Volume 40(1), pp. 1–24
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)

Assessing the performance of machine learning models for default prediction under missing data and class imbalance: A simulation study

Lindani Dube* and Tanja Verster*

Received: 16 September 2023; Revised: 9 February 2024; Accepted: 19 March 2024

Abstract

In the field of machine learning, robust model performance is essential for accurate predictions and informed decision-making. One critical challenge that hampers the performance of machine learning algorithms is the presence of missing data. Missing values are ubiquitous in real-world datasets and can substantially impact the performance of predictive models. This study explored the impact of increasing levels of missing values on the performance of machine learning models. Simulated samples with missing values ranging from 5% to 50% were generated, and various models were evaluated accordingly. The results demonstrated a consistent trend of deteriorating model performance as the amount of missing values increases. Higher levels of missing values lead to decreased accuracy scores across all models. Among the models evaluated, decision trees (DT) and random forests (RF) consistently demonstrated high accuracy scores across all sampling techniques, showcasing their robustness in handling missing values. Logistic regression (LR) also performed relatively well, showing consistent performance across different levels of missing values. On the other hand, stochastic gradient descent classifier (SGDC), K-nearest neighbours (kNN), and naïve Bayes (NB) models consistently exhibited lower accuracy scores across all sampling techniques, indicating limitations in handling missing values even when the dataset was more balanced. Furthermore, the study highlights the superiority of the SMOTE (Synthetic Minority OVER-sampling Technique) sampling technique compared to the UNDER-sampling approach. Models trained using SMOTE consistently achieved higher accuracy scores across all levels of missing values. This suggests that SMOTE sampling effectively handles imbalanced datasets and enhances classification performance, particularly when dealing with missing values. As the quest for accurate predictions gains paramount importance, addressing the pervasive challenge of missing data emerges as a cornerstone for unlocking the true potential of machine learning in real-world applications.

Key words: Credit Risk, Machine Learning, Modelling, Classification, Imbalance, Supervised Models, Missing Values

*Centre for Business Mathematics & Informatics, North-West University, 11 Hoffman St, Potchefstroom, 2531, South Africa
National Institute for Theoretical and Computational Sciences (NITheCS), Stellenbosch, 7600, South Africa
lindani.dube@nwu.ac.za

http://dx.doi.org/10.5784/40-1-001

1 Introduction

Credit risk prediction is a critical task in the financial industry, aiming to assess the likelihood of borrowers defaulting on their loan obligations. Accurate credit risk prediction models empower financial institutions to make informed decisions when granting credit, thus mitigating potential financial losses. However, one of the challenges faced in credit risk prediction, as well as in many other domains, is the presence of missing values and class imbalance in the dataset. Missing values can arise due to various reasons such as data entry errors, incomplete customer information, or intentional omissions. Addressing missing values is crucial as they can potentially bias the predictions and hinder the overall performance of machine learning models. Class imbalance occurs whenever the number of instances from one class is significantly larger/lower than the number of instances from the other class. In the context of credit risk modelling (Dube and Verster, 2023), this will refer to situations where the number of non-defaulters outweighs the number of defaulters by a large margin.

This research paper aims to investigate the performance of various machine learning models when dealing with missing values in credit risk prediction, while also considering the effect of different sampling techniques in addressing the class imbalance. The focus was on simulated samples with varying degrees of missing values, ranging from 5% to 50%. Additionally, class imbalance is a common challenge in credit risk prediction, where the number of defaulting borrowers is lower than the number of non-defaulting borrowers. Hence, this study will examine the impact of different sampling techniques, including synthetic minority oversampling technique (SMOTE), adaptive synthetic sampling (ADASYN), UNDER-sampling, and OVER-sampling (Section 3.2), on model performance.

Understanding the behaviour of machine learning models in the presence of missing values and class imbalance is important due to their prevalence in real-world scenarios. Missing data can introduce bias, distort patterns, and affect the generalisation capabilities of models, while class imbalance can lead to skewed predictions and reduced performance. Therefore, developing robust strategies to handle missing values and address class imbalance is crucial for maintaining the integrity and accuracy of credit risk models.

The research will be conducted using simulated datasets that allow for control over the degree and distribution of missing values, as well as the level of class imbalance. Simulating missing values provide an opportunity to systematically explore the impact of varying levels of incompleteness on model performance, without being confounded by other factors present in real-world datasets. Additionally, employing different sampling techniques enables an investigation into their performance of addressing class imbalance, which can significantly impact the predictive performance of models. Moreover, financial institutions can benefit from this research by gaining a deeper understanding of how missing data and class imbalance impact model performance, and by leveraging the identified techniques to improve their credit risk assessment processes. Ultimately, the goal is to develop more accurate and reliable credit risk prediction models that can facilitate better decision-making and risk management practices in the financial industry, while effectively handling missing values and addressing class imbalance.

The introduction Section 1 highlights the significance of credit risk prediction and the

growing use of machine learning algorithms in this domain. Section 2 discusses previous studies that have explored similar research topics, while Section 3 outlines the experimental approach used in this study. The paper considers a diverse set of machine learning models (Section 4), including decision trees, random forests, logistic regression, gradient boosting, extreme gradient boosting, light gradient boosting machine, Gaussian naïve Bayes, and stochastic gradient descent classifier. Section 5 provides insights into the datasets used in the study, emphasising the presence of missing values and the strategies employed to handle them. The results Section 6 presents the performance metrics of the machine learning models under different sampling techniques, highlighting the overall performance of various machine learning models under missing values and class imbalance. The discussion Section 8 delves into the factors that influenced the models' performance and compares the advantages and limitations of the different sampling techniques. Finally, the conclusion and future research Section 9 summarises the findings, suggesting avenues for further exploration, such as investigating additional sampling techniques and optimising hyper-parameters to improve model performance.

