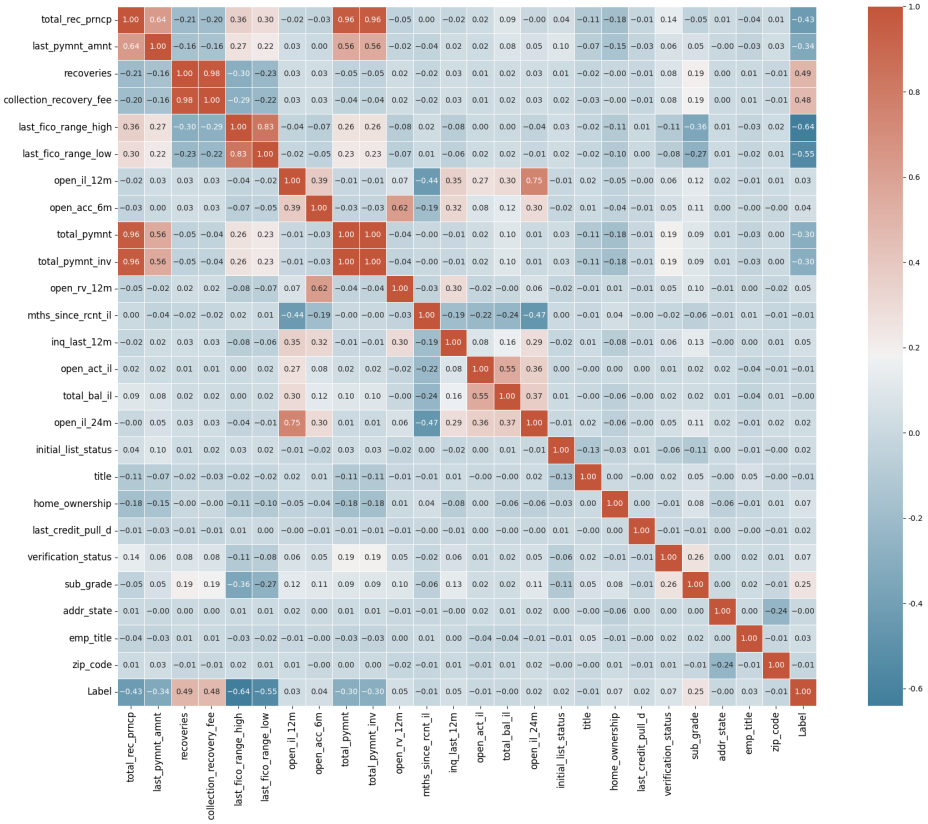


Figure 4. Correlation matrix on the first 25 highest mutual information features.

3.2. Model Creations and Evaluations

An overview of the processes in this work is depicted in Figure 5. The raw dataset was explored for characteristics such as data types and missing values. Subsequently, the data were preprocessed to handle missing values. The dataset was then separated into training and testing sets. Two data splitting protocols were experimented with, i.e., hold-out cross-validation with a 70:30 ratio of training and testing sets and 4-fold cross-validation. Next, the training data were prepared in four versions based on imbalanced data handling methods, including original (no sampling), over-sampling, under-sampling, and combined sampling training data. Each training dataset version was used to create three models using logistic regression, random forest, and gradient boosting approaches. The testing dataset was employed to evaluate model performance by calculating *Accuracy*, *Precision*, *Recall*, *F1 score*, and Matthews Correlation Coefficient (*MCC*). In the context of imbalanced data, where one class may dominate the others, using macro-averaging for *Precision*, *Recall*, *F1 score*, and *MCC* can provide a more balanced evaluation across different classes. Then, the confusion matrix was displayed, which is a tabular representation commonly employed to assess the effectiveness of a classification algorithm. This matrix provides a concise overview of the model's performance by detailing the distribution of predicted and actual class labels (Figure 6). Denote that green and red cells stand for the number of correctly predicted and wrongly predicted samples, respectively. *TP* and *TN* are the sample numbers of correctly classified to positive and negative classes, respectively, while *FP* and *FN* are the sample numbers of wrongly classified to positive and negative classes, respectively. Subsequently, key performance metrics such as *Accuracy*, *Precision*, *Recall*, *F1 score*, and *MCC* were computed as follows:

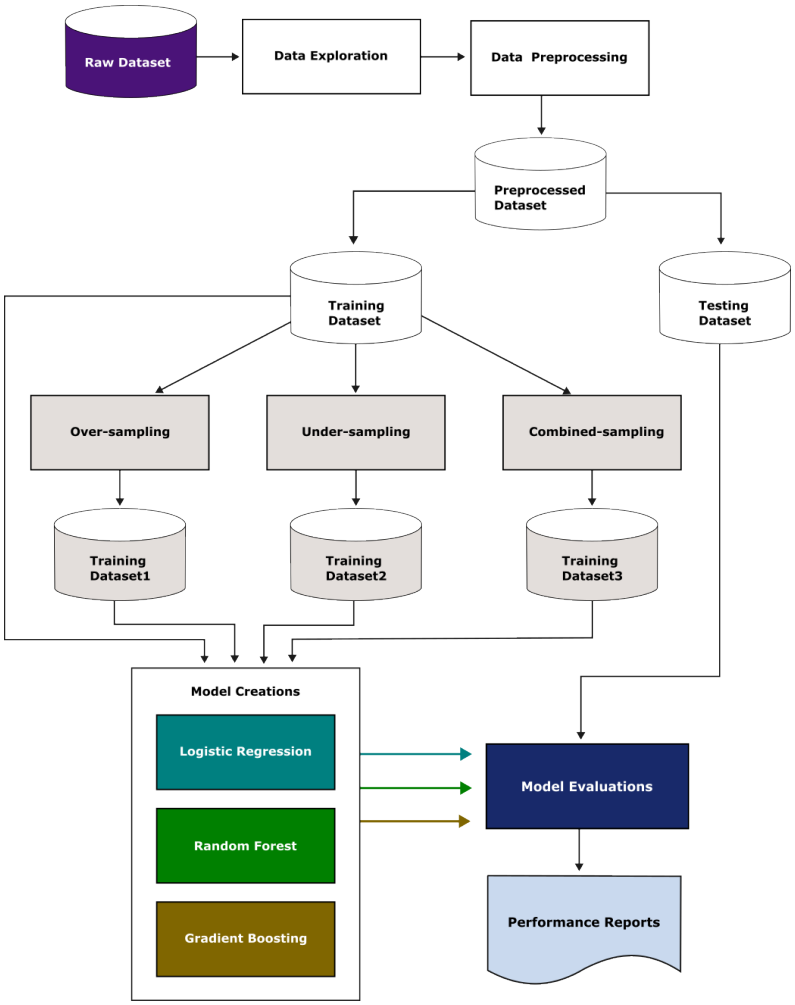


Figure 5. Overview of proposed methodology.

Confusion Matrix:

	Actual Positive	*Actual Negative*
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Figure 6. Two-by-two confusion matrix.

