Data Science in Finance and Economics

http://www.aimspress.com/journal/dsfe

DSFE, 3(4): 354–379.
DOI:10.3934/DSFE.2023021
Received: 16 June 2023
Revised: 14 August 2023
Accepted: 5 September 2023
Published: 13 October 2023

Research article

Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models

Lindani Dube[1,2,*] **and Tanja Verster**[1,2]

[1] Centre for Business Mathematics & Informatics, North West University, Potchefstroom, 2531, South Africa
[2] National Institute for Theoretical and Computational Sciences (NITheCS), South Africa

* **Correspondence:** Email: 45026602@nwu.ac.za.

Abstract: In the realm of machine learning, where data-driven insights guide decision-making, addressing the challenges posed by class imbalance in datasets has emerged as a crucial concern. The effectiveness of classification algorithms hinges not only on their intrinsic capabilities but also on their adaptability to uneven class distributions, a common issue encountered across diverse domains. This study delves into the intricate interplay between varying class imbalance levels and the performance of ten distinct classification models, unravelling the critical impact of this imbalance on the landscape of predictive analytics. Results showed that random forest (RF) and decision tree (DT) models outperformed others, exhibiting robustness to class imbalance. Logistic regression (LR), stochastic gradient descent classifier (SGDC) and naïve Bayes (NB) models struggled with imbalanced datasets. Adaptive boosting (ADA), gradient boosting (GB), extreme gradient boosting (XGB), light gradient boosting machine (LGBM), and k-nearest neighbour (kNN) models improved with balanced data. Adaptive synthetic sampling (ADASYN) yielded more reliable predictions than the under-sampling (UNDER) technique. This study provides insights for practitioners and researchers dealing with imbalanced datasets, guiding model selection and data balancing techniques. RF and DT models demonstrate superior performance, while LR, SGDC and NB models have limitations. By leveraging the strengths of RF and DT models and addressing class imbalance, classification performance in imbalanced datasets can be enhanced. This study enriches credit risk modelling literature by revealing how class imbalance impacts default probability estimation. The research deepens our understanding of class imbalance's critical role in predictive analytics. Serving as a roadmap for practitioners and researchers dealing with imbalanced data, the findings guide model selection and data balancing strategies, enhancing classification performance despite class imbalance.

Keywords: credit risk; class imbalance; default prediction; supervised learning; classification

Abbreviations: ADA: adaBoost; ADASYN: Adaptive synthetic sampling; ANN: Artificial neural network; AUC: Area under the curve; AUROC: Area under the ROC curve; DT: Decision tree; GB: Gradient boosting; kNN: k-Nearest neighbors; LGBM: Light gradient boosting machine; LGBM: LightGBM; LR: Logistic regression; MCC: Mathews' correlation coefficient; NB: Naïve Bayesian; RF: Random forest; ROC: Receiver operating characteristic; SGDC: Stochastic descent gradient; UNDER: Under-sampling; XGB: Extreme gradient boosting

1. Introduction

The accurate estimation of credit default risk plays a critical role in the financial industry, enabling banks and lenders to make informed decisions regarding loan approvals, credit limits, and pricing. Traditionally, credit risk assessment has relied on statistical models and expert judgment. However, with the advancements in machine learning techniques and the availability of large-scale credit data, there has been growing interest in utilizing these methods for credit risk prediction.

One significant challenge in credit risk modeling is dealing with class imbalance, where the number of default instances is significantly smaller than the non-default instances. This imbalance can lead to biased models and poor predictive performance (Thabtah et al., 2020), as most machine learning algorithms are designed to maximize overall accuracy and may struggle to accurately classify the minority class. Consequently, misclassification errors related to defaults can have severe financial implications. The consequences of misclassifying credit defaults can be significant. Khemakhem and Boujelbene (2018) argued that false negatives (predicting a non-default when it is actually a default) can expose lenders to potential losses and increased credit risk. On the other hand, false positives (predicting a default when it is actually a non-default) can result in unnecessary restrictions on credit access for borrowers and potential loss of business for lenders. The impact of class imbalance on credit risk prediction has gained attention in recent research. It is crucial to understand the behaviour and limitations of machine learning algorithms (Leo et al., 2019) under imbalanced conditions to develop robust models that effectively capture the risk associated with credit defaults. This understanding can help financial institutions enhance their decision-making processes, mitigate potential losses, and ensure fair access to credit for borrowers.

The objective of this research article is to investigate the effect of class imbalance on the estimation of default probabilities using machine learning algorithms. We aim to analyse the performance of various algorithms under different class distributions and evaluate their effectiveness in capturing default events accurately. Additionally, we will explore different techniques for addressing class imbalance, such as over-sampling, particularly ADASYN sampling and under-sampling to assess their impact on model performance. To achieve our research objectives, we will utilize a dataset of credit information, including borrower characteristics, historical payment behaviour and other relevant factors. We will compare the performance of different machine learning algorithms, such as logistic regression, random forest, gradient boosting and decision tree in estimating default probabilities. Additionally, we will evaluate the Mathews' correlation coefficient (MCC) and F1-scores of the models using metrics like the confusion matrix and area under the receiver operating characteristic curve (AUC-ROC).

The findings of this study will contribute to the existing literature on credit risk modeling by providing insights into the effect of class imbalance on default probability estimation. This systematic investigation provides a deeper understanding of the critical impact of class imbalance on predictive analytics. By evaluating ten distinct classification models using rigorous evaluation metrics such as the area under the ROC curve, Mathews' correlation coefficient (MCC) and F1-scores, this research offers empirical insights into the strengths and weaknesses of these models in the context of imbalanced datasets. With the evidence-based assessment of various classification models and balancing techniques, this study serves as a valuable guide for practitioners and researchers dealing with imbalanced datasets. The findings offer clear directions for selecting appropriate models and applying tailored data balancing strategies, ultimately enhancing classification performance in the presence of class imbalance. It will also shed light on the strengths and limitations of different machine learning algorithms in the context of imbalanced credit data. The results will be valuable for financial institutions and regulators in developing more accurate and reliable credit risk models, enhancing credit decision-making processes, and promoting fair access to credit.

In summary, this research article aims to bridge the gap in understanding the impact of class imbalance on credit risk prediction. By investigating the behavior of machine learning algorithms under imbalanced conditions and exploring techniques to address the imbalance, we seek to improve the accuracy and reliability of default probability estimation. The outcomes of this study will contribute to the advancement of credit risk modeling and have practical implications for the financial industry.

2. Proposed architecture

One of the major challenges when building default prediction models, is the issue of imbalanced data. Class imbalance occurs whenever one majority class's training samples vastly outnumber those of the other minority class. Research has revealed that algorithms trained on an imbalanced dataset tend to suffer from a prediction biasedness and this often results in poor performance in the minority class. This paper will be exploring the results across ADASYN sampling and under-sampling. Figure 1 below outlines the proposed methodology that is adopted in this paper.

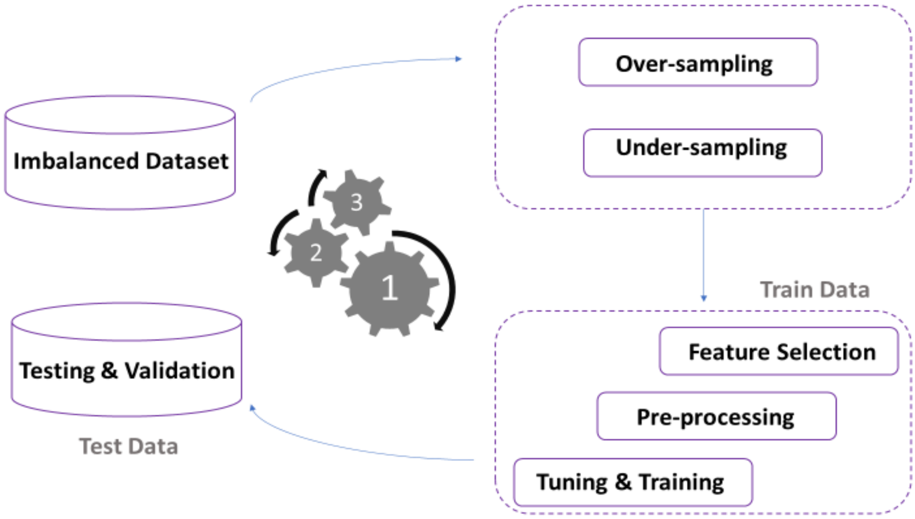


Figure 1. Adopted proposed approach.

2.1. Pre-processing

The initial and fundamental step in dealing with any sort of data is to first clean the data thoroughly and make sure that it makes sense (e.g. no nuisance entries). Given the nature of our dataset (see Section 4.), the first step was to standardise all the explanatory variables min/max scaling. The motivation for this was that whenever training an algorithm with a variable such as salary, which can range from 10000 to 665000 while a variable such as credit utilization is captured as a ratio, the classifier might assume salary is more important than credit utilization. Min/max scaling (Patro and Sahu, 2015), which was done in R-studio, scales all the numerical variables to range between 0 and 1.

2.1.1. K-fold cross-validation

Cross-validation consists of developing models explaining relationships among variables based on a subset of data, called the training data, and then testing the model on the testing data. K-fold cross-validation splits the data (K) times into training and testing data and then identifies the model that performs best in the aggregate (Cawley and Talbot, 2010). An even more refined approach reserves another 20% of the data for a separate testing stage, which is not part of model development and testing, but is used instead for out-of-sample testing of the model obtained on the basis of the training and testing datasets. In cross-validation, the training and testing data are separated and the testing data are used only when a best-fitting model has emerged (Anguita et al., 2012). In our case, we have used 5-fold cross validation on the 80% (in-sample) of the data and used the remaining 20% (out-of-sample) of the data for model testing in order to get more accurate results.

2.1.2. Missingness

There are two different strategies for handling missing data (Han et al., 2012). The first strategy is to simply ignore missing values and the second strategy is to consider imputation of missing values.