2 Related Work

DT, kNN, NB, LR, light gradient boosting machine (LGBM), adaptive boosting (ADA), gradient boosting (GB), extreme gradient boosting (XGB), random forest (RF), and stochastic gradient decent classifier (SGDC) have been widely used in the literature for default prediction in finance, credit risk and banking, specifically when dealing with missing data and class imbalance. DT, LR and RF have been used in various studies for default prediction when dealing with class imbalance. For example, in a paper written by (Chang et al., 2016) the authors used decision trees to classify credit risk and handle class imbalance by adopting SMOTE sampling technique. Another example is (Alam et al., 2020), where the authors applied decision trees to credit card default prediction and handled class imbalance by using UNDER-sampling and OVER-sampling methods.

XGB, GB and kNN have been used in various studies for default prediction when dealing with class imbalance. For example, (He et al., 2022) the authors applied XGB to credit scoring and handled class imbalance by using a cost-sensitive approach and sampling techniques. Another example is by Wang et al. (2022), where the authors applied XGB to credit default prediction and handle class imbalance by adoption of SMOTE approach and adjusting the misclassification cost.

NB has been used in various studies for default prediction when dealing with class imbalance. For example, in a paper authored by Mahajan et al. (2022) the authors used naïve Bayes to classify credit risk and handle class imbalance by Gaussian-SMOTE method. Another example is by Ferreira et al. (2017), where the authors found that on average, sampling techniques outperform ensembles and cost-sensitive approaches.

LGBM has been employed in various studies for default prediction when addressing class imbalance. For instance, in the research article "Prediction of 30-day readmission: an improved gradient boosting decision tree approach" by Du et al. (2019), LGBM was utilised to predict the 30-day patient readmission in a hospital and tackled class imbalance by applying SMOTE sampling. Another study by Zhou et al. (2019), found that LGBM,

in conjunction with feature engineering and UNDER-sampling techniques, demonstrated superior performance in credit scoring with imbalanced data. The SGDC has been investigated in several studies concerning class imbalance and missing values. For example, in the paper authored by Aydın et al. (2023), the SGDC classifier was utilised for soil classification and handling missing values by employing imputation techniques such as kNN imputation and mean imputation.

In general, these techniques have been found to be effective in handling class imbalance in default prediction, but the specific approach that works best may depend on the specific characteristics of the dataset and the problem at hand. It is also worth mentioning, that other techniques such as UNDER-sampling, OVER-sampling, and synthetic data generation have been used to handle class imbalance. The best approach is to try different techniques and see which one gives the best results.

3 Proposed Methodology

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. Various approaches have been adopted, in this paper we will be exploring the results across OVER-sampling, UNDER-sampling, SMOTE as well as ADASYN sampling. Figure 1 outlines the proposed methodology that is adopted in this paper. We follow the traditional approach in model development, with data preparation as our first step, balancing our dataset using 4 sampling techniques as our second step, creating various samples of varying missing values in the third step, model training and testing in the forth step and finally concluding in the final step.

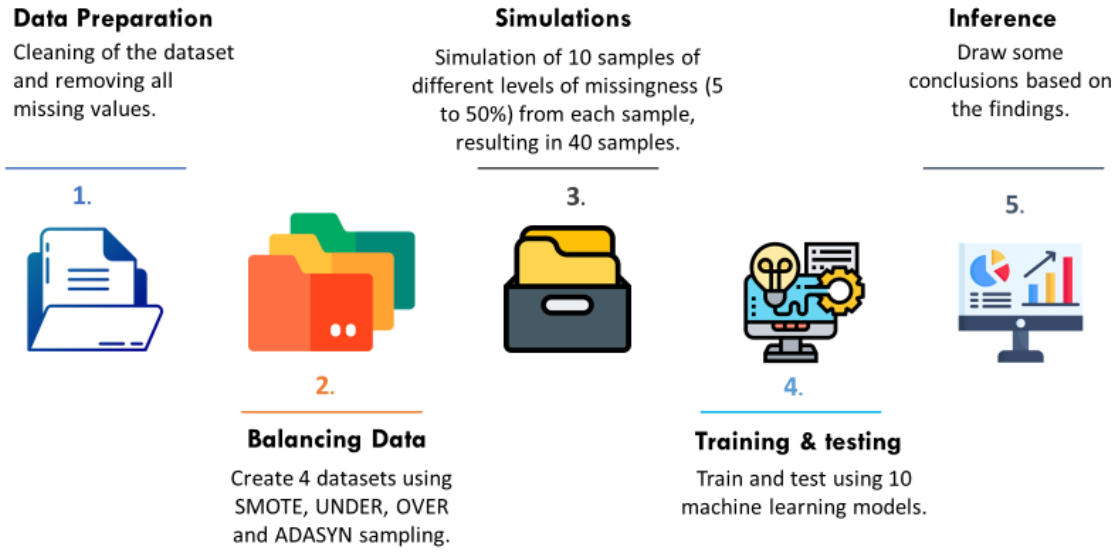

Figure 1: Adopted Proposed Approach

3.1 Data Preparation

The primary and essential stage in handling any data involves thoroughly cleaning and ensuring its coherence by removing irrelevant entries. With regard to our dataset's characteristics, the initial action taken was to standardise all explanatory variables through min/max scaling. This approach aimed to address the potential bias that may arise when training an algorithm with disparate variable ranges. For instance, a salary variable ranging from 10,000 to 665,000 could inadvertently be assigned greater importance than a ratio-based variable like Credit utilisation. By utilizing min/max scaling, as described in the work by Patro and Sahu (2015), the numerical variables were rescaled within the range of 0 to 1, thus mitigating such biases.