Accuracy is a fundamental metric that measures the overall correctness of a classification model by assessing the proportion of testing data that are correctly predicted out of the total testing data size.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Precision (Macro-Averaged) is a metric used to evaluate the *Precision* of a classification model when dealing with imbalanced datasets. In the context of macro-averaging, *Precision* is calculated individually for each class and then averaged across all classes.

$$Precision = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i} \tag{9}$$

where C is the number of classes and TP_i and FP_i are the true positives and false positives for class i.

Recall (Macro-Averaged) is a metric used to evaluate the *Recall* of a classification model in the context of imbalanced datasets. In macro averaging, *Recall* is calculated individually for each class and then averaged across all classes.

$$Recall = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FN_i} \tag{10}$$

where FN_i represents the false negatives for class i.

F1 score (Macro-Averaged) is a metric that combines both *Precision* and *Recall*, offering a balanced assessment of a model's performance on imbalanced datasets. In macro averaging, the *F1 Score* is calculated individually for each class and then averaged across all classes.

$$F1\ score = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{11}$$

where $Precision_i$ and $Recall_i$ are the *Precision* and *Recall* for class i.

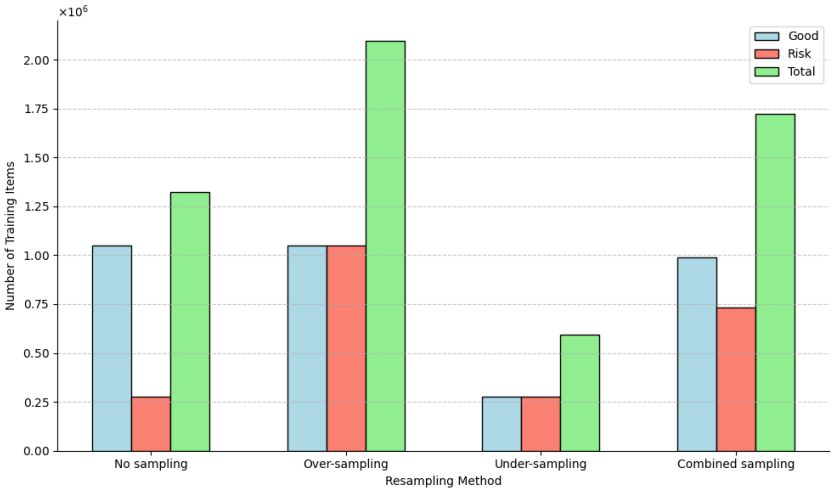
The Matthews Correlation Coefficient (*MCC*) is one of the metrics suitable for evaluating binary classification models, especially models that were trained by imbalanced datasets, because true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are all taken into account in its formula. The *MCC* is defined as:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{12}$$

The *MCC* value ranges between -1 and 1. The best and worst *MCC* values are 1 and -1, respectively. When the *MCC* value is 0, this means that the model performance is not greater than that of random guessing.

4. Results and Discussion

To solve an imbalanced data issue, four versions of training datasets, including data with no sampling, over-sampling, under-sampling, and combined sampling, were experimented with. The number of data samples in each training dataset is illustrated in Figure 7. Experiments with two methods of training and testing data splitting, 70:30 hold-out cross-validation and 4-fold cross-validation, were performed.



	# items 'Good'	*# items 'Risk'*	*Total*
No sampling	1,048,015	274,750	1,322,765
Over-sampling	1,048,015	1,048,015	2,096,030
Under-sampling	274,750	274,750	594,500
Combined sampling	989,477	734,581	1,724,058

Figure 7. Comparison of training data sizes on various sampling methods.

4.1. Hold-Out Cross-Validation with 70:30 Ratio of Training and Testing Sets

The confusion matrices of testing data prediction were yielded by logistic regression, random forest, and gradient boosting models, which were trained by four versions of training data as shown in Figure 8a–d. The five performance metrics, i.e., *Accuracy*, *Precision*, *Recall*, *F1 score*, and *MCC*, are shown in Table 3. The comparably highest effective performance, i.e., the first and second ranks across the five metrics, was yielded by the random forest model as well as the gradient boosting model trained by data with over-sampling. In detail, the gradient boosting model with the over-sampling technique showed slightly better results, i.e., performance values were 1 for all five measures, but this hold-out cross-validation experiment was performed one time due to convenience for a very large dataset, at first. Therefore, for a solid experimental conclusion, another 4-fold cross-validation experiment was also studied.

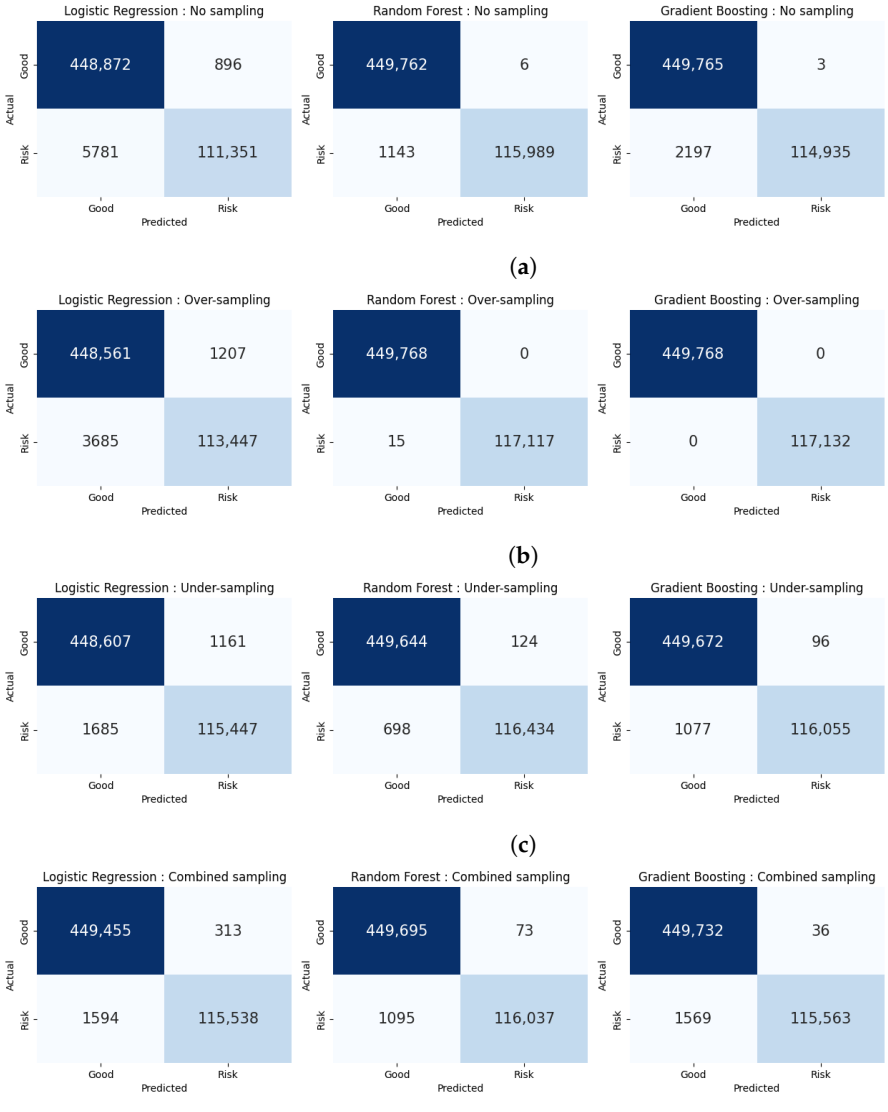


Figure 8. Confusion matrices of 70:30 hold-out cross-validation results. (**a**) No sampling testing data (original data). (**b**) Over-sampling testing data. (**c**) Under-sampling testing data. (**d**) Combined sampling testing data.