Omit missing values

The serious problem with omitting observations with missing values is that it reduces the dataset size. This is appropriate when your dataset has a small amount of missing values. There are two general approaches for ignoring missing data: listwise deletion (case deletion or complete case analysis) and pairwise deletion (available case analysis) approach. Complete case analysis approach excludes all observations with missing values for any variable of interest. This approach limits the analysis to those observations for which all values are observed which often results in biased estimate and loss of precision (Schafer and Graham, 2002). In pairwise deletion, we perform analysis with all cases in which the variables of interest are present. It does not exclude the entire unit but uses as much data as possible from every unit. The advantage of this method is that it keeps maximum available data for analysis even when some of its variables have missing values. The disadvantage of this method is that it uses different sample size for different variables (Schafer and Graham, 2002). The sample size for each individual analysis is higher than the complete case analysis.

Impute missing values

Missing data imputation is a procedure that replaces missing values with some plausible values (Rubin, 1976). The various imputation techniques aim to provide accurate estimation of population parameters so that power of data mining and data analysis techniques is not reduced. Optimal treatment to be given to the missing data depends on amount of missing data. Although there is no general rule on what percentage of missing data is bad, it is always better to do comparison of results before and after imputation.

In this paper we have adopted the median imputation method for handling missingness. Median imputation is used for numerical data and our dataset was of this composition. Median imputation is a method for handling missing values by replacing missing values in a dataset with the median value of the non-missing observations of the same variable. This method assumes that the missing data are missing at random and that the median is a good representation of the central tendency of the data. The median is calculated by first ordering the nonmissing observations of a variable and then identifying the middle value or the average of the two middle values, depending on whether the number of observations is odd or even. This imputed median value is then used to replace all the missing values of that variable.

Initially, we present a comprehensive review of the relevant literature pertaining to the research topic. Subsequently, we expound on the adopted methodology employed in this study. The models utilized are discussed and references to additional research papers are provided for supplementary understanding. Additionally, we furnish a detailed account of the analyzed dataset, coupled with an exploratory data analysis. Finally, an extensive analysis of the results of our machine learning algorithms is presented, along with recommendations for future research.

2.2. Hyper-parameter tuning

Hyper-parameter tuning in machine learning is the process of selecting the optimal values for hyper-parameters, which are parameters set by the user that control the behaviour of the learning algorithm. The goal is to find the hyper-parameters that result in the best balance of model complexity and performance. This process can be time-consuming and computationally expensive but it is an important step in developing accurate and reliable machine learning models. If default hyper-parameters were used for the models in R, the opportunity to fine-tune the models and achieve optimal performance for the specific task or dataset may have been missed.

In order to develop and evaluate the performance of our machine learning models, we utilized default hyper-parameters in the R programming language. While this approach may not have allowed for the fine-tuning of hyper-parameters to achieve optimal performance for our specific task and dataset, it allowed us to establish a baseline level of performance and compare the relative performance of different models. This information was valuable in guiding our model selection process and identifying areas for future improvement.

2.3. ADASYN sampling and under-sampling

ADASYN and under-sampling are techniques used in machine learning to address class imbalance in datasets. Under-sampling involves reducing the number of instances in the majority class to create a more balanced dataset, allowing the classifier to learn effectively from both classes. This can be achieved through methods such as random under-sampling or removing instances close to the decision boundary. On the other hand, ADASYN sampling takes a more adaptive approach by generating

Figure 2. Machine learning in statistics.

synthetic examples for the minority class, particularly focusing on difficult-to-learn instances. By augmenting the minority class, ADASYN aims to improve the classifier's performance and achieve better predictive accuracy. While under-sampling can lead to loss of information from the majority class, ADASYN sampling leverages the distribution of the minority class to generate synthetic samples and overcome the imbalance issue. Both techniques aim to enhance the learning process in imbalanced datasets, but they adopt different strategies to achieve a balanced representation of the classes.

3. Machine learning algorithms

In data analytics, (Breeden, 2021) machine learning is a set of computational methods which use experience to improve performance or to make accurate predictions. Here, the word ”experience” refers to past information available to the machine learning technique, classifier. In particular, data quality and data size are at the core of machine learning and, since the success of a learning algorithm depends on the data used, machine learning is strictly related to data analysis and statistics.

Learning is a wide domain, consequently it can be ranched into subfields dealing with different types of learning. The most common partition is the one that distinguishes between supervised and unsupervised learning according to the types of training data available to the classifier (Breeden, 2021). Figure 2 depicts the word-cloud jargon of machine learning in credit risk modelling. In supervised learning, an algorithm is trained using labelled data to make predictions; this is the most common scenario when dealing with classification or regression problems. In unsupervised learning, an algorithm is fed with unlabelled data where an algorithm is tasked with learning from the data on its own and be able to make accurate predictions when given unseen data; this approach is popular in clustering and association problems. Another type of machine learning is reinforcement learning, where an intelligent agent ought to take actions in an environment in order to maximize the notion of cumulative reward. This is used largely for classification and control problems.

Decision tree (DT): Decision tree algorithm is a popular method for default prediction due to its simplicity, interpretability and its ability to handle large datasets with high dimensionality. It uses a tree-like model of decisions and their possible consequences, by recursively partitioning the feature space into smaller regions, in which the most homogeneous set of outcomes is found. However, decision trees are known to be sensitive to class imbalance since they tend to be biased towards the majority class. Breiman et al. (1984) and Fayyad and Irani (1992) give full description of the model.

k-Nearest neighbor (kNN): The k-nearest neighbour classifier (kNN) is known to be most useful instance-based learners. kNN is a non-parametric model. kNN (Yao and Ruzzo, 2006) is a non-parametric method that makes predictions based on the majority class of the k-nearest points to a given test point. It is simple and efficient but can be sensitive to the choice of k and the distance metric used. kNN has been used in the literature for default prediction, mainly in the credit risk domain. A comprehensive description of kNN is provided by Kelleher et al. (2020), Stephens and Diesing (2014) and Wilson and Martinez (1997).

Logistic regression model (LR): One of the most commonly used statistical models is the logistic regression model that explains the relationship of several covariates $\mathbf{x}$ to a binary response variable. The primary objective of the logistic regression model (Zhang et al., 2017) with multiple predictors is to construct a model to describe the relationship between a binary response variable and one or more predictor variables. Logistic regression is widely used in various fields, including medicine, finance, and social sciences, where binary classification tasks are common. It provides interpretable results because the coefficients can be examined to understand the impact of the features on the probability of the positive class. However, logistic regression assumes a linear relationship between the features and the log-odds, and may not perform well when dealing with complex nonlinear patterns in the data.

Naïve Bayesian approach (NB): The naïve Bayes classifier is a probabilistic classification algorithm based on Bayes' Theorem that has been widely used for default prediction problems. It makes the assumption that the predictors are independent given the class label, which is called the ”naive” assumption. Despite its ”naive” assumption, it has been shown to be effective in several studies and it can handle class imbalance by adjusting class weights or using techniques like oversampling, undersampling and synthetic data generation. A full description of the algorithm can be found in (De Campos et al., 2011) and (Stephens and Diesing, 2014).

Light gradient boosting machine (LGBM): LGBM is a powerful machine learning model that has gained popularity in both regression and classification tasks. It is a gradient boosting framework that utilizes tree-based learning algorithms (Ke et al., 2017). LGBM is designed to handle large-scale datasets efficiently, making it suitable for real-world applications with high-dimensional features. One of the key strengths of LGBM lies in its ability to handle imbalanced datasets effectively. It employs a technique called gradient-based one-side sampling (GOSS) to downsample the majority class during the boosting process, which helps to improve the model's performance on minority classes. This makes LGBM particularly well-suited for classification tasks where class imbalances are common. Ke et al. (2017) and Li et al. (2022) provide more context to the LGBM model.

Random forest (RF): As proposed by Breiman (2001), random forest is a combination of decision trees (Ho, 1995) used as an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classification or regression of the individual trees. Deng et al. (2018) defined random forest as a type of learning method that can be used for both classification and regression problems. Furthermore, random forests are suitable for large quantities of data with substantial noise, can prevent over-fitting, and are able to distinguish important features in classification. Breiman (2001), Calderoni et al., (2015) and Booth et al., (2015) give a more detailed discussion on random forests, including a more rigorous mathematical description.

Adaptive boosting (ADA): The Adaptive Boosting (ADA) Classifier is a popular machine learning algorithm used for classification tasks. It is a type of ensemble learning method that combines multiple weak classifiers to create a strong and accurate classifier. ADA iteratively trains a series of weak classifiers, where each subsequent classifier is designed to focus on the instances that were misclassified by the previous classifiers. This iterative process helps to improve the overall performance of the classifier. Granström and Abrahamsson (2019) provides the full details regarding the implementation of this algorithm.

Stochastic gradient descent classifier (SGDC): The stochastic gradient descent (SGDC) classifier is a popular and efficient algorithm used in machine learning for solving classification problems. It is a variant of the gradient descent optimization algorithm (Lokeswari and Amaravathi, 2018), specifically designed for large-scale datasets. The SGDC iteratively updates the model parameters by taking small steps in the direction of the steepest gradient, aiming to minimize the loss function. Unlike traditional gradient descent (Liu et al., 2021), which calculates the gradient over the entire training dataset, SGDC performs updates on randomly selected subsets of data called mini-batches. This stochastic nature of SGDC allows for faster convergence and makes it highly suitable for working with massive datasets.

Gradient boosting (GB): Gradient boosting is a powerful machine learning algorithm that combines multiple weak learners, typically decision trees, to create a strong predictive model. It operates by sequentially adding new models to correct the errors made by the previous models, thereby gradually improving its predictive accuracy. The algorithm works by optimizing a specific loss function through an iterative process. Each subsequent model is trained to minimize the errors or residuals of the previous models, using gradient descent optimization. Gradient boosting is known for its ability to handle complex nonlinear relationships in data and is widely used in various domains, including regression, classification, and ranking problems. It has gained popularity due to its high predictive performance and robustness. Dorogush et al., (2018) and Bentéjac et al., (2021) expand more on the model.

Extreme gradient boosting (XGB): Another integration technique constructed by continuous iterations of weak classifier is the extreme gradient boosting. According to Ogunleye and Wang (2019), the model was proposed by Chen and Guestrin (2016) to optimize memory usage and exploit the hardware computing power, XGB decreases the execution time with an increased performance compared to many machine learning algorithms and even deep learning models. The main idea of boosting is to sequentially build sub-trees from an original tree such that each subsequent tree reduces the errors of the previous one. In this procedure, k number of regression trees are created to ensure that the prediction of the tree cluster is as close to the actual value as possible and that the generalization capability is as high as possible. More details about the procedure can be read by Dhieb et al., (2019), Ogunleye and Wang (2019) and Chen and Guestrin (2016).