3.1.1 Missingness

In the presence of missing values in data, one option is to either eliminate those entries or fill in the gaps through imputation Han et al. (2012). The first strategy is simply to ignore missing values and the second strategy is to consider the 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 small number 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. The complete case analysis approach excludes all observations with missing values for any variable of interest. This approach thus limits the analysis to those observations for which all values are observed which often results in biased estimates 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 the maximum available data for analysis even if some of its variables have missing values. A disadvantage of this method is that it uses different sample sizes for different variables Schafer and Graham (2002). The sample size for each individual analysis is higher than the complete case analysis.

Impute Missing Values

Imputation of missing data involves replacing missing values with plausible alternatives to maintain the power of data mining and analysis techniques Rubin (1976). Different imputation methods aim to accurately estimate population parameters based on the extent of missing data. It is beneficial to compare results before and after imputation, although there is no fixed rule for determining the threshold of problematic missing data.

In this study, we employ median imputation as a technique to handle missingness. Median imputation replaces missing values in a dataset with the median value of non-missing observations for the corresponding variable. This method assumes that the missing data are randomly distributed and that the median represents the central tendency of the data.

To calculate the median, the non-missing observations are ordered, and the middle value or the average of two middle values (in case of an even number of observations) is determined. The imputed median value replaces all missing values for that variable.

3.2 Balancing Data

This section introduces several data level methods that are often implemented for the issue of class imbalance. 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. We first describe the random OVER-sampling and random UNDER-sampling methods and follow to further discuss sampling techniques that are based on synthetic sampling namely SMOTE and ADASYN sampling methods.

3.2.1 Over-sampling and under-sampling

While a number of strategies have been proposed for supervised learning with imbalanced data, possibly the two simplest are random over-sampling (OVER) and random under-sampling (UNDER). While OVER-sampling randomly samples from the minority case to produce an equal distribution of positive and negative cases, UNDER-sampling randomly removes the majority cases to produce an equal distribution Liu (2004). As an example, of an original dataset with 10 positive cases and 100 negative cases, OVER-sampling would produce a new dataset with 100 positive and negative cases each, while UNDER-sampling would create a dataset with 10 positive and negative cases each. Both methods have well-understood drawbacks: while UNDER-sampling discards potentially useful data, OVER-sampling increases the probability of over-fitting.

3.2.2 Synthetic minority over-sampling technique (SMOTE)

These two (OVER & UNDER) can also be combined in several ways to remedy class imbalance. One popular method is called SMOTE-ENN (Synthetic Minority OVER-sampling Technique - Edited Nearest Neighbours). SMOTE Chawla et al. (2002) is a method for OVER-sampling the minority class by creating synthetic samples. SMOTE-ENN first applies the SMOTE algorithm to oversample the minority class, and then it applies the ENN algorithm to the resulting dataset. ENN is used to remove any synthetic samples that are too similar to existing minority class examples, leaving a final dataset that is balanced and less prone to over-fitting. By combining OVER-sampling and UNDER-sampling techniques, SMOTE-ENN link can balance the class distribution while also reducing the risk of over-fitting. We will call this procedure SMOTE.

3.2.3 Adaptive Synthetic Sampling (ADASYN)

ADASYN sampling He et al. (2008) is a machine learning technique specifically developed to handle the issue of class imbalance commonly encountered in datasets. In contrast to traditional approaches, ADASYN sampling takes a unique approach by generating synthetic examples for the minority class, with a particular emphasis on instances that are

considered difficult to learn. The fundamental objective of ADASYN sampling is to enhance the performance of classifiers by augmenting the representation of the minority class. By leveraging the underlying distribution of the minority class, ADASYN sampling dynamically adjusts its sampling strategy, ensuring a more balanced and representative dataset. This adaptability allows ADASYN sampling to effectively address varying degrees of class imbalance within a given dataset. The generation of synthetic samples in ADASYN sampling provides the classifier with additional training data specifically tailored to the minority class, facilitating the capturing of intricate patterns and improving the accuracy of predictions. Overall, ADASYN sampling presents a promising solution for handling imbalanced datasets and can significantly contribute to the advancement of machine learning algorithms in such scenarios.

3.3 Simulations

To create different samples of varying levels of class missing values, the credit risk dataset (Section 5) was balanced using four (4) sampling techniques. We then simulated ten (10) samples from each balanced sample of varying missing values (5%, 10%, ...,50%) which resulted in forty samples. We randomly removed a certain proportion of observations from each independent variable, and subsequently filled in the missing values by using median imputation. Each of these datasets was further split using (80/20%) rule for training and testing the models. To train the models, we used 5-fold cross-validation on the 80% training datasets to avoid model over-fit.

3.4 Training & Testing

After spending a good amount of time cleaning and preprocessing our data we then moved to the following stage of training the models on the 80% of the data that were set aside for training the models. The accuracy measure was used to compare the performance of various machine learning models across different sampling techniques. The remaining 20% bulk of the data was used to validate the models. Again, accuracy measure was taken as well as F1-scores and AUROC.

3.5 Inference

Based on the results, we then drew some conclusions on the performance of the models.

4 Machine Learning Algorithms

In the field of data analytics, machine learning refers to a collection of computational techniques that leverage past information to enhance performance or make accurate predictions, Breeden (2021). Experience gained from analysing historical data, specifically the classifier's past encounters, plays a pivotal role in the performance of machine learning. The term "experience" in this context pertains to the historical data utilised by the machine learning method, specifically the classifier. The performance of machine learning

relies heavily on data quality and quantity, making it closely associated with data analysis and statistics.