Table 3. The performance of three different machine learning techniques with various sampling approaches in the 70:30 hold-out cross-validation experiment. The superscript numbers in the brackets denote the performance ranking based on the evaluation measure in each column.

Imbalanced Data Handling Technique	Model	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*MCC*
No sampling (original data)	Logistic regression	0.9882 (12)	0.9920 (10)	0.9506 (12)	0.9709 (12)	0.9639 (12)
	Random forest	0.9979(4)	0.9999 (3)	0.9902 (6)	0.9951 (4)	0.9938 (4)
	Gradient boosting	0.9961 (9)	0.9999 (3)	0.9812 (10)	0.9905 (9)	0.9882 (9)
Over-sampling	Logistic regression	0.9914 (11)	0.9895 (12)	0.9685 (11)	0.9789 (11)	0.9736 (11)
	Random forest	0.9999 (2)	1.0000 (1)	0.9999 (2)	0.9999 (2)	0.9999 (2)
	Gradient boosting	1.0000 (1)	1.0000 (1)	1.0000 (1)	1.0000 (1)	1.0000 (1)
Under-sampling	Logistic regression	0.9950 (10)	0.9900 (11)	0.9856 (9)	0.9878 (10)	0.9847 (10)
	Random forest	0.9986 (3)	0.9989 (8)	0.9940 (3)	0.9965 (3)	0.9956 (3)
	Gradient boosting	0.9979 (4)	0.9992 (7)	0.9908 (4)	0.9950 (5)	0.9937 (5)
Combined sampling	Logistic regression	0.9966 (8)	0.9973 (9)	0.9864 (8)	0.9918 (8)	0.9897 (8)
	Random forest	0.9979 (4)	0.9994 (6)	0.9907 (5)	0.9950 (5)	0.9937 (5)
	Gradient boosting	0.9972 (7)	0.9997 (5)	0.9866 (7)	0.9931 (7)	0.9914 (7)

4.2. Four-Fold Cross-Validation

The average confusion matrices for 4-fold cross-validation results are illustrated in Figure 9a–d. The performance metrics for logistic regression, random forest, and gradient boosting are shown in Tables 4, 5, and 6, respectively. Three imbalanced data handling approaches, including over-sampling, under-sampling, and combined sampling, can improve the performance of models trained by logistic regression, random forest, and gradient boosting algorithms. Considering only the performance of logistic regression models, models with the combined sampling approach outperform the others. For random forest and gradient boosting models, when the under-sampling approach was employed, they both showed better model performance compared to the other sampling approaches. In general, from all 4-fold cross-validation results, gradient boosting models with the under-sampling method gave the superior performance. The additionally depicted comparisons of *MCC* and *F1 score* are shown in Figure 10 and Figure 11, respectively.

Overall result summation from both experiments of the two cross-validation methods indicates that the gradient boosting algorithm with an appropriate data solving technique for supervised model training offers the very impressive ability of resulting in models that correctly classify both "Good" and "Risk" instances.

Next, the feature selection method was applied, i.e., computing and ranking the mutual information (*MI*) values of each feature, in order to reasonably select important features of the smaller feature size k of the training set. So, the training data with the best imbalanced data handling technique for each model were further explored by preparing a smaller number of k features via their *MI* values to assess the trade-offs between the different important feature sizes and their impact on model performance. The features were ranked based on their computed values of mutual information. Three numbers of feature size, i.e., $k = 25$, 50, and 100, were experimented with. The results of logistic regression, random forest, and gradient boosting models on both 70:30 hold-out cross-validation and 4-fold cross-validation with three different feature sizes are shown in Figures 12, 13, and 14, respectively. Generally, the performance of all three supervised models was reduced slightly. For $k = 25$ and 50 important features as training data, gradient boosting models showed better results than logistic regression and random forest models. Focusing on $k = 50$ important features, random forest and gradient boosting models yielded five performance values, i.e., *Accuracy*, *Precision*, *Recall*, *F1 score*, and *MCC*, greater than 99%, whereas logistic regression models gave four performance values, excepting *MCC*, higher than 95%. For $k = 25$ important features, gradient boosting models still yielded *Accuracy*,

Precision, *Recall*, and *F1 score* values not less than 99%, but *MCC* values reduced to around 97.5%. These show that when the number of features was reduced by half ($k = 50$), the performance values were reduced by only less than 1%. Although the number of features was approximately reduced by 75% ($k = 25$), the performance values were reduced by only less than 1–2%. Apart from that, the performance of gradient boosting models using $k = 100$ important features was better than that of the others on both 70:30 hold-out cross-validation and 4-fold cross-validation experiments.