4. Data description

In this section we provide some information on the dataset utilised, exploratory data analysis and we also motivate the aptness for the selection of our model choice. Kaggle dataset was used in this paper, which contained 11 features and 150000 observations (Kaggle, 2023). Kaggle is a well-known platform for data science competitions, collaboration and learning. It hosts a wide variety of datasets contributed

Table 1. Credit risk dataset.

Classifier	Method	Type
SeriousDlqin2yrs	Person experienced 90 days past due delinquency or worse	Binary
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no instalment debt like car loans divided by the sum of credit limits	Ratio
Age	Age of borrower in years	Integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years (Bucket_1)	Integer
DebtRatio	Monthly debt payments, alimony,living costs divided by monthly gross income	Ratio
MonthlyIncome	Monthly income	Numeric
NumberOfOpenCreditLinesAndLoans	Number of Open loans (insta.0=lment like car loan or mortgage) and Lines of credit (e.g. credit cards)	Integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due (Bucket_3)	Integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	Integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years (Bucket_2)	Integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	Integer

Table 2. Simulated sample sizes.

Response		Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Sample 9
UNDER	**Yes**	10026	10026	10026	10026	10026	10026	10026	10026	10026
	No	100260	66840	50130	40104	33420	28646	25065	22280	20052
ADASYN	**Yes**	15553	24699	34994	46658	59989	75371	93316	114524	139974
	No	139974	139974	139974	139974	139974	139974	139974	139974	139974

by the community, covering diverse topics and domains. These datasets are often used for data analysis, machine learning projects and research. Kaggle datasets range from structured data in CSV files to images, videos and more complex data types. Table 1, gives the dictionary to the dataset being adopted in this paper.

Roughly 2% of the data was missing, particularly within the monthly income variable as well as the number of dependents. This is shown visually in Figure 3. We thought it was worthwhile to check if there is no relationship in our explanatory variables before attempting to fit any models. This task is termed as checking for the existence of multicollinearity within the data. Figure 4 displays the results that were obtained after the test was conducted. The results show a very strong correlation in the Bucket 1 through Bucket 3 variable. Moreover, a high correlation was also seen between Bucket 1 and Number Real Estate Loans Or Lines.

The dataset had originally 7% (10026) positive cases and 93% (139974) negative cases. Since the objective of this paper was to investigate the effectiveness of various machine learning models under class imbalance, we have generated nine samples of different levels of class imbalance for each of the sampling techniques discussed in Section 2.3. As a result, we ended up with eighteen (18) samples as shown in Table 2. In under-sampling, the minority class was kept the same while the majority class was under sampled to meet the desired class imbalance. On the other hand, in ADASYN sampling the majority class was fixed at original observations while the minority class was over sampled to meet the desired samples of different class imbalance.

4.1. Measures of performance

We adopt the widely used measures of performance in the fields of credit risk to evaluate our classification algorithms. These include the area covered by the receiver operating characteristics (ROC)

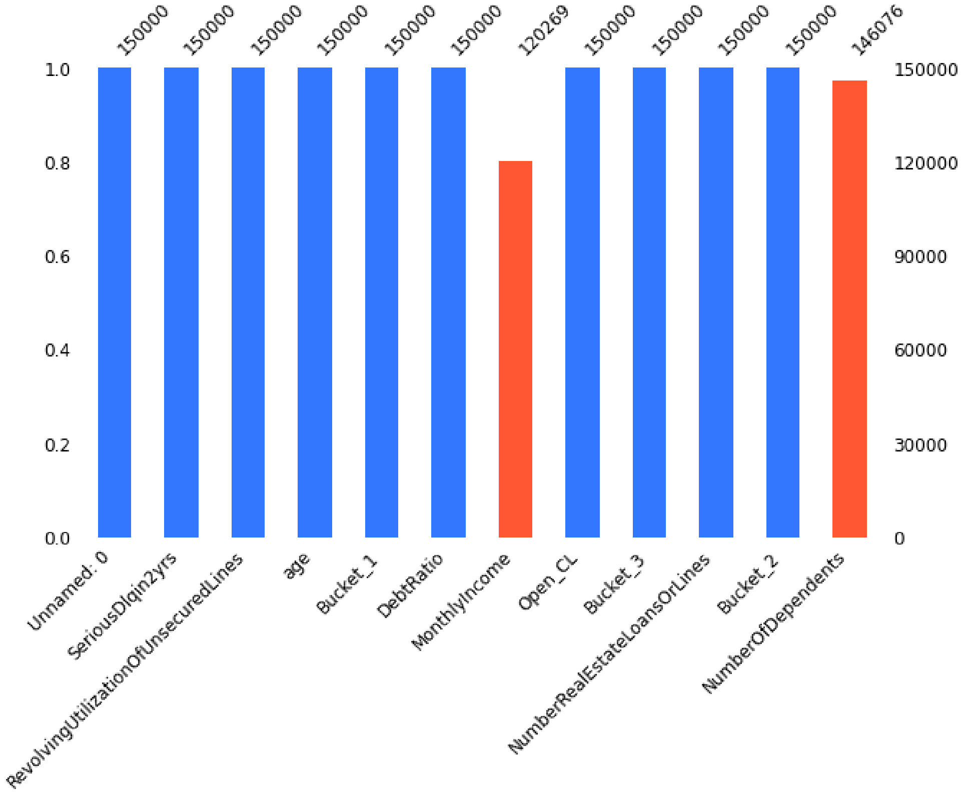
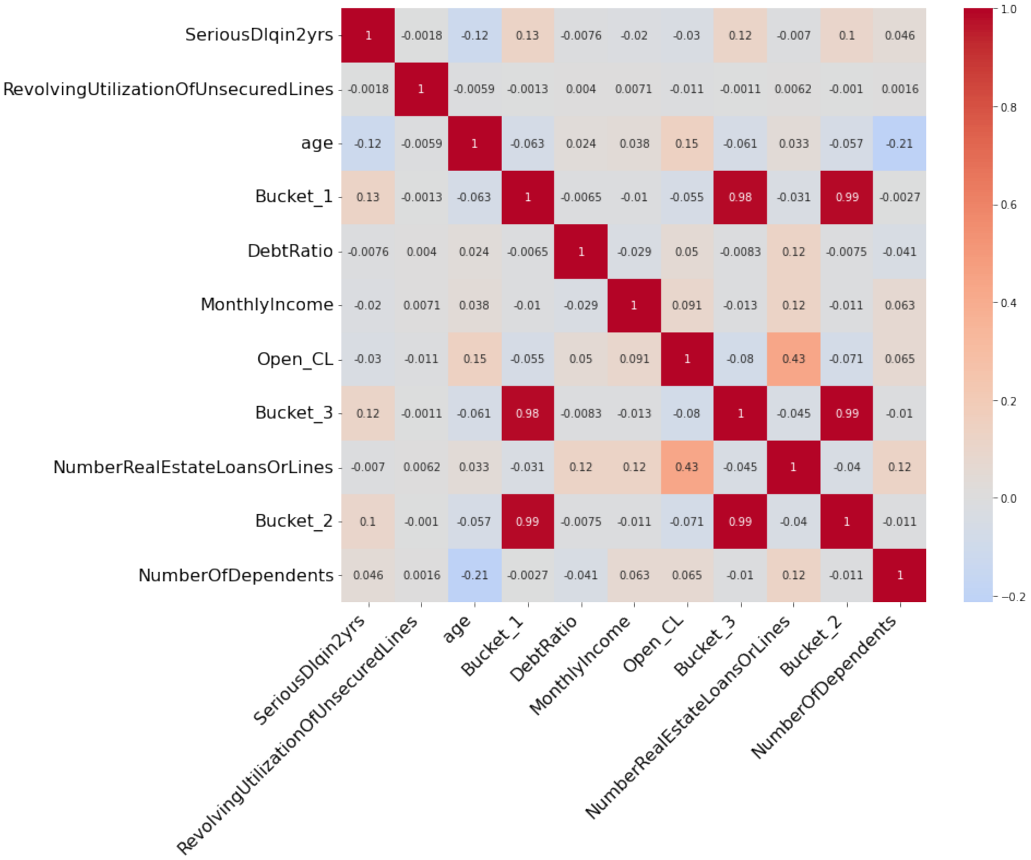


Figure 3. Missing values.



	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	Bucket_1	DebtRatio	MonthlyIncome	Open_CL	Bucket_3	NumberRealEstateLoansOrLines	Bucket_2	NumberOfDependents
SeriousDlqin2yrs	1	-0.0018	-0.12	0.13	-0.0076	-0.02	-0.03	0.12	-0.007	0.1	0.046
RevolvingUtilizationOfUnsecuredLines	-0.0018	1	-0.0059	-0.0013	0.004	0.0071	-0.011	-0.0011	0.0062	-0.001	0.0016
age	-0.12	-0.0059	1	-0.063	0.024	0.038	0.15	-0.061	0.033	-0.057	-0.21
Bucket_1	0.13	-0.0013	-0.063	1	-0.0065	-0.01	-0.055	0.98	-0.031	0.99	-0.0027
DebtRatio	-0.0076	0.004	0.024	-0.0065	1	-0.029	0.05	-0.0083	0.12	-0.0075	-0.041
MonthlyIncome	-0.02	0.0071	0.038	-0.01	-0.029	1	0.091	-0.013	0.12	-0.011	0.063
Open_CL	-0.03	-0.011	0.15	-0.055	0.05	0.091	1	-0.08	0.43	-0.071	0.065
Bucket_3	0.12	-0.0011	-0.061	0.98	-0.0083	-0.013	-0.08	1	-0.045	0.99	-0.01
NumberRealEstateLoansOrLines	-0.007	0.0062	0.033	-0.031	0.12	0.12	0.43	-0.045	1	-0.04	0.12
Bucket_2	0.1	-0.001	-0.057	0.99	-0.0075	-0.011	-0.071	0.99	-0.04	1	-0.011
NumberOfDependents	0.046	0.0016	-0.21	-0.0027	-0.041	0.063	0.065	-0.01	0.12	-0.011	1

Figure 4. Multicollinearity.