Machine learning encompasses various subfields that deal with different types of learning, often categorised as supervised and unsupervised learning based on the availability of training data for the classifier Breeden (2021). In supervised learning, an algorithm is trained using labelled data to make predictions, commonly employed for classification and regression problems. Unsupervised learning involves algorithms processing unlabelled data to autonomously learn patterns and predict outcomes, commonly used in clustering and association problems. Reinforcement learning is another type of machine learning where an intelligent agent takes actions in an environment to maximise cumulative reward, typically utilised in classification and control problems. For this paper, our interest will be in the supervised machine learning only. Table 1 depicts ten supervised machine learning models that are investigated in this paper along with more literature for the reader.

Table 1: Supervised Machine Models

Algorithm	**Python Package**	**References**
Logistic Model (LR)	Sklearn	Schein and Ungar (2007) & Bittencourt et al. (2007)
Decision Tree (DT)	Sklearn	Swain and Hauska (1977) & Du and Zhan (2002)
Random Forest (RF)	Sklearn	Pal (2005) & Liu et al. (2012)
Adaptive Boosting (ADA)	Sklearn	An and Kim (2010) & Hu et al. (2008)
Gradient Boosting (GB)	Sklearn	Xu et al. (2014) & Ahmed (2021)
Extreme Gradient Boosting (XGB)	XGBoost	Dhieb et al. (2019) & Bansal and Kaur (2018)
Light Gradient Boosting Classifier (LGBM)	LightGBM	Taha and Malebary (2020) & Khafajeh (2020)
Stochastic Gradient Descent Classifier (SGDC)	Sklearn	Kabir et al. (2015) & Osho and Hong (2021)
K-Nearest Neighbour (kNN)	Sklearn	Yigit (2013) & Islam et al. (2007)
Naïve Bayesian (NB)	Sklearn	Leung et al. (2007) & Murphy et al. (2006)

5 Credit Risk Dataset

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 is a well-known platform for data science competitions, collaboration, and learning. It hosts a wide variety of datasets contributed 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. A Kaggle dataset "*Give Me Some Credit*" was used in this paper, which contained 11 features and 150 000 observations (Kaggle, 2023). Table 2 gives the description of the dataset being adopted in this paper. The dataset had originally 7% (10 026) positive cases and 93% (139 974) negative cases. Roughly about 2% of the data was missing, particularly within the monthly income variable as well as the number of dependents.

Table 2: Credit Risk Dataset

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

5.1 Assessment Measures

We adopted the widely used measures of performance in the fields of credit risk to evaluate our classification algorithms. These include accuracy scores, F1-scores and the area covered by the receiver operating characteristics (also called AUROC) curve. The receiver operating characteristics (ROC) curve Chicco and Jurman (2020) tells how much a model is capable of distinguishing between classes; an excellent model will have 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. These formulas are derived using 2x2 confusion matrix 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. An AUC of 1 represents a perfect classifier and an AUC of 0.5 represents a random classifier.

6 Training and Testing Results by Accuracy

We fitted the models on four (4) sampling techniques namely UNDER-sampling, OVER-sampling, SMOTE and ADASYN sampling. However, since OVER-sampling, SMOTE and ADASYN sampling all balance the data by OVER-sampling the minority class their performance was very closely related. We will only present results obtained from SMOTE and UNDER-sampling here but include the rest of the results in the Appendix.