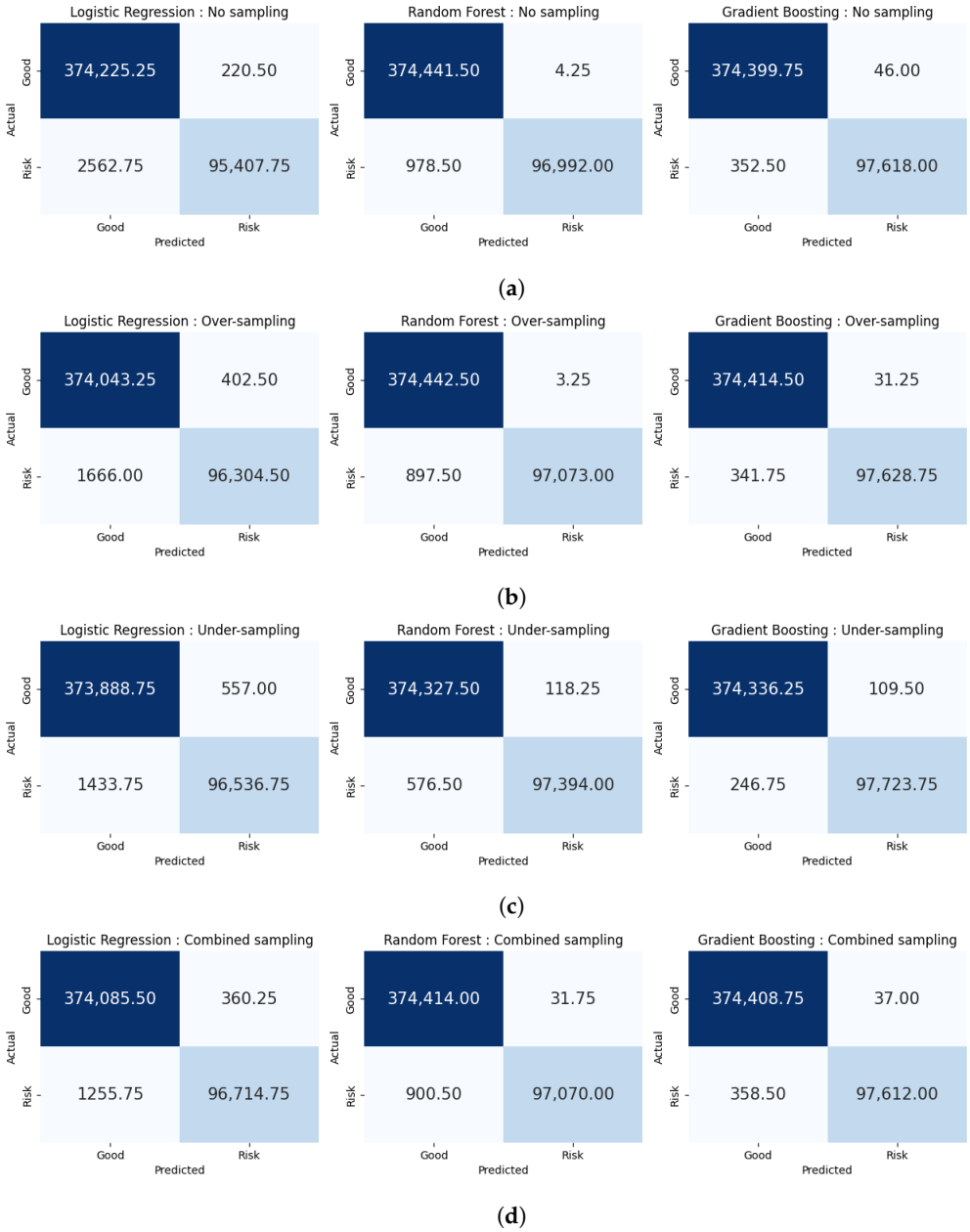


Figure 9. Average confusion matrices of four-fold cross-validation results. (**a**) No sampling testing data (original data). (**b**) Over-sampling testing data. (**c**) Under-sampling testing data. (**d**) Combined sampling testing data.

Table 4. Performance metrics for logistic regression with different sampling approaches in 4-fold cross-validation (4-fold cv) experiments.

Method	4-Fold cv	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*MCC*
Logistic regression: No sampling	Fold 1	0.993637	0.993660	0.993637	0.993607	0.980512
	Fold 2	0.994283	0.994302	0.994283	0.994258	0.982578
	Fold 3	0.993988	0.994015	0.993988	0.993960	0.981772
	Fold 4	0.994526	0.994546	0.994526	0.994503	0.983267
	Average	0.994108	0.994131	0.994108	0.994082	0.982032
Logistic regression: Over-sampling	Fold 1	0.995970	0.995972	0.995970	0.995960	0.987666
	Fold 2	0.996202	0.996201	0.996202	0.996196	0.988435
	Fold 3	0.995349	0.995352	0.995349	0.995337	0.985900
	Fold 4	0.994964	0.994964	0.994964	0.994951	0.984602
	Average	0.995621	0.995622	0.995621	0.995611	0.986651
Logistic regression: Under-sampling	Fold 1	0.996188	0.996185	0.996188	0.996182	0.988336
	Fold 2	0.995838	0.995836	0.995838	0.995831	0.987324
	Fold 3	0.995170	0.995165	0.995170	0.995161	0.985355
	Fold 4	0.995948	0.995945	0.995948	0.995943	0.987621
	Average	0.995786	0.995783	0.995786	0.995779	0.987159
Logistic regression: Combined sampling	Fold 1	0.997049	0.997050	0.997049	0.997044	0.990975
	Fold 2	0.996418	0.996421	0.996418	0.996411	0.989094
	Fold 3	0.996833	0.996833	0.996833	0.996828	0.990406
	Fold 4	0.996016	0.996012	0.996016	0.996011	0.987829
	Average	0.996579	0.996579	0.996579	0.996573	0.989576

Table 5. Performance metrics for random forest with different sampling approaches in 4-fold cross-validation (4-fold cv) experiments.

Method	4-Fold cv	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*MCC*
Random forest: No sampling	Fold 1	0.997993	0.997998	0.997993	0.997990	0.993868
	Fold 2	0.997949	0.997954	0.997949	0.997945	0.993762
	Fold 3	0.997818	0.997823	0.997818	0.997813	0.993395
	Fold 4	0.997919	0.997924	0.997919	0.997915	0.993651
	Average	0.997920	0.997925	0.997920	0.997916	0.993669
Random forest: Over-sampling	Fold 1	0.998114	0.998118	0.998114	0.998111	0.994237
	Fold 2	0.998167	0.998171	0.998167	0.998164	0.994425
	Fold 3	0.998050	0.998055	0.998050	0.998047	0.994100
	Fold 4	0.998042	0.998047	0.998042	0.998039	0.994026
	Average	0.998093	0.998098	0.998093	0.998090	0.994197
Random forest: Under-sampling	Fold 1	0.998567	0.998568	0.998567	0.998566	0.995621
	Fold 2	0.998548	0.998548	0.998548	0.998547	0.995583
	Fold 3	0.998478	0.998478	0.998478	0.998477	0.995393
	Fold 4	0.998525	0.998525	0.998525	0.998523	0.995497
	Average	0.998529	0.998530	0.998529	0.998528	0.995523
Random forest: Combined sampling	Fold 1	0.998076	0.998079	0.998076	0.998073	0.994120
	Fold 2	0.997987	0.997990	0.997987	0.997984	0.993877
	Fold 3	0.998059	0.998063	0.998059	0.998056	0.994125
	Fold 4	0.997985	0.997989	0.997985	0.997981	0.993851
	Average	0.998027	0.998030	0.998027	0.998023	0.993993

Table 6. Performance metrics for gradient boosting with different sampling approaches in 4-fold cross-validation (4-fold cv) experiments.