Table 3. Confusion matrix.

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True Positive (TP)	False positive (FP)
	Negative	False Negative (FN)	True Negative (TP)

curve. The ROC curve tells how much a model is capable of distinguishing between classes; an excellent model will have an ROC close to 1, a poor model will have ROC close to 0.5. The ROC curve is constructed by evaluating the fraction of "true positives"(TP) and "false positives" (FP) for different threshold values. Table 3 shows the so-called confusion matrix that contains basic ingredients that we usually report on.

We report on the following metrics,

Note:

These formulas are derived using 2x2 confusion matrix (Table 3) (Mitchell and Mitchell, 1997), for multi-class classification or multi-label classification the formulas will be different. ROC-AUC is a measure of the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) in a binary classification problem. AUC of 1 represents a perfect classifier and AUC of 0.5 represents a random classifier. Recall gain is used when the data is imbalanced and the goal is to improve recall and it is defined as: recall gain = (recall of model - recall of baseline)

5. Results

This section presents an overview of the evaluation measures obtained after training the classification models in R. Prior to fitting the models, our dataset underwent a simulation process to create nine samples with varying levels of class imbalance using both under-sampling and ADASYN sampling techniques, as depicted in Figure 5. Each sample was characterized by a specific percentage of positive target variables, such as 10% in the first sample and 90% for the remaining variables. Subsequent samples followed a similar pattern, with increasing percentages of positive target variables.

The evaluation focused on five key metrics: the area under the ROC curve, Mathews' correlation coefficient (MCC), Gini coefficient, recall scores and F1-scores. The results by the Gini coefficient and the recall scores can be found in Appendix 7. These metrics were analysed for the two sampling techniques investigated in the study. The area under the ROC curve provided insights into the models' performance in distinguishing between the positive and negative classes. MCC served as a measure of the models' overall performance, taking into account true positives, true negatives, false positives, and false negatives. Last, the F1-scores provided a balanced assessment of precision and recall for the models. These evaluation measures were instrumental in assessing the performance of the classification models and drawing meaningful conclusions about their effectiveness under different levels of class imbalance.

After generating and securely saving our simulated samples with varying degrees of class imbalance,

Measure (Chicco and Jurman, 2020)	**Description**
Mathews' correlation coefficient	The Matthews correlation coefficient (MCC) is a measure used to assess the quality of binary classification models, particularly when dealing with imbalanced datasets. It takes into account true positives, true negatives, false positives, and false negatives to provide a balanced evaluation of the classifier's performance.
F1-score	Harmonic mean of precision and recall
ROC-AUC	Receiver operating characteristic - area under the curve
Gini coefficient	This measures the area between the model's ROC curve and the baseline (random guessing) line. It quantifies how well the model can distinguish between positive and negative instances.
Recall score	Also known as true positive rate, it evaluates the model's ability to identify actual positive instances out of all the true positive instances. It is a measure of the model's sensitivity to detecting positive cases.

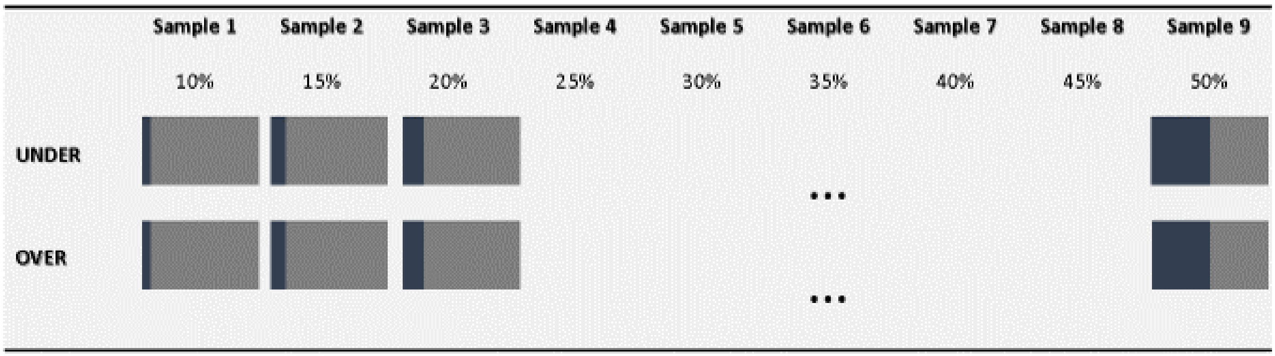


Figure 5. Simulation of class imbalance samples.

we proceeded to the modeling phase. Each simulated dataset was divided into two parts: 80% for training the model and 20% for model testing. We employed 10 machine learning models (as discussed on Section 3.) on each of the simulated samples and reported on Mathews' correlation coefficient to compare training sets. We further used F1-scores and area under the ROC curves to validate the performance of the models on out-of-sample datasets. Table 4 below, summarises the results obtained after training and the testing the models according to Mathews' correlation coefficient.

First, it is imperative to highlight the exceptional performance of the random forest (RF) and decision tree (DT) algorithms in both the training and testing phases. Even when trained on highly imbalanced datasets (Sample 1), both algorithms achieved Mathews' correlation coefficient (MCC) scores exceeding 99% and demonstrated slight improvements as the data approached balance. Remarkably, these models appeared to be relatively insensitive to class imbalance during the training stage. Visually, the dominance of RF and DT algorithms over other models is evident, as depicted in the line plots presented in Figure 6 and Figure 8. The second set of models that exhibited noteworthy performance included ADA, GB, XGB, LGBM and kNN. These models showcased significant improvements as the data became more balanced. With a 10% balanced dataset, these models achieved MCC scores ranging from 35.5–55.8%, which escalated to 54.6–61.1% for a fully balanced dataset. This trend was consistent across both sampling techniques. The line plots in Figure 6 and Figure 8 offer insights into the sensitivity of imbalanced data to these models. In contrast, LR, SGDC, and NB algorithms exhibited the poorest performance across both sampling techniques, scoring MCC values below 50% regardless of the class imbalance in the data. Further analysis on this matter will be explored in the upcoming section.

The subsequent step of the analysis involved evaluating the performance of these models on separate out-of-sample datasets, which were reserved for model testing. Once again, the standout performers were the RF and DT models. When employing under-sampling, these models achieved scores ranging from as low as 63.4% and 73.3% to as high as 65.7% and 75.5%, respectively. In the case of ADASYN sampling, their scores ranged from as low as 70.8% and 80.5% to as high as 94.2% and 97.3%, respectively. Although the scores decreased compared to the training sets, the RF and DT models continued to outperform the other models. This trend is also visually represented in the line plots depicted in Figure 7 and Figure 9.

Similarly, ADA, GB, XGB, LGBM and kNN models exhibited comparable patterns in the out-of-sample datasets as observed in the training samples, albeit with slightly lower scores. Notably, the kNN model demonstrated the most significant improvements as the data became more balanced. Despite fully balancing the datasets, the LD, LGBM, kNN, and NB models still scored MCC values below 50% when utilizing under-sampling. However, the kNN model did exhibit some enhancements in prediction quality when employing the ADASYN sampling technique, achieving an MCC score of 86.6%. On the other hand, the LD, LGBM, and NB models remained below 50% in terms of MCC scores.

The F1-scores presented in Table 5 further reinforce the observation that the RT and DT models were the top performers during the testing stage. When employing the under-sampling technique, these models achieved scores as low as 66.7% and 74.6%, respectively, which improved to 75.5% and 83.1% as the data became more balanced. In the case of ADASYN sampling, their scores improved from 73.5% and 81.6% to 96.1% and 98.2%, respectively. Across both sampling techniques, ADA, GB, XGB, and LGBM models demonstrated gradual improvements in predictive power as the data distribution approached equilibrium. SGDC and kNN models exhibited the most significant improvements as the dataset became more balanced. At a 10% class imbalance, SGDC and kNN models achieved F1-scores

Table 4. Mathews' correlation coefficient from training and testing sets.