Table 3: Training Results by Accuracy Scores

		Model	Sample 1 5%	Sample 2 10%	Sample 3 15%	Sample 4 20%	Sample 5 25%	Sample 6 30%	Sample7 35%	Sample 8 40%	Sample 9 45%	Sample 10 50%
Training Sets	SMOTE Sampling	**LR**	0.7371	0.7367	0.7286	0.7209	0.7224	0.7085	0.6988	0.7012	0.6896	0.6811
		DT	**0.9998**	**0.9999**	**0.9998**	**0.9998**	**0.9997**	**0.9990**	**0.9986**	**0.9967**	**0.9941**	**0.9880**
		RF	**0.9998**	**0.9999**	**0.9998**	**0.9998**	**0.9997**	**0.9990**	**0.9985**	**0.9967**	**0.9941**	**0.9880**
		ADA	0.9339	0.9329	0.9312	0.9313	0.9230	0.9122	0.9101	0.8943	0.8765	0.8580
		GB	0.9483	0.9479	0.9455	0.9446	0.9395	0.9317	0.9273	0.9161	0.9008	0.8828
		XGB	0.9698	0.9700	0.9687	0.9677	0.9668	0.9633	0.9614	0.9547	0.9467	0.9349
		LGBM	0.9627	0.9620	0.9617	0.9612	0.9598	0.9557	0.9535	0.9459	0.9327	0.9231
		SGDC	0.7472	0.7610	0.7448	0.7274	0.7290	0.7238	0.7144	0.7009	0.6738	0.6787
		KNN	0.9173	0.9144	0.9150	0.9138	0.9138	0.9114	0.9122	0.9102	0.9070	0.9034
		NB	0.5478	0.5218	0.5332	0.5347	0.5232	0.5352	0.5387	0.5267	0.5344	0.5263
	UNDER Sampling	**LR**	0.6897	0.6883	0.6883	0.6843	0.6887	0.6685	0.6661	0.6565	0.6729	0.6460
		DT	**0.9997**	**0.9996**	**0.9995**	**0.9995**	**0.9989**	**0.9969**	**0.9946**	**0.9887**	**0.9837**	**0.9738**
		RF	**0.9997**	**0.9996**	**0.9995**	**0.9995**	**0.9989**	**0.9968**	**0.9946**	**0.9886**	**0.9837**	**0.9738**
		ADA	0.7764	0.7695	0.7690	0.7618	0.7548	0.7515	0.7473	0.7348	0.7326	0.7280
		GB	0.7915	0.7859	0.7822	0.7729	0.7688	0.7618	0.7588	0.7473	0.7407	0.7369
		XGB	0.8896	0.8810	0.8709	0.8685	0.8484	0.8533	0.8422	0.8252	0.8168	0.8029
		LGBM	0.8315	0.8244	0.8206	0.8114	0.8046	0.7999	0.7902	0.7850	0.7801	0.7671
		SGDC	0.7094	0.6922	0.6909	0.6967	0.6946	0.6770	0.6681	0.6612	0.6629	0.6422
		KNN	0.7585	0.7553	0.7532	0.7485	0.7490	0.7480	0.7489	0.7501	0.7398	0.7488
		NB	0.5176	0.5165	0.5172	0.5184	0.5174	0.5192	0.5131	0.5126	0.5106	0.5169
Testing Sets	SMOTE Sampling	**LR**	0.7356	0.7303	0.7307	0.7168	0.7206	0.7130	0.6997	0.6966	0.6902	0.6775
		DT	**0.9323**	**0.9331**	**0.9300**	**0.9316**	**0.9284**	**0.9282**	**0.9225**	**0.9104**	**0.9085**	**0.8963**
		RF	**0.9577**	**0.9586**	**0.9567**	**0.9574**	**0.9563**	**0.9533**	**0.9527**	**0.9436**	**0.9366**	**0.9268**
		ADA	0.9322	0.9325	0.9301	0.9311	0.9242	0.9128	0.9116	0.8923	0.8745	0.8561
		GB	**0.9470**	**0.9467**	**0.9426**	**0.9447**	**0.9388**	**0.9332**	**0.9271**	**0.9140**	**0.8989**	**0.8822**
		XGB	**0.9618**	**0.9620**	**0.9608**	**0.9617**	**0.9604**	**0.9585**	**0.9567**	**0.9474**	**0.9393**	**0.9280**
		LGBM	**0.9599**	**0.9599**	**0.9586**	**0.9599**	**0.9587**	**0.9563**	**0.9524**	**0.9430**	**0.9303**	**0.9213**
		SGDC	0.7441	0.7566	0.7440	0.7266	0.7250	0.7244	0.7169	0.6970	0.6716	0.6737
		KNN	0.8660	0.8629	0.8674	0.8687	0.8688	0.8681	0.8719	0.8640	0.8617	0.8594
		NB	0.5462	0.5194	0.5315	0.5338	0.5229	0.5360	0.5393	0.5208	0.5325	0.5216
	UNDER Sampling	**LR**	0.6924	0.6949	0.6836	0.6857	0.6818	0.6637	0.6605	0.6559	0.6662	0.6527
		DT	0.6882	0.6967	0.6885	0.6641	0.6800	0.6754	0.6499	0.6460	0.6442	0.6343
		RF	**0.7648**	**0.7687**	**0.7591**	**0.7542**	**0.7492**	**0.7457**	**0.7336**	**0.7222**	**0.6935**	**0.6928**
		ADA	**0.7662**	**0.7666**	**0.7595**	**0.7595**	**0.7496**	**0.7442**	**0.7460**	**0.7279**	**0.7201**	**0.7180**
		GB	**0.7740**	**0.7747**	**0.7687**	**0.7676**	**0.7570**	**0.7535**	**0.7474**	**0.7396**	**0.7201**	**0.7194**
		XGB	**0.7733**	**0.7666**	**0.7620**	**0.7499**	**0.7520**	**0.7400**	**0.7428**	**0.7364**	**0.7190**	**0.7176**
		LGBM	**0.7730**	**0.7701**	**0.7648**	**0.7627**	**0.7545**	**0.7527**	**0.7527**	**0.7386**	**0.7169**	**0.7190**
		SGDC	0.7127	0.7049	0.6878	0.6956	0.6775	0.6658	0.6548	0.6538	0.6626	0.6396
		KNN	0.6236	0.6293	0.6204	0.6194	0.6179	0.6272	0.6211	0.5960	0.6073	0.6048
		NB	0.5158	0.5140	0.5144	0.5161	0.5161	0.5151	0.5108	0.5112	0.5055	0.5087

Table 3 provides a summary of accuracy scores from various machine learning models trained on different training sets and tested on corresponding testing sets of different levels