Method	4-Fold cv	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*MCC*
Gradient boosting: No sampling	Fold 1	0.999172	0.999173	0.999172	0.999172	0.997471
	Fold 2	0.999130	0.999130	0.999130	0.999130	0.997355
	Fold 3	0.999217	0.999217	0.999217	0.999216	0.997630
	Fold 4	0.999107	0.999107	0.999107	0.999106	0.997275
	Average	0.999156	0.999157	0.999156	0.999156	0.997433
Gradient boosting: Over-sampling	Fold 1	0.999280	0.999281	0.999280	0.999280	0.997801
	Fold 2	0.999179	0.999179	0.999179	0.999178	0.997503
	Fold 3	0.999208	0.999209	0.999208	0.999208	0.997605
	Fold 4	0.999174	0.999175	0.999174	0.999174	0.997482
	Average	0.999210	0.999211	0.999210	0.999210	0.997598
Gradient boosting: Under-sampling	Fold 1	0.999285	0.999284	0.999285	0.999284	0.997814
	Fold 2	0.999276	0.999276	0.999276	0.999276	0.997799
	Fold 3	0.999257	0.999257	0.999257	0.999257	0.997752
	Fold 4	0.999166	0.999166	0.999166	0.999166	0.997456
	Average	0.999246	0.999246	0.999246	0.999246	0.997705
Gradient boosting: Combined sampling	Fold 1	0.999164	0.999164	0.999164	0.999163	0.997446
	Fold 2	0.999177	0.999177	0.999177	0.999176	0.997496
	Fold 3	0.99913	0.99913	0.99913	0.999129	0.997367
	Fold 4	0.999181	0.999181	0.999181	0.99918	0.997501
	Average	0.999163	0.999163	0.999163	0.999162	0.997453

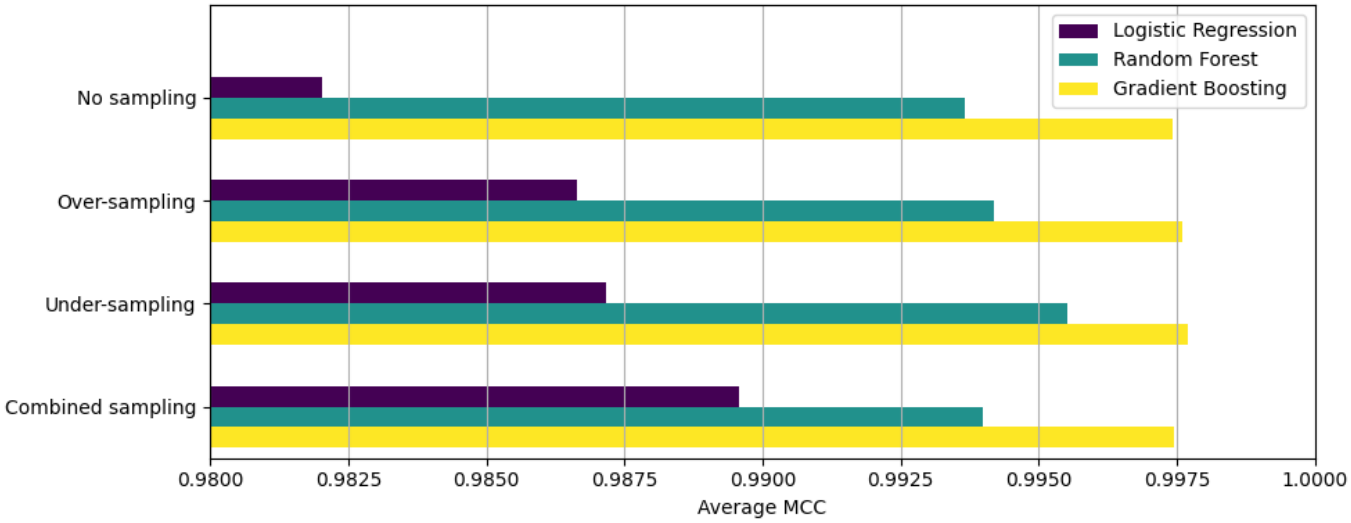


Figure 10. Average *MCC* comparison for different sampling methods.

In order to additionally display our results compared with previous research, the performance comparison of the proposed methods with other existing works on various versions of LendingClub data is shown in Figure 15. Based on *Accuracy*, the proposed methods outperform the others.

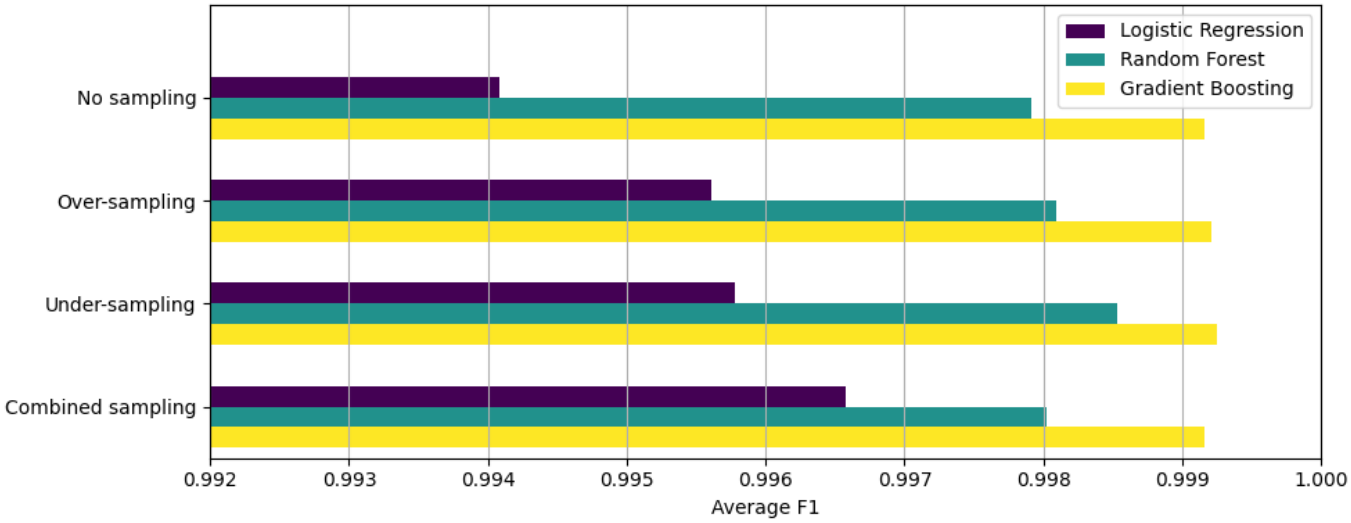


Figure 11. Average *F*1 *score* comparison for different sampling methods.