		Model	**Sample 1**	**Sample 2**	**Sample 3**	**Sample 4**	**Sample 5**	**Sample 6**	**Sample 7**	**Sample 8**	**Sample 9**
Training Sets	**Under-Sampling**	**LR**	0.1717	0.2210	0.2259	0.2642	0.2840	0.3341	0.3062	0.4202	0.4117
		DT	**0.9977**	**0.9987**	**0.9988**	**0.9989**	**0.9990**	**0.9993**	**0.9991**	**0.9993**	**0.9994**
		RF	**0.9975**	**0.9986**	**0.9987**	**0.9989**	**0.9990**	**0.9993**	**0.9991**	**0.9993**	**0.9993**
		ADA	0.3547	0.4298	0.4565	0.4893	0.5181	0.5190	0.5261	0.5353	0.5460
		GB	0.4011	0.4743	0.4884	0.5217	0.5491	0.5552	0.5581	0.5658	0.5813
		XGB	0.5842	0.6472	0.6645	0.6941	0.7101	0.7170	0.7284	0.7316	0.7546
		LGBM	0.4700	0.5393	0.5570	0.5884	0.6049	0.6130	0.6222	0.6292	0.6388
		SGDC	0.0934	0.1457	0.1155	0.2283	0.2959	0.3614	0.3845	0.4441	0.4415
		KNN	0.5584	0.5888	0.5743	0.6001	0.6097	0.6046	0.6028	0.6049	0.6106
		NB	0.1107	0.0984	0.0955	0.0942	0.1313	0.1296	0.1045	0.1392	0.1126
	ADASYN Sampling	**LR**	0.1823	0.2143	0.2369	0.2769	0.2741	0.3415	0.3733	0.3712	0.3843
		DT	**0.9976**	**0.9981**	**0.9983**	**0.9984**	**0.9982**	**0.9987**	**0.9987**	**0.9988**	**0.9987**
		RF	**0.9975**	**0.9980**	**0.9983**	**0.9984**	**0.9981**	**0.9987**	**0.9987**	**0.9988**	**0.9987**
		ADA	0.3741	0.4277	0.4588	0.4888	0.5035	0.5202	0.5325	0.5438	0.5421
		GB	0.4081	0.4574	0.4930	0.5178	0.5354	0.5446	0.5586	0.5658	0.5644
		XGB	0.5666	0.5987	0.6179	0.6387	0.6470	0.6541	0.6644	0.6690	0.6680
		LGBM	0.4572	0.4996	0.5313	0.5514	0.5671	0.5782	0.5868	0.5959	0.5931
		SGDC	0.0000	0.0924	0.1761	0.2688	0.2640	0.3902	0.4159	0.4330	0.4251
		KNN	0.6121	0.7047	0.7576	0.8016	0.8307	0.8525	0.8694	0.8792	0.8886
		NB	0.1012	0.1024	0.1065	0.1233	0.1299	0.1412	0.1260	0.1207	0.1257
Testing Sets	**Under-Sampling**	**LR**	0.1813	0.2096	0.2388	0.2754	0.2617	0.3036	0.2967	0.4239	0.4164
		DT	0.**6337**	**0.6225**	**0.6324**	**0.6228**	**0.6349**	**0.6374**	**0.6567**	**0.6246**	**0.6240**
		RF	**0.7333**	**0.7371**	**0.7393**	**0.7394**	**0.7387**	**0.7261**	**0.7551**	**0.7389**	**0.7472**
		ADA	0.3565	0.4099	0.4489	0.4863	0.4995	0.4919	0.5256	0.5470	0.5448
		GB	0.3929	0.4356	0.4756	0.5064	0.5293	0.5177	0.5472	0.5742	0.5799
		XGB	0.4706	0.4917	0.5258	0.5714	0.5754	0.5611	0.6095	0.6054	0.6275
		LGBM	0.4159	0.4577	0.4973	0.5403	0.5443	0.5387	0.5753	0.5816	0.6061
		SGDC	0.1024	0.1344	0.1272	0.2351	0.2958	0.3302	0.3936	0.4503	0.4548
		KNN	0.3708	0.4109	0.3795	0.4279	0.4383	0.4115	0.3921	0.4213	0.4072
		NB	0.1199	0.0944	0.1008	0.1135	0.1387	0.1279	0.1153	0.1243	0.0989
	ADASYN Sampling	**LR**	0.1718	0.2050	0.2277	0.2747	0.2806	0.3399	0.3754	0.3701	0.3887
		DT	**0.7082**	**0.7999**	**0.8592**	**0.8806**	**0.9068**	**0.9211**	**0.9324**	**0.9363**	**0.9421**
		RF	**0.8048**	**0.8815**	**0.9178**	**0.9349**	**0.9491**	**0.9612**	**0.9627**	**0.9677**	**0.9725**
		ADA	0.3641	0.4236	0.4544	0.4875	0.5113	0.5209	0.5324	0.5413	0.5444
		GB	0.3940	0.4534	0.4771	0.5157	0.5322	0.5439	0.5549	0.5559	0.5666
		XGB	0.4767	0.5319	0.5625	0.5861	0.6143	0.6237	0.6299	0.6369	0.6459
		LGBM	0.4180	0.4814	0.5025	0.5331	0.5590	0.5632	0.5766	0.5784	0.5894
		SGDC	0.0000	0.0894	0.1810	0.2768	0.2650	0.3921	0.4125	0.4273	0.4275
		KNN	0.3950	0.4919	0.5875	0.6383	0.7005	0.7374	0.7684	0.7942	0.8164
		NB	0.0895	0.1003	0.1065	0.1225	0.1296	0.1439	0.1157	0.1198	0.1209

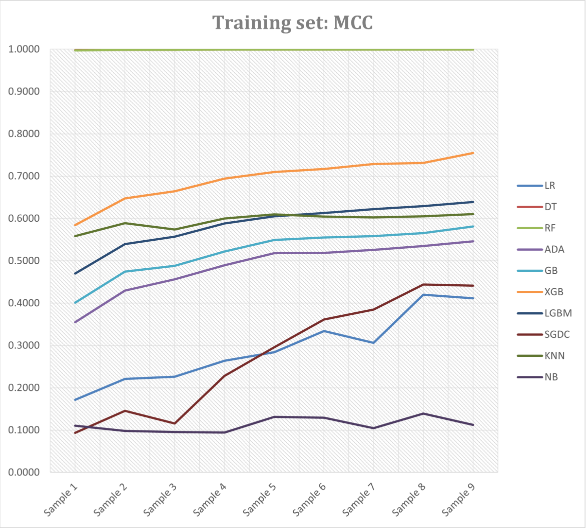


Figure 6. Under-sampling training set.

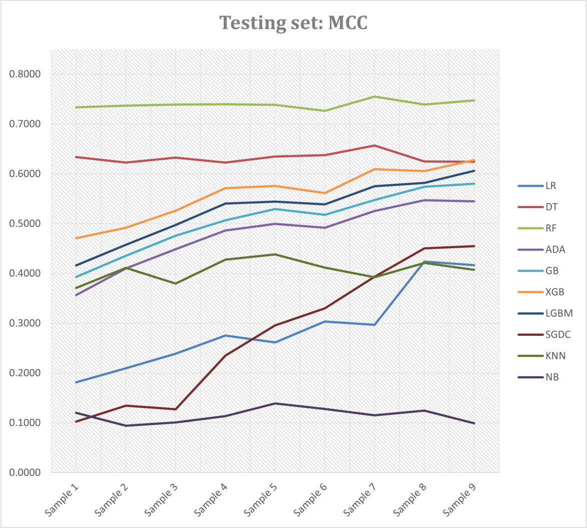


Figure 7. Under-sampling testing set.

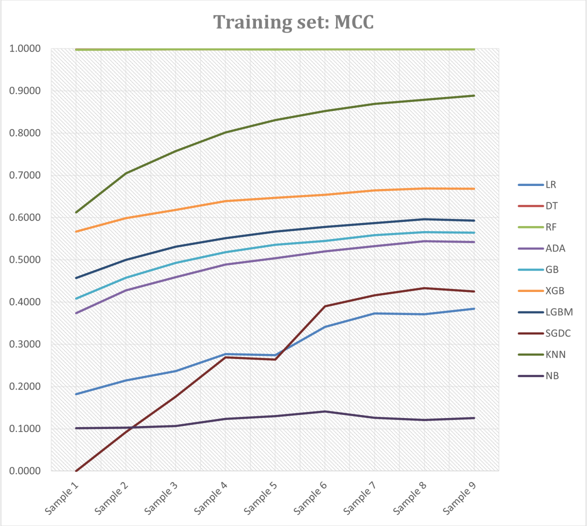


Figure 8. ADASYN sampling training set.

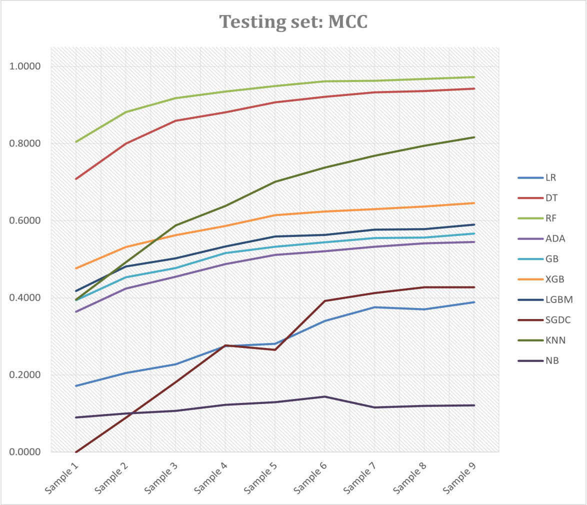


Figure 9. ADASYN sampling testing set.

Figure 10. Training and testing sample by Mathews' correlation coefficient.

Table 5. F1-score comparison for testing sets.

	Model	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Sample 9
Under-Sampling	LR	0.1078	0.1426	0.1827	0.2357	0.2348	0.3123	0.3101	0.4820	0.4882
	DT	**0.6673**	**0.6727**	**0.6976**	**0.6995**	**0.7231**	**0.7357**	**0.7588**	**0.7449**	**0.7554**
	RF	**0.7461**	**0.7633**	**0.7789**	**0.7866**	**0.7966**	**0.7958**	**0.8244**	**0.8181**	**0.8314**
	ADA	0.3604	0.4391	0.5002	0.5502	0.5782	0.5897	0.6342	0.6588	0.6687
	GB	0.3937	0.4705	0.5298	0.5698	0.6103	0.6172	0.6561	0.6849	0.7045
	XGB	0.4695	0.5244	0.5789	0.6349	0.6561	0.6599	0.7108	0.7157	0.7456
	LGBM	0.4147	0.4894	0.5541	0.6064	0.6302	0.6404	0.6833	0.6956	0.7265
	SGDC	0.0335	0.0591	0.0606	0.1611	0.2443	0.3261	0.4259	0.5126	0.5223
	KNN	0.3898	0.4505	0.4440	0.5042	0.5344	0.5268	0.5320	0.5646	0.5721
	NB	0.0680	0.0455	0.0530	0.0588	0.0933	0.0737	0.0709	0.0698	0.0573
ADASYN Sampling	LR	0.1059	0.1397	0.1713	0.2413	0.2557	0.3442	0.4043	0.4194	0.4580
	DT	**0.7349**	**0.8259**	**0.8826**	**0.9041**	**0.9282**	**0.9412**	**0.9517**	**0.9562**	**0.9614**
	RF	**0.8162**	**0.8961**	**0.9315**	**0.9480**	**0.9610**	**0.9713**	**0.9734**	**0.9779**	**0.9817**
	ADA	0.3678	0.4535	0.4987	0.5510	0.5873	0.6087	0.6346	0.6552	0.6682
	GB	0.3966	0.4870	0.5292	0.5838	0.6134	0.6353	0.6614	0.6734	0.6927
	XGB	0.4776	0.5615	0.6083	0.6474	0.6867	0.7059	0.7239	0.7388	0.7557
	LGBM	0.4181	0.5133	0.5559	0.6030	0.6397	0.6573	0.6822	0.6945	0.7130
	SGDC	0.0000	0.0259	0.0956	0.2140	0.1999	0.3953	0.4495	0.4818	0.4879
	KNN	0.4225	0.5467	0.6521	0.7108	0.7722	0.8078	0.8374	0.8607	0.8793
	NB	0.0510	0.0486	0.0632	0.0646	0.0805	0.0862	0.0668	0.0705	0.0730

of 3.4% and 40.0% respectively, which increased to 52.2% and 57.2% as the data became more balanced. This pattern held true for both sampling techniques employed in this study. On the other hand, LR and NB models performed consistently regardless of the model or sampling technique used.