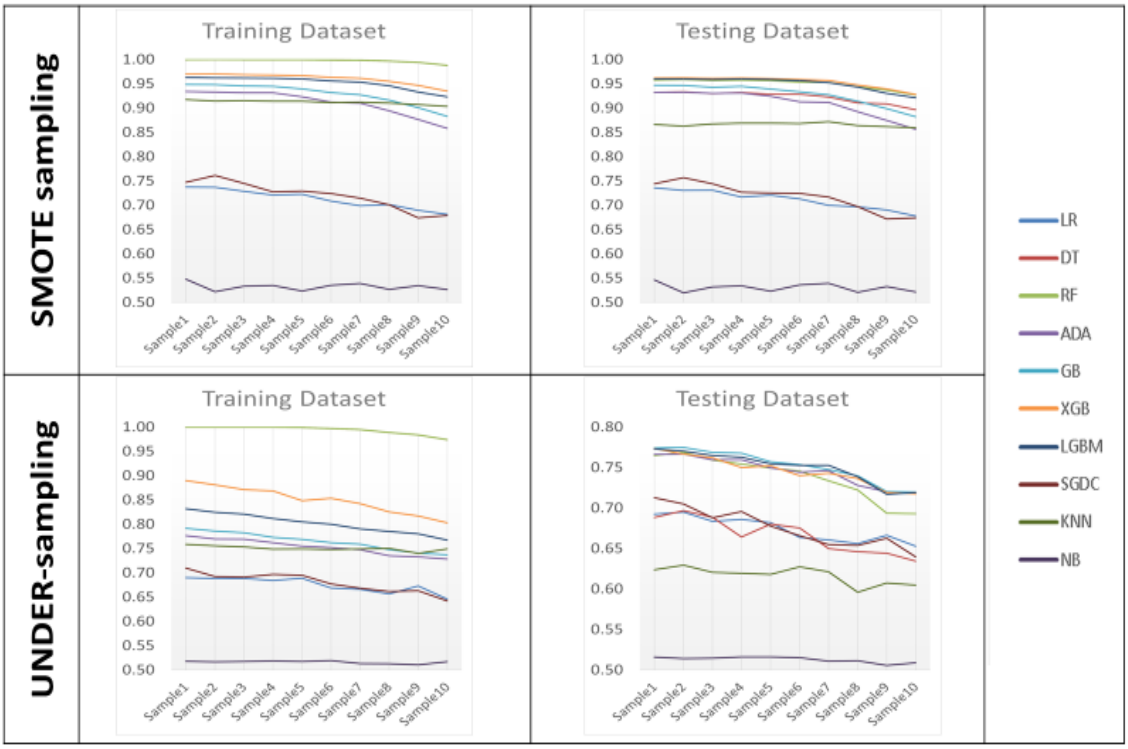

Figure 2: SMOTE and UNDER-sampling results by accuracy

of missing values. Each row corresponds to a specific model, and each column represents a different sample of different levels of missing values ranging from 5% to 50%. We further provided an infographic visual in Figure 2 to the accuracy scores to help visualising the results better.

Firstly, we observed the training results for SMOTE sampling. The decision tree and random forest models achieved exceptionally high accuracy scores, consistently above 99%. This indicated that these models had learned the training data very well, resulting in high performance. Other models like logistic regression, gradient boosting, extreme gradient boosting, and light gradient boosting machine also exhibited relatively high accuracy scores, ranging from 68% to 97%. However, models like Gaussian naïve Bayes and stochastic gradient descent classifier had lower accuracy scores, ranging from 54% to 76%, indicating suboptimal performance on the training data.

Secondly, we analysed the training results for UNDER-sampling. Similar to the SMOTE sampling results, the DT and RF models achieved excellent accuracy scores, above 99%, implying successful learning on the training data. The LR, GB, and XGB models maintained accuracy scores between 73% and 89%, while LGBM performed slightly lower with scores ranging from 77% to 83%. SGDC and K-nearest neighbours models exhibited accuracy scores in the range of 64% to 75%. Notably, the NB model consistently performed poorly with accuracy scores around 51%, indicating limitations in capturing the training data's complexity.

Thirdly, we shifted the focus to the testing results with SMOTE sampling. The DT and RF models, which had shown high accuracy during training, also demonstrated strong performance on the testing data with scores above 93%. LR, GB, and XGB models maintained accuracy scores ranging from 68% to 95%, suggesting good generalisation. However, the performance of the NB model remained consistently low, achieving accuracy scores around 52%, indicating limited ability to generalise on unseen data. Overall, the testing results aligned with the training results, with DT and RF models performing the best.

Fourthly, we examined the testing results for UNDER-sampling. The DT and RF models continued to excel, with accuracy scores exceeding 92%. LR, GB, and XGB models achieved accuracy scores ranging from 88% to 94%, indicating good generalisation on the testing data. The performance of the NB model remained consistently low, with accuracy scores around 51%, reinforcing its limited ability to generalise. Comparing the testing results with SMOTE sampling, we observed similar trends across models, indicating consistency in their performance across different sampling techniques.

Finally, we can summarise the trends observed. DT and RF models consistently performed the best, achieving high accuracy scores across both training and testing sets with different sampling techniques. LR, GB, and XGB machines showed slightly lower accuracy but remained competitive. Models like NB and SGDC consistently underperformed, indicating limitations in their ability to capture the complexities of the dataset. Overall, the results suggested that DT and RF models were well-suited for this particular task, while other models may have required further improvements to achieve better performance.