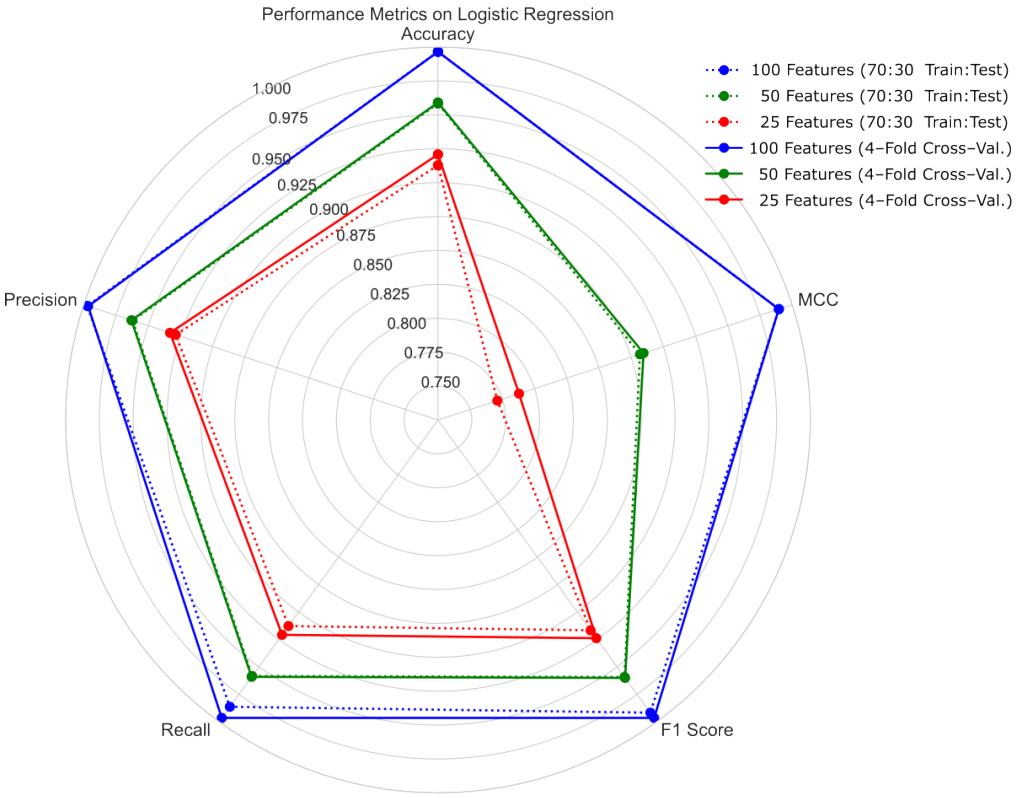


Figure 12. Logistic regression model performance on five metrics for k different numbers of features, i.e., $k = 25$, 50, and 100.

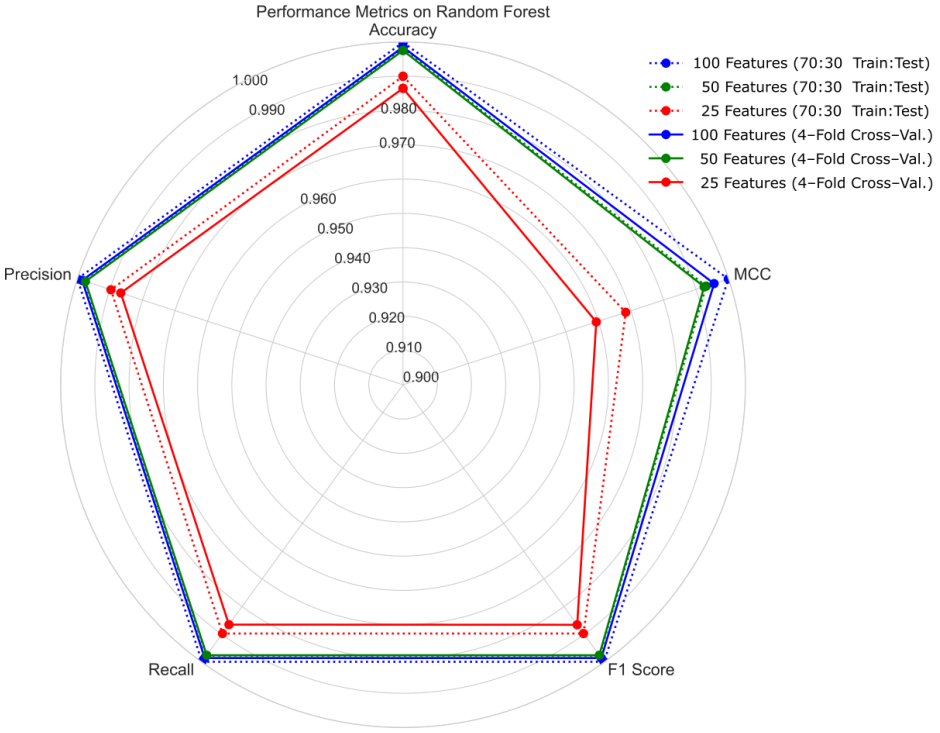


Figure 13. Random forest model performance on five metrics for k different numbers of features, i.e., $k = 25$, 50, and 100.

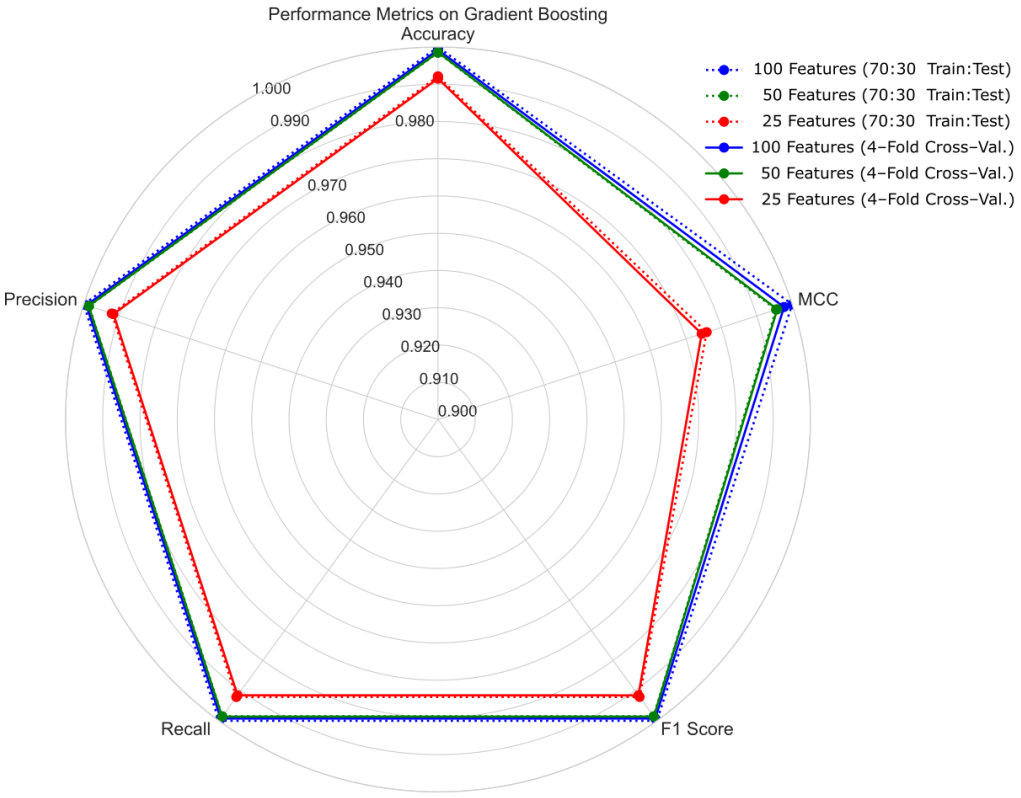


Figure 14. Gradient boosting model performance on five metrics for k different numbers of features, i.e., $k = 25$, 50, and 100.