Finally, we present visualizations of the ROC curves, depicting the area under the curve (AUC), for the testing stage across various sample sizes. The ROC curves, as shown in Figure 12, provide further confirmation of the previous findings. While random forest (RF) and decision tree (DT) models consistently exhibited superior performance compared to other classifiers across different levels of class imbalance, it is crucial for readers to pay attention to the performance improvements of all models as the data becomes more balanced. Notably, a noteworthy observation from the ROC curves is that the AUC values for the under-sampling technique, as illustrated in Figure 13, appeared to be relatively flatter and closer to the diagonal line compared to the AUC values for the ADASYN sampling technique, as depicted in Figure 12. This observation implies that ADASYN sampling tends to produce more reliable predictions compared to the under-sampling technique.

The visualizations of the ROC curves provide additional evidence of the strength of RF and DT models throughout various class imbalance scenarios. Furthermore, the results highlight the importance of considering the performance improvements of all models as data balance improves. Additionally, the ROC curves suggest that ADASYN sampling may offer enhanced prediction reliability compared to under-sampling.

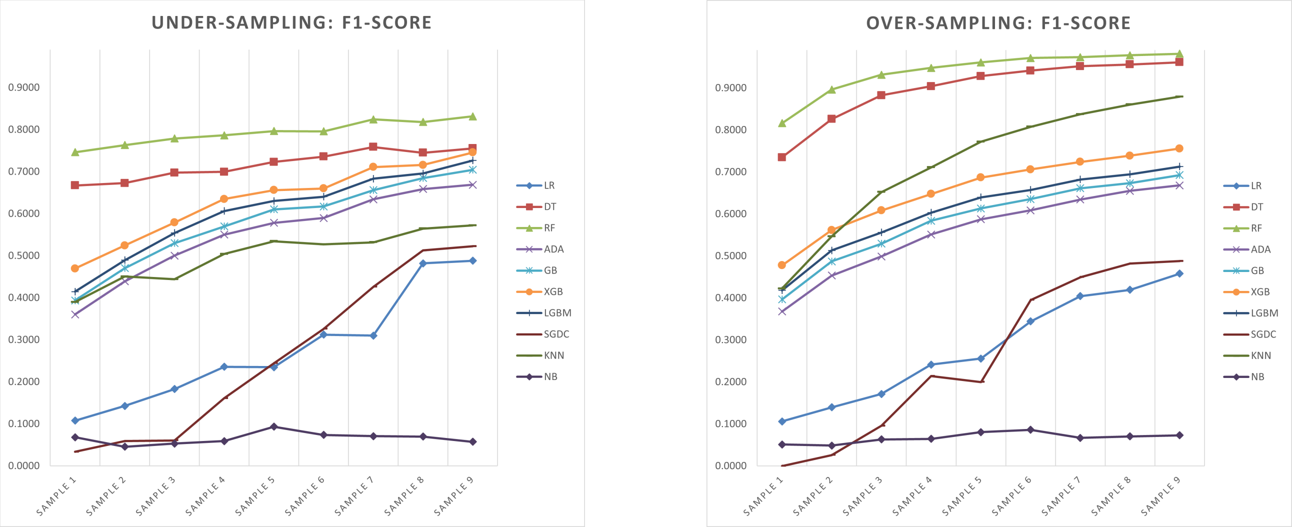


Figure 11. F1-Scores comparison.

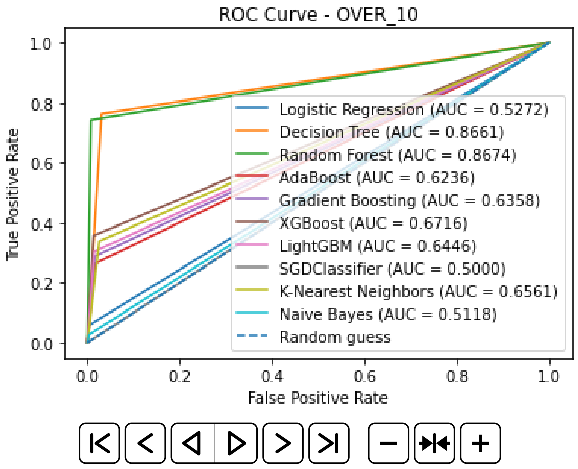


Figure 12. ADASYN sampling ROC curve comparison.

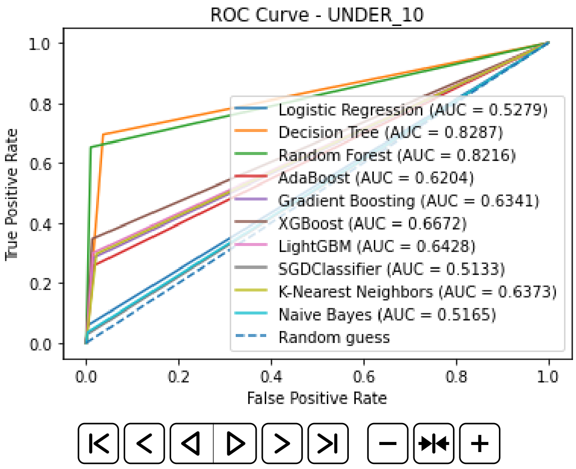


Figure 13. Under-sampling ROC curve comparison.

In summary, the results highlight the superior performance of RF and DT models in both training and testing stages, emphasizing their robustness to class imbalance. Other models, such as ADA, GB, XGB, LGBM and kNN, showed improvements as the data became more balanced but did not surpass the performance of RF and DT models. LR, SGDC and NB models consistently performed poorly regardless of the sampling technique used. The findings also suggest that ADASYN sampling technique yielded more reliable predictions compared to under-sampling technique.

6. Discussion

The results obtained in this study align with some findings reported in related work on the topic of class imbalance and classification models.

The superior performance of random forest (RF) and decision tree (DT) algorithms, especially in the presence of class imbalance, is consistent with previous research. Alija et al., (2023) and Zhou and Wang (2012) conducted a similar investigation under class imbalance and discovered that random forest tend to outperform many state-of-art classifiers such as SVM, ANN, naïve Bayes and C4.5. RF and DT models are known for their ability to handle imbalanced datasets effectively by capturing complex decision boundaries and handling both minority and majority classes well. In the paper written by Sun et al., (2018), it was also discovered that decision tree significantly outperforms other models and is effective for imbalanced enterprise credit evaluation. The high MCC scores achieved by RF and DT models in this study support their suitability for imbalanced classification tasks, as reported in previous studies. RF and DT algorithms perform well on imbalanced data due to their inherent robustness to class imbalance (Singhal et al., 2018), the use of sampling and randomness, their ability to handle overlapping regions and the benefits of ensemble methods. DT are less affected by class imbalance as they can capture patterns in both minority and majority classes (Liu et al., 2010) during the splitting process. RF, consisting of multiple decision trees trained on bootstrap samples, introduces randomness and diversity, enabling the algorithm to learn from both classes. DTs can form partitions that help separate the minority class instances, improving classification performance. Finally, the ensemble nature of RF leverages the collective wisdom of multiple trees, further enhancing its ability to handle imbalanced data (Liu et al., 2010).

The results also align with previous studies that have highlighted the challenges faced by logistic regression, stochastic gradient descent classifier, and naïve Bayes algorithms in imbalanced classification tasks. These models often struggle to handle class imbalance, resulting in lower MCC scores and poorer performance compared to other algorithms. Logistic regression, stochastic gradient descent classifier, and naïve Bayes algorithms face challenges in imbalanced classification tasks due to the skewed class distribution (Aljedaani et al., 2022), loss function optimization, assumption of feature independence, and sensitivity to data representation. According to Das et al., (2018), the skewed class distribution can lead to biased models and difficulties in capturing patterns for the minority class. The loss functions used by LR and SGDC may prioritize the majority class, resulting in biased decision boundaries and poor performance on the minority class. NB's assumption of feature independence can disregard rare but discriminative features for the minority class. Additionally, these algorithms may struggle to find sufficient evidence to accurately model the minority class (Das et al.,, 2018) due to its under-representation. To overcome these challenges, techniques like resampling, adjusting class weights, using different loss functions, or employing specialized algorithms designed for imbalanced data can be applied. The consistent poor performance of LR, SGDC and NB models in this study reinforces the need to carefully select appropriate classifiers when dealing with imbalanced datasets.

ADA, gradient boosting, extreme gradient boosting, light gradient boosting machine, and k-nearest neighbors models can be sensitive to imbalanced data due to their underlying mechanisms and characteristics:

1. **Data weighting and boosting:** Models like ADA, GB, XGB and LGBM utilize boosting techniques, where multiple weak classifiers are combined to form a strong classifier. In the presence of imbalanced data, these models tend to assign higher weights to misclassified instances from the minority class during the training process. This weighting scheme can result in an overemphasis on the minority class, potentially leading to misclassifications and biased decision boundaries (Okey et al., 2022).

2. **Loss function optimization:** Boosting algorithms aim to minimize a loss function by iteratively fitting models to the training data. In imbalanced datasets, the loss function (Fernando and Tsokos, 2021) used may not adequately capture the cost of misclassifying the minority class. As a result, the models might prioritize minimizing the overall loss (Laradji et al., 2015), which is dominated by the majority class, leading to a bias towards the majority class and reduced performance on the minority class.

3. **Nearest neighbor-based approach:** kNN algorithm makes predictions based on the class labels of its nearest neighbors. In the presence of imbalanced data, the sparsity (Padmaja et al., 2007) of the minority class can lead to situations where the nearest neighbors of a minority instance predominantly belong to the majority class. This can result in misclassifications and a tendency to favor the majority class during classification.

The gradual improvements observed in the performance of these models as the data became more balanced are consistent with the notion that as the class distribution becomes more even, classifiers tend to achieve better results. This observation supports the idea that balancing techniques, such as ADASYN sampling, can alleviate the negative impact of class imbalance on the performance of classifiers.