7 Model Evaluation by F1-scores and AUROC

Table 4: Testing Results by F1-scores

	Model	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample7	Sample 8	Sample 9	Sample 10
SMOTE Sampling	**LR**	0.7183	0.7135	0.7156	0.6982	0.7021	0.6908	0.6691	0.6658	0.655	0.6379
	DT	**0.9325**	**0.9334**	**0.9302**	**0.9319**	**0.9288**	**0.9286**	**0.9228**	**0.9113**	**0.9084**	**0.8962**
	RF	**0.9568**	**0.9577**	**0.9558**	**0.9565**	**0.9554**	**0.9522**	**0.9515**	**0.9429**	**0.9351**	**0.9254**
	ADA	0.9302	0.9307	0.9284	0.9290	0.9220	0.9096	0.9072	0.8887	0.8687	0.8482
	GB	**0.9454**	**0.9452**	**0.9409**	**0.9430**	**0.9366**	**0.9306**	**0.9235**	**0.9103**	**0.8924**	**0.8749**
	XGB	**0.9608**	**0.9611**	**0.9598**	**0.9608**	**0.9594**	**0.9573**	**0.9552**	**0.9459**	**0.9366**	**0.9250**
	LGBM	**0.9588**	**0.9589**	**0.9575**	**0.9588**	**0.9575**	**0.9549**	**0.9507**	**0.9412**	**0.9268**	**0.9176**
	SGDC	0.6954	0.7186	0.6986	0.6643	0.6676	0.6702	0.6506	0.6263	0.5714	0.5739
	KNN	0.8600	0.8570	0.8620	0.8626	0.8627	0.8629	0.8660	0.8600	0.8557	0.8535
	NB	0.1820	0.0789	0.1305	0.1374	0.0999	0.1450	0.1571	0.1062	0.1347	0.1091
UNDER Sampling	**LR**	0.6816	0.6824	0.6677	0.6684	0.6609	0.6439	0.6343	0.6266	0.6276	0.6048
	DT	0.6875	0.6963	0.6919	0.6681	0.6813	0.6726	0.6489	0.6469	0.6462	0.6303
	RF	**0.7641**	**0.7683**	**0.7543**	**0.7464**	**0.7454**	**0.7405**	**0.7276**	**0.7201**	**0.6899**	**0.6855**
	ADA	**0.7575**	**0.7588**	**0.7472**	**0.7491**	**0.7391**	**0.7309**	**0.7330**	**0.7076**	**0.7046**	**0.7369**
	GB	**0.7731**	**0.7741**	**0.7633**	**0.7624**	**0.7523**	**0.7425**	**0.7363**	**0.7239**	**0.7019**	**0.6945**
	XGB	**0.7720**	**0.7625**	**0.7559**	**0.7470**	**0.7492**	**0.7318**	**0.7336**	**0.7275**	**0.7080**	**0.7032**
	LGBM	**0.7729**	**0.7707**	**0.7593**	**0.7582**	**0.7498**	**0.7459**	**0.7431**	**0.7301**	**0.7070**	**0.7029**
	SGDC	0.7042	0.7149	0.6364	0.6509	0.6309	0.6688	0.5903	0.6049	0.6088	0.6013
	KNN	0.6013	0.5985	0.5853	0.5915	0.5903	0.6035	0.5892	0.5654	0.5831	0.5793
	NB	0.0966	0.0927	0.0964	0.1050	0.1073	0.1001	0.0861	0.0898	0.0556	0.0930

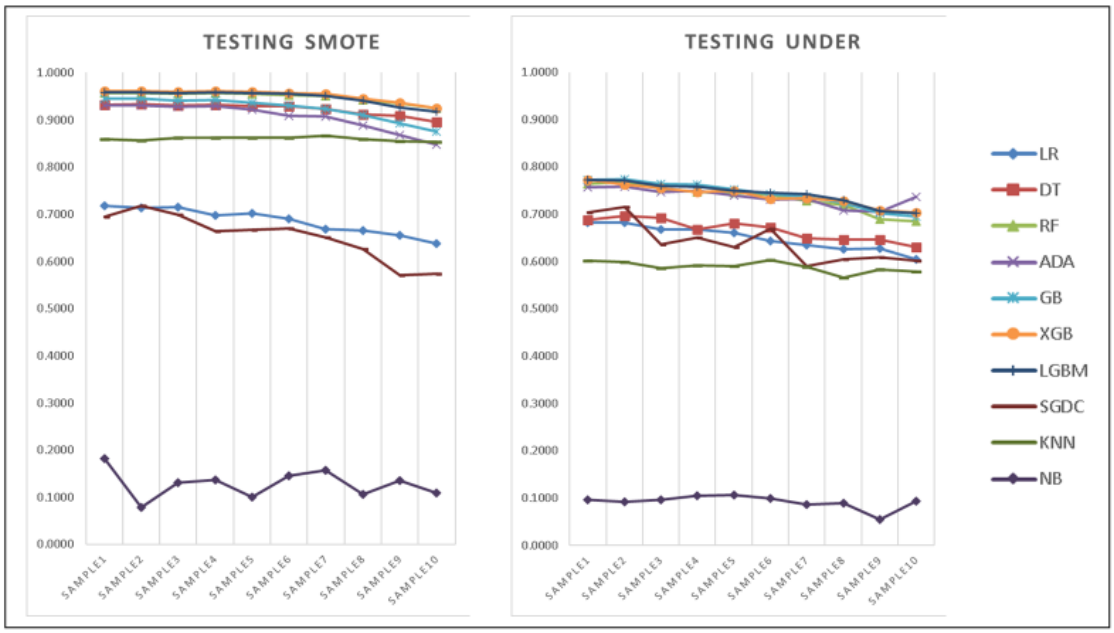

Figure 3: SMOTE and UNDER-sampling F1-scores

The next part of the analysis, we provide the testing/evaluation results in Table 4 for two different sampling techniques as summarised by F1-scores: SMOTE and UNDER-Sampling. Trends are also presented in line graphs in Figure 3.

In the SMOTE Sampling section, several models were evaluated based on their performance on different samples of varying missingness levels. Looking at the LR model, the F1-score ranged from 71.83% (Sample 1) to 63.79% (Sample 10). This indicated some

variation in the model's performance across different samples. On the other hand, the DT (Decision Tree) model consistently performed well with F1-score ranging from 93.25% (Sample 1) to 89.62% (Sample 10), showing a more stable and robust performance. Similar trends could be observed for the RF, ADA, GB, XGB, and LGBM models, where their accuracies remained relatively high and consistent across samples.