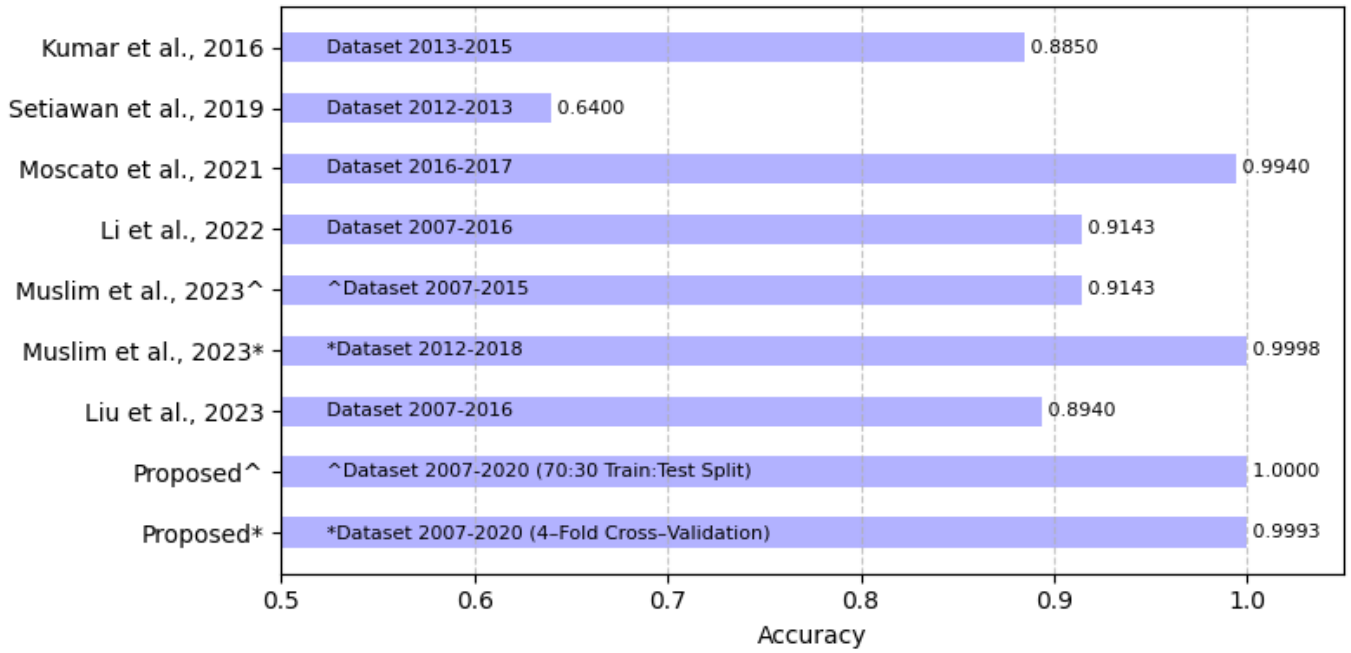


Figure 15. *Accuracy* of the proposed method compared with existing works on various versions of LendingClub data. Denote that ^ and * symbols stand for the different dataset or experiment in the same work.

5. Conclusions and Future Work

This study provided a very efficient solution to the problem of credit risk prediction. To investigate the improved predictive model results that could be better than those from previous works, three popular machine learning methods, including logistic regression, random forest, and gradient boosting, were employed. Additionally, the imbalanced data problem was resolved by experimenting with various sampling strategies: under-sampling, over-sampling, and combined sampling. Based on our best model performance outcomes, the over-sampling as well as under-sampling methods robustly manage class-imbalanced data, especially when the training model uses the gradient boosting method. In addition, the feature numbers of the data were reduced by selecting only important features for the training set according to their ranks computed by mutual information. Another experiment was performed using two reduced feature sets, the half size as well as the one-fourth size of its original feature size. The resulting model performance was just barely decreased. Remarkably, both random forest and gradient boosting models created by the reduced feature sets with the half size showed impressive Accuracy values, higher than 99%.

This comprehensive analysis enhances better understanding of credit risk prediction using a supervised learning method combined with various imbalanced data solving strategies. Furthermore, the importance of features based on mutual information was addressed in order to increase model performance with the smaller feature size of training data. Our proposed method and results offer a simple way to select important features with the reduced size by ranking the mutual information values of each feature. In spite of this method not providing the most optimal size with the best performance, it can apply to other large credit risk data with different feature sets. This approach does not significantly decrease performance, but there might be better methods available. In future work, it may be beneficial to further investigate parameter optimization, particularly in handling imbalanced data, and explore alternative feature selection methods beyond mutual information, such as correlation and symmetrical uncertainty, to improve model performance. In addition, ensemble techniques could offer performance improvement of those small feature sizes. Apart from that, real-time data streams and dynamic model updating may increase the adaptability of credit risk prediction systems.

Author Contributions: Conceptualization, N.W. and S.T.; methodology, N.W. and S.T.; validation, N.W. and S.T.; formal analysis, N.W. and P.W.; investigation, N.W., P.W. and S.T.; data curation, N.W. and P.W.; writing—original draft preparation, N.W. and S.T.; writing—review and editing, N.W., S.J., S.S. and S.T.; visualization, N.W., P.W. and S.T.; supervision, S.J. and S.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.

Informed Consent Statement: Not applicable.

Data Availability Statement: The used data are publicly available at https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1 (accessed on 17 January 2024).