Overall, the findings of this study align with existing research on the performance of classification models in the presence of class imbalance. The superiority of RF and DT models, the challenges faced by LR, SGDC and NB models, and the performance improvements with data balancing techniques are consistent with previous findings. These results contribute to the growing body of knowledge on class imbalance and provide further evidence of the effectiveness of certain algorithms in imbalanced classification tasks.

7. Conclusions

In conclusion, this study investigated the performance of various classification models in the presence of class imbalance. The results shed light on the impact of class distribution on the effectiveness of different algorithms and the importance of data balancing techniques. The findings highlight the outstanding performance of random forest and decision tree algorithms, which consistently outperformed other models in both training and testing stages. These models demonstrated robustness to class imbalance and achieved high Mathews' correlation coefficient scores even when trained on highly imbalanced datasets. The visual representations and area under the ROC curves further supported their superiority over other classifiers.

On the other hand, logistic regression, stochastic gradient descent classifier, and naïve Bayes models exhibited poor performance regardless of the class imbalance in the data. These models struggled to handle imbalanced datasets and scored lower MCC values compared to other algorithms. The study also highlighted the performance improvements of models such as ADA, GB, XGB, LGBM, and kNN as the data became more balanced. These models showed increased predictive power and achieved higher MCC scores as the class distribution became more even. The results further emphasized the effectiveness of ADASYN sampling techniques in producing more reliable predictions compared to under-sampling techniques. The findings of this study align with prior research on imbalanced classification tasks, providing further evidence of the superiority of RF and DT models and the challenges faced by LR, SGDC, and NB models. The results contribute to the existing body of knowledge on class imbalance and highlight the importance of selecting appropriate algorithms and employing data balancing techniques for improved classification performance.

Overall, this study emphasizes the need for careful consideration of the choice of classification models and the implementation of data balancing techniques when dealing with imbalanced datasets. The results can inform practitioners and researchers in selecting the most suitable models for imbalanced classification tasks and guide the development of more effective approaches to address class imbalance challenges. This thorough investigation provides a deeper comparison of class imbalance's pivotal influence on predictive analytics. Through assessing ten diverse classification models using robust evaluation metrics, such as ROC curve area, Mathews' correlation coefficient (MCC) and F1-scores, this study furnishes empirical insights into these models' strengths and weaknesses within imbalanced datasets. Guided by data-driven model assessments and balancing approaches, this research serves as a valuable roadmap for practitioners and researchers grappling with imbalanced datasets. The results provide explicit guidelines for model selection and tailored data balancing techniques, ultimately enhancing classification performance in the face of class imbalance.

However, this study has certain limitations that should be acknowledged. First, the analysis was conducted using a specific dataset with its own characteristics, and the results may not generalize to other datasets or domains. Therefore, it is crucial to validate these findings on different datasets to ensure their applicability in diverse contexts. In our case, we have simulated eighteen (18) different samples. Second, the study focused solely on the performance of classification models and did not delve into the underlying reasons for the observed differences in performance. Future research could explore the specific factors contributing to the effectiveness or ineffectiveness of different models in handling class imbalance. This could involve examining feature importance, model interpretability, or identifying specific patterns in the data that affect model performance.

Funding

This work is based on the research supported wholly/in part by the National Research Foundation of South Africa (Grant Number 126885). This work is based on research supported in part by the Department of Science and Innovation (DSI) of South Africa. The grant holder acknowledges that opinions, findings, and conclusions or recommendations expressed in any publication generated by DSI-supported research are those of the authors and that the DSI accepts no liability whatsoever in this regard.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article, except for the machine learning models used for the analysis.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Tanja Verster, for their unwavering guidance, invaluable guidance, and exceptional mentorship throughout the course of this research. I would also like to extend my gratitude to the NWU BMI Centre for availing the resources for me and the wonderful staff we have.

Conflict of interest

The authors declare no conflict of interest.

References

Alija S, Beqiri E, Gaafar AS, et al. (2023) Predicting students performance using supervised machine learning based on imbalanced dataset and wrapper feature selection. *Informatica* 47. https://doi.org/10.31449/inf.v47i1.4519

Aljedaani W, Rustam F, Mkaouer MW, et al. (2022) Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry. *Knowl-Based Syst* 255: 109780. https://doi.org/10.1016/j.knosys.2022.109780

Anguita D, Ghelardoni L, Ghio A, et al. (2012) The'k'in k-fold cross validation. *in* 'ESANN', 441–446.

Bentéjac C, Csörgő A, Martínez-Muñoz G (2021) A comparative analysis of gradient boosting algorithms. *Artif Intell Rev* 54: 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Booth A, Gerding E, McGroarty F (2015) Performance-weighted ensembles of random forests for predicting price impact. *Quant Financ* 15: 1823–1835. https://doi.org/10.1080/14697688.2014.983539

Breeden J (2021) A survey of machine learning in credit risk. *J Credit Risk* 17. https://ssrn.com/abstract=3946261

Breiman L (2001) Random forests. *Mach learn* 45: 5–32. https://doi.org/10.1023/A:1010933404324

Breiman L, Friedman J, Olshen R, et al. (1984) Classification and regression trees (wadsworth, belmont, ca). 13: 978–0412048418.

Calderoni L, Ferrara M, Franco A, et al. (2015) Indoor localization in a hospital environment using random forest classifiers. *Expert Syst Appl* 42: 125–134. https://doi.org/10.1016/j.eswa.2014.07.042

Cawley GC, Talbot NL (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. *J Mach Learn Res* 11: 2079–2107.

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

Chicco D, Jurman G (2020) The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics* 21: 1–13. https://doi.org/10.1186/s12864-019-6413-7

Das S, Datta S, Chaudhuri BB (2018) Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recogn* 81: 674–693. https://doi.org/10.1016/j.patcog.2018.03.008

De Campos LM, Cano A, Castellano JG, et al. (2011) Bayesian networks classifiers for gene-expression data, in *2011 11th International Conference on Intelligent Systems Design and Applications*, IEEE, 1200–1206. https://doi.org/10.1109/ISDA.2011.6121822

Deng M, Chen J, Huang J, et al. (2018) Agricultural drought risk evaluation based on an optimized comprehensive index system. *Sustainability* 10: 3465. https://doi.org/10.3390/su10103465

Dhieb N, Ghazzai H, Besbes H, et al. (2019) Extreme gradient boosting machine learning algorithm for safe auto insurance operations, in *2019 IEEE international conference on vehicular electronics and safety (ICVES)*, IEEE, 1–5. https://doi.org/10.1109/ICVES.2019.8906396

Dorogush AV, Ershov V, Gulin A (2018) Catboost: gradient boosting with categorical features support. *arXiv preprint*. https://doi.org/10.48550/arXiv.1810.11363

Fayyad UM, Irani KB (1992) The attribute selection problem in decision tree generation, *in* 'AAAI', 104–110.

Fernando KRM, Tsokos CP (2021) Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE T Neur Net Learn Syst* 33: 2940–2951. https://doi.org/10.1109/TNNLS.2020.3047335

Granström D, Abrahamsson J (2019) Loan default prediction using supervised machine learning algorithms.

Han J, Kamber M, Pei J (2012) Data mining concepts and techniques third edition, *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University* .

Ho TK (1995) Random decision forests, in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, 1: 278–282.

Kaggle (2023) Give me some credit. Available from:
https://www.kaggle.com/competitions/GiveMeSomeCredit/dataselect=cs-training.csv.
Accessed: 2023-02-05.

Ke G, Meng Q, Finley T, et al. (2017) Lightgbm: A highly efficient gradient boosting decision tree. *Adv Neural Inf Process Syst* 30.

Kelleher JD, Mac Namee B, D'arcy A (2020) *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*, MIT press.

Khemakhem S, Boujelbene Y (2018) Predicting credit risk on the basis of financial and non-financial variables and data mining. *Rev Account Financ* 17: 316–340. https://doi.org/10.1108/RAF-07-2017-0143

Laradji IH, Alshayeb M, Ghouti L (2015) Software defect prediction using ensemble learning on selected features. *Inform Software Tech* 58: 388–402. https://doi.org/10.1016/j.infsof.2014.07.005

Leo M, Sharma S, Maddulety K (2019) Machine learning in banking risk management: A literature review. *Risks* 7: 29. https://doi.org/10.3390/risks7010029

Li K, Xu H, Liu X (2022) Analysis and visualization of accidents severity based on lightgbm-tpe. *Chaos, Solitons Fract* 157: 111987. https://doi.org/10.1016/j.chaos.2022.111987

Liu L, Li P, Chu M, et al. (2021) Stochastic gradient support vector machine with local structural information for pattern recognition. *Int J Mach Learn Cybe* 12: 2237–2254. https://doi.org/10.1007/s13042-021-01303-x

Liu W, Chawla S, Cieslak DA, et al. (2010) A robust decision tree algorithm for imbalanced data sets, in*Proceedings of the 2010 SIAM International Conference on Data Mining*, SIAM, 766–777.

Lokeswari N, Amaravathi K (2018) Comparative study of classification algorithms in sentiment analysis. *Int Res J Sci Eng Technol* 4: 31–39.

Mitchell TM, Mitchell TM (1997) *Machine learning*, 1: McGraw-hill New York.

Ogunleye A, Wang QG (2019) Xgboost model for chronic kidney disease diagnosis. *IEEE/ACM T Comput Bi* 17: 2131–2140. https://doi.org/10.1109/TCBB.2019.2911071

Okey OD, Maidin SS, Adasme P, et al. (2022) Boostedenml: Efficient technique for detecting cyberattacks in iot systems using boosted ensemble machine learning. *Sensors* 22: 7409. https://doi.org/10.3390/s22197409

Padmaja TM, Dhulipalla N, Bapi RS, et al. (2007) Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection. in *15th International Conference on Advanced Computing and Communications* (ADCOM 2007), IEEE, 511–516. https://doi.org/10.1109/ADCOM.2007.74

Patro S, Sahu KK (2015) Normalization: A preprocessing stage. *arXiv preprint arXiv: 1503.06462*. https://doi.org/10.48550/arXiv.1503.06462

Rubin DB (1976) Inference and missing data. *Biometrika* 63: 581–592.

Schafer JL, Graham JW (2002) Missing data: our view of the state of the art. *Psychol Methods* 7: 147. https://doi.org/10.1037/1082-989X.7.2.147

Singhal Y, Jain A, Batra S, et al. (2018) Review of bagging and boosting classification performance on unbalanced binary classification, in *2018 IEEE 8th International Advance Computing Conference (IACC)*, IEEE, 338–343. https://doi.org/10.1109/IADCC.2018.8692138

Stephens D, Diesing M (2014) A comparison of supervised classification methods for the prediction of substrate type using multibeam acoustic and legacy grain-size data. *PloS One* 9: e93950. https://doi.org/10.1371/journal.pone.0093950

Sun J, Lang J, Fujita H, et al. (2018) Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates. *Inform Sci* 425: 76–91. https://doi.org/10.1016/j.ins.2017.10.017

Thabtah F, Hammoud S, Kamalov F, et al. (2020) Data imbalance in classification: Experimental evaluation. *Inform Sci* 513: 429–441. https://doi.org/10.1016/j.ins.2019.11.004

Wilson DR, Martinez TR (1997) Improved heterogeneous distance functions. *J Artif Intell Res* 6: 1–34.

Yao Z, Ruzzo WL (2006) A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data, *BMC Bioinformatics*, BioMed Central, 7: 1–11. https://doi.org/10.1186/1471-2105-7-S1-S11

Zhang C, Liu C, Zhang X, et al. (2017) An up-to-date comparison of state-of-the-art classification algorithms. *Expert Syst Appl* 82: 128–150. https://doi.org/10.1016/j.eswa.2017.04.003

Zhou L, Wang H (2012) Loan default prediction on large imbalanced data using random forests. *TELKOMNIKA Indonesian J Electr Eng* 10: 1519–1525. https://doi.org/10.11591/telkomnika.v10i6.1323

A. Appendix A: Additional training and testing results summarised by Gini and recall measures.

This section provides more results that were obtained from training and testing of the machine learning models as summarised by the Gini score (Table 6) and the recall measure (Table 7). The results also conformed with the findings obtained using MCC, F1-score and the AUROC measures in Section 5.Once again, decision tree and random forest models outperformed the rest of the models across various samples of varying class imbalance regardless of the sampling technique used.

Table 6. Training results by the gini score.

	Model	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Sample 9
Under-sampling	LR	0.0558	0.0744	0.0968	0.1288	0.1254	0.1744	0.1704	0.3076	0.3088
	DT	**0.6574**	**0.6506**	**0.6652**	**0.6552**	**0.6602**	**0.6448**	**0.6756**	**0.6414**	**0.6502**
	RF	**0.6432**	**0.6698**	**0.7030**	**0.7088**	**0.7154**	**0.7136**	**0.7504**	**0.7276**	**0.7426**
	ADA	0.2408	0.3064	0.3574	0.4054	0.4254	0.4312	0.4764	0.4986	0.5022
	GB	0.2682	0.34	0.3904	0.4276	0.465	0.4658	0.5056	0.534	0.5506
	XGB	0.3344	0.3936	0.4452	0.5074	0.5248	0.5232	0.9414	0.5784	0.6126
	LGBM	0.2856	0.357	0.4194	0.4734	0.4934	0.4976	0.5436	0.5498	0.5834
	SGDC	0.0266	0.0646	0.0824	0.1386	0.1094	0.1796	0.1768	0.2898	0.3068
	KNN	0.2746	0.323	0.3044	0.3594	0.3792	0.3592	0.352	0.3804	0.3748
	NB	0.033	0.0208	0.0238	0.027	0.0428	0.034	0.0312	0.032	0.0244
ADASYN sampling	LR	0.0544	0.0726	0.0898	0.1318	0.1394	0.1998	0.2438	0.2514	0.2804
	DT	**0.7322**	**0.8368**	**0.8952**	**0.9164**	**0.9362**	**0.9476**	**0.9534**	**0.955**	**0.9578**
	RF	**0.7348**	**0.8616**	**0.9156**	**0.9424**	**0.9592**	**0.9708**	**0.9726**	**0.9768**	**0.9800**
	ADA	0.2472	0.3202	0.3546	0.4036	0.4344	0.4518	0.4744	0.4914	0.5006
	GB	0.2716	0.3554	0.3906	0.4438	0.4682	0.4862	0.5104	0.516	0.5338
	XGB	0.3432	0.4298	0.4754	0.5178	0.5592	0.5778	0.5946	0.6026	0.6242
	LGBM	0.2892	0.381	0.4212	0.4686	0.5016	0.5168	0.5392	0.5454	0.5626
	SGDC	0	0.0126	0.0486	0.1168	0.1076	0.2414	0.2824	0.3076	0.311
	KNN	0.3122	0.4454	0.5638	0.64	0.7172	0.7624	0.7968	0.8236	0.843
	NB	0.0236	0.0226	0.0282	0.0302	0.0368	0.0406	0.0298	0.0316	0.0328

Table 7. Training and testing results by recall.

		Model	Sample 1	sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Sample 9
Training	Under-sampling	LR	0.0613	0.0899	0.1304	0.1838	0.2603	0.3653	0.4479	0.5493	0.6505
		DT	**0.9957**	**0.9987**	**0.9986**	**0.9995**	**0.9995**	**0.9999**	**0.9998**	**0.9999**	0.9999
		RF	**0.996**	**0.9989**	**0.9988**	**0.9995**	**0.9995**	**0.9999**	**0.9998**	**0.9999**	**0.9999**
		ADA	0.2868	0.375	0.4438	0.5088	0.5526	0.6044	0.6424	0.6988	0.7434
		GB	0.3034	0.4183	0.4909	0.5481	0.5944	0.6394	0.6873	0.753	0.7894
		XGB	0.3999	0.507	0.5876	0.6403	0.6897	0.7322	0.7717	0.8142	0.8494
		LGBM	0.3264	0.4557	0.5279	0.5774	0.6328	0.6759	0.7209	0.7672	0.807
		SGDC	0.0152	0.0283	0.0897	0.1911	0.2922	0.372	0.4042	0.4393	0.5186
		KNN	0.4448	0.7002	0.8518	0.9336	0.9749	0.9918	0.9974	0.9994	0.9998
		NB	0.0271	0.0315	0.0369	0.0329	0.0343	0.0357	0.039	0.0453	0.0442
	ADASYN sampling	LR	0.0629	0.0853	0.1348	0.1904	0.2653	0.3608	0.4547	0.5487	0.6561
		DT	0.6707	0.8811	0.962	0.9832	0.9928	0.9982	0.9994	0.9997	0.9999
		RF	**0.6531**	**0.8801**	**0.9579**	**0.9827**	**0.9934**	**0.9986**	**0.9994**	**0.9997**	**0.9999**
		ADA	**0.2875**	**0.3687**	**0.4433**	**0.51**	**0.5583**	**0.606**	**0.6475**	**0.6948**	**0.7485**
		GB	0.2985	0.4083	0.4919	0.5444	0.6002	0.6434	0.6937	0.7492	0.7916
		XGB	0.3302	0.4595	0.5568	0.6171	0.6761	0.727	0.7669	0.8031	0.8471
		LGBM	0.3036	0.4405	0.5194	0.571	0.6315	0.6758	0.7232	0.762	0.8102
		SGDC	0.0154	0.0288	0.0927	0.1949	0.2994	0.3703	0.415	0.4372	0.5213
		KNN	0.2811	0.5047	0.6958	0.8272	0.9172	0.9674	0.9876	0.9973	0.9993
		NB	0.026	0.0297	0.0346	0.0329	0.0354	0.0377	0.038	0.045	0.0441
Testing	Under-sampling	LR	0.0525	0.0699	0.0762	0.0841	0.0978	0.0949	0.0976	0.0996	0.101
		DT	**0.9972**	**0.9975**	**0.9969**	**0.9974**	**0.9979**	**0.9975**	**0.9976**	**0.9982**	**0.998**
		RF	**0.9969**	**0.997**	**0.9971**	**0.9974**	**0.9979**	**0.9971**	**0.9974**	**0.9984**	**0.9978**
		ADA	0.2419	0.313	0.3514	0.372	0.3808	0.3857	0.3951	0.4001	0.4004
		GB	0.2576	0.3428	0.3884	0.3998	0.4218	0.4247	0.4336	0.4461	0.4465
		XGB	0.3569	0.4476	0.4768	0.5018	0.527	0.5282	0.5345	0.5475	0.5546
		LGBM	0.2789	0.3677	0.4191	0.4469	0.4628	0.4716	0.4784	0.493	0.4911
		SGDC	0.0149	0.0149	0.0014	0.023	0.0467	0.0507	0.0147	0.0491	0.0511
		KNN	0.2569	0.317	0.3387	0.3542	0.363	0.3588	0.3682	0.3782	0.3666
		NB	0.0278	0.0276	0.033	0.0336	0.0336	0.0356	0.0392	0.0339	0.0327
	ADASYN sampling	LR	0.0474	0.072	0.0943	0.089	0.0979	0.1003	0.1022	0.1037	0.1026
		DT	**0.3115**	**0.3724**	**0.3939**	**0.4327**	**0.4347**	**0.4062**	**0.4187**	**0.4269**	**0.4422**
		RF	**0.2448**	**0.3229**	**0.3687**	**0.4122**	**0.4101**	**0.4072**	**0.4152**	**0.4107**	**0.4328**
		ADA	0.2409	0.3106	0.3495	0.3847	0.38	0.3892	0.3885	0.3865	0.3981
		GB	0.2423	0.3248	0.3756	0.4232	0.4091	0.417	0.4253	0.42	0.4373
		XGB	0.2483	0.3263	0.3766	0.4072	0.4061	0.4082	0.4152	0.4181	0.4417
		LGBM	0.2473	0.3283	0.3727	0.4292	0.4292	0.4286	0.4424	0.4368	0.4517
		SGDC	0.0138	0.0137	0.002	0.0285	0.0492	0.0458	0.0111	0.0469	0.0501
		KNN	0.151	0.2126	0.2379	0.2656	0.2485	0.2655	0.2386	0.2572	0.2395
		NB	0.0242	0.0265	0.041	0.035	0.0382	0.0326	0.0302	0.0296	0.0263

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)