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

References

1. Noriega, J.P.; Rivera, L.A.; Herrera, J.A. Machine Learning for Credit Risk Prediction: A Systematic Literature Review. *Data* **2023**, *8*, 169. [CrossRef]
2. Gjeçi, A.; Marinč, M.; Rant, V. Non-performing loans and bank lending behaviour. *Risk Manag.* **2023**, *25*, 7. [CrossRef]
3. Liu, H.; Qiao, H.; Wang, S.; Li, Y. Platform Competition in Peer-to-Peer Lending Considering Risk Control Ability. *Eur. J. Oper. Res.* **2019**, *274*, 280–290. [CrossRef]
4. Sulastri, R.; Janssen, M. Challenges in Designing an Inclusive Peer-to-Peer (P2P) Lending System. In Proceedings of the 24th Annual International Conference on Digital Government Research, DGO '23, New York, NY, USA, 11–14 July 2023; pp. 55–65. [CrossRef]
5. Ko, P.C.; Lin, P.C.; Do, H.T.; Huang, Y.F. P2P Lending Default Prediction Based on AI and Statistical Models. *Entropy* **2022**, *24*, 801. [CrossRef]
6. Kurniawan, R. Examination of the Factors Contributing To Financial Technology Adoption in Indonesia using Technology Acceptance Model: Case Study of Peer to Peer Lending Service Platform. In Proceedings of the 2019 International Conference on Information Management and Technology (ICIMTech), Denpasar, Indonesia, 19–20 August 2019; Volume 1, pp. 432–437. [CrossRef]
7. Wang, Q.; Xiong, X.; Zheng, Z. Platform Characteristics and Online Peer-to-Peer Lending: Evidence from China. *Financ. Res. Lett.* **2021**, *38*, 101511. [CrossRef]
8. Ma, Z.; Hou, W.; Zhang, D. A credit risk assessment model of borrowers in P2P lending based on BP neural network. *PLoS ONE* **2021**, *16*, e0255216. [CrossRef]
9. Moscato, V.; Picariello, A.; Sperlí, G. A benchmark of machine learning approaches for credit score prediction. *Expert Syst. Appl.* **2021**, *165*, 113986. [CrossRef]
10. Liu, W.; Fan, H.; Xia, M. Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Syst. Appl.* **2022**, *189*, 116034. [CrossRef]
11. Kriebel, J.; Stitz, L. Credit default prediction from user-generated text in peer-to-peer lending using deep learning. *Eur. J. Oper. Res.* **2022**, *302*, 309–323. [CrossRef]
12. Uddin, N.; Uddin Ahamed, M.K.; Uddin, M.A.; Islam, M.M.; Talukder, M.A.; Aryal, S. An ensemble machine learning based bank loan approval predictions system with a smart application. *Int. J. Cogn. Comput. Eng.* **2023**, *4*, 327–339. [CrossRef]
13. Yin, W.; Kirkulak-Uludag, B.; Zhu, D.; Zhou, Z. Stacking ensemble method for personal credit risk assessment in Peer-to-Peer lending. *Appl. Soft Comput.* **2023**, *142*, 110302. [CrossRef]
14. Muslim, M.A.; Nikmah, T.L.; Pertiwi, D.A.A.; Dasril, Y. New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning. *Intell. Syst. Appl.* **2023**, *18*, 200204. [CrossRef]
15. Niu, K.; Zhang, Z.; Liu, Y.; Li, R. Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Inf. Sci.* **2020**, *536*, 120–134. [CrossRef]
16. Li, X.; Ergu, D.; Zhang, D.; Qiu, D.; Cai, Y.; Ma, B. Prediction of loan default based on multi-model fusion. *Procedia Comput. Sci.* **2022**, *199*, 757–764. [CrossRef]
17. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 1–6 June 2008; pp. 1322–1328. [CrossRef]
18. Chen, Y.R.; Leu, J.S.; Huang, S.A.; Wang, J.T.; Takada, J.I. Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets. *IEEE Access* **2021**, *9*, 73103–73109. [CrossRef]
19. Kumar, V.L.; Natarajan, S.; Keerthana, S.; Chinmayi, K.M.; Lakshmi, N. Credit Risk Analysis in Peer-to-Peer Lending System. In Proceedings of the 2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA), Singapore, 28–30 September 2016; pp. 193–196. [CrossRef]

20. Setiawan, N. A Comparison of Prediction Methods for Credit Default on Peer to Peer Lending using Machine Learning. *Procedia Comput. Sci.* **2019**, *157*, 38–45. [CrossRef]
21. Liu, Z.; Zhang, Z.; Yang, H.; Wang, G.; Xu, Z. An innovative model fusion algorithm to improve the recall rate of peer-to-peer lending default customers. *Intell. Syst. Appl.* **2023**, *20*, 200272. [CrossRef]
22. Ziemba, P.; Becker, J.; Becker, A.; Radomska-Zalas, A.; Pawluk, M.; Wierzba, D. Credit Decision Support Based on Real Set of Cash Loans Using Integrated Machine Learning Algorithms. *Electronics* **2021**, *10*, 2099. [CrossRef]
23. Dong, H.; Liu, R.; Tham, A.W. Accuracy Comparison between Five Machine Learning Algorithms for Financial Risk Evaluation. *J. Risk Financ. Manag.* **2024**, *17*, 50. [CrossRef]
24. Stoltzfus, J.C. Logistic regression: A brief primer. *Acad. Emerg. Med.* **2011**, *18*, 1099–1104. [CrossRef]
25. Manglani, R.; Bokhare, A. Logistic Regression Model for Loan Prediction: A Machine Learning Approach. In Proceedings of the 2021 Emerging Trends in Industry 4.0 (ETI 4.0), Raigarh, India, 19–21 May 2021; pp. 1–6. [CrossRef]
26. Kadam, E.; Gupta, A.; Jagtap, S.; Dubey, I.; Tawde, G. Loan Approval Prediction System using Logistic Regression and CIBIL Score. In Proceedings of the 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 7–9 August 2023; pp. 1317–1321. [CrossRef]
27. Zhu, X.; Chu, Q.; Song, X.; Hu, P.; Peng, L. Explainable prediction of loan default based on machine learning models. *Data Sci. Manag.* **2023**, *6*, 123–133. [CrossRef]
28. Lin, M.; Chen, J. Research on Credit Big Data Algorithm Based on Logistic Regression. *Procedia Comput. Sci.* **2023**, *228*, 511–518. [CrossRef]
29. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
30. Zhu, L.; Qiu, D.; Ergu, D.; Ying, C.; Liu, K. A study on predicting loan default based on the random forest algorithm. *Procedia Comput. Sci.* **2019**, *162*, 503–513. [CrossRef]
31. Rao, C.; Liu, M.; Goh, M.; Wen, J. 2-stage modified random forest model for credit risk assessment of P2P network lending to "Three Rurals" borrowers. *Appl. Soft Comput.* **2020**, *95*, 106570. [CrossRef]
32. Reddy, C.S.; Siddiq, A.S.; Jayapandian, N. Machine Learning based Loan Eligibility Prediction using Random Forest Model. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 12–14 November 2022; pp. 1073–1079. [CrossRef]
33. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
34. Zhou, L.; Fujita, H.; Ding, H.; Ma, R. Credit risk modeling on data with two timestamps in peer-to-peer lending by gradient boosting. *Appl. Soft Comput.* **2021**, *110*, 107672. [CrossRef]
35. Zhu, X.; Chen, J. Risk Prediction of P2P Credit Loans Overdue Based on Gradient Boosting Machine Model. In Proceedings of the 2021 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 29–31 July 2021; pp. 212–216. [CrossRef]
36. Miaojun Bai, Y.Z.; Shen, Y. Gradient boosting survival tree with applications in credit scoring. *J. Oper. Res. Soc.* **2022**, *73*, 39–55. [CrossRef]
37. Qian, H.; Wang, B.; Yuan, M.; Gao, S.; Song, Y. Financial distress prediction using a corrected feature selection measure and gradient boosted decision tree. *Expert Syst. Appl.* **2022**, *190*, 116202. [CrossRef]
38. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority over-Sampling Technique. *J. Artif. Int. Res.* **2002**, *16*, 321–357. [CrossRef]
39. Bach, M.; Werner, A.; Palt, M. The Proposal of Undersampling Method for Learning from Imbalanced Datasets. *Procedia Comput. Sci.* **2019**, *159*, 125–134. [CrossRef]
40. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
41. Ethon0426. Lending Club 2007–2020Q3. Available online: https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1 (accessed on 17 January 2024).