In contrast, the SMOTE Sampling results for the SGDC, KNN, and NB models showed lower accuracies compared to the other models. For instance, SGDC ranged from 69.54% (Sample 1) to 57.39% (Sample 10), indicating more fluctuation in performance. KNN also showed lower accuracies ranging from 86.00% (Sample 1) to 85.35% (Sample 10), which suggested less stability. The NB model performed poorly with accuracies below 20% for all samples, indicating an ineffective classification performance.

Moving on to the UNDER-Sampling section, similar observations could be made. LR's F1-score ranged from 68.16% (Sample 1) to 60.48% (Sample 10), showing a slight decrease compared to its SMOTE Sampling performance. The DT model exhibited accuracies ranging from 69.19% (Sample 1) to 63.03% (Sample 10), which was lower than its SMOTE Sampling performance. However, the RF, ADA, GB, XGB, and LGBM models maintained relatively high accuracies and consistency across samples, similar to their performance in the SMOTE Sampling section.

Once again, the SGDC, KNN, and NB models showed lower accuracies compared to other models in the UNDER-Sampling section. SGDC's F1-score ranged from 70.42% (Sample 1) to 60.13% (Sample 10), showing more fluctuation. KNN performed with accuracies ranging from 60.13% (Sample 1) to 57.93% (Sample 10), while the NB model continued to perform poorly with accuracies below 10% for most samples.

In summary, the evaluation results demonstrated that the SMOTE Sampling technique generally produced higher and more consistent accuracy scores across various models compared to the UNDER-Sampling technique. The DT, RF, ADA, GB, XGB, and LGBM models performed particularly well in both sampling techniques, showcasing their robustness in classification tasks. On the other hand, the SGDC, KNN, and NB models exhibited lower accuracies and more variation in performance, suggesting limitations in their ability to effectively classify the data.

The ROC curve plots presented in the Appendix also confirmed the deterioration in the predictive quality of the models as the number of missing values in the dataset increases. Of course, the prominent significant difference in AUROC of the two sampling techniques adopted cannot be left out. SMOTE sampling produced AUROC curves that were higher than that of UNDER-sampling by a large margin, thus affirming its superior performance over UNDER-sampling. This pattern was consistent throughout the samples, please refer to the appendix section for more results.

8 Discussion

The observed results in Section 6 can be attributed to several factors that may influence the performance of the machine learning models. These factors can help explain why certain models exhibit higher or lower accuracy & F1-scores under different sampling techniques.

Firstly, DT and RF models consistently performed the best across both training and testing sets, regardless of the sampling technique. These models are known for their ability to handle complex datasets and capture intricate relationships between features, which could explain their high accuracy, AUROC & F1-score scores. Their non-linear nature allows them to effectively learn from the training data and generalise well to unseen data, resulting in strong performance. These findings were in line with the comparative study that was done by Madaan et al. (2021) & Padimi et al. (2022), where the authors found that RF and DT perform exceptionally well in default prediction. RF and DT often outperform other machine learning models in scenarios involving missing data and class imbalance due to their robustness to missing values, ability to handle class imbalance by creating adaptive splits, capacity to capture non-linear relationships, and model interpretability. DT can naturally accommodate missing data by assigning them to separate branches, and they can prioritise minority classes during splits, making them suitable for imbalanced data. RF builds an ensemble of trees on diverse data subsets, mitigating class imbalance effects (Leevy et al., 2018), and their averaging reduces the impact of missing data. Additionally (Rokach, 2016), both models can capture intricate patterns in data, which is crucial when facing non-linear relationships, while DT offer transparency, aiding in understanding the model's behaviour in these challenging data situations.

Secondly, models like LR, GB, XGB, and LGBM also showed relatively high accuracy & F1-score scores, although slightly lower than the DT and RF models. Again, the findings aligned with the outcomes of Anand et al. (2022) These models utilise different techniques, such as optimising a differentiable loss function or combining weak learners (GB, XGB, LGBM), to make accurate predictions (Machado et al., 2019). While they may not match the performance of DT and RF models, their robustness and ability to handle diverse datasets contribute to their competitive performance. Furthermore, the motivation behind the commendable performance of models like LR, GB, XGB, and LGBM lies in their distinct methodologies and techniques. Although their accuracy and F1-score scores are slightly lower compared to DT and RF models, their competitive performance can be attributed to their unique strengths. This alignment with prior research reaffirms their efficacy in real-world applications, further establishing them as valuable tools for predictive tasks, even in the face of challenging data scenarios.

On the other hand, models like NB and SGDC consistently underperformed in terms of accuracy. NB (Li et al., 2022) models make strong assumptions about the independence of features, which might not hold in complex datasets, leading to suboptimal performance. SGDC relies on stochastic optimisation (Hasan et al., 2023), and its performance can vary depending on the dataset and hyper-parameter settings. The limitations of these models in capturing the complexities of the dataset could explain their lower accuracy & F1-score scores. NB and SGDC are prone to underperforming when faced with missing data and class imbalance. NB assumes feature independence according to Ray (2019) and Cohen et al. (2003), which becomes problematic when dealing with missing data, leading to inaccurate probability estimates and decision boundaries. Additionally, its reliance on imputation for missing values can introduce bias and noise (Cohen et al., 2003). The model's sensitivity to class imbalance further hampers its performance (Ray, 2019), as it might favour the majority class and overlook the minority class. Similarly, Karvouniaris et al. (2021) argue that SGDC's optimisation process can be disrupted by the absence of

certain features due to missing data, potentially causing slow convergence or suboptimal solutions. The model's inability to robustly handle missing data and its susceptibility to the dominant class in imbalanced datasets contribute to its subpar performance (Karvouniaris et al., 2021). In contrast to more adaptable models, NB and SGDC classifiers struggle to effectively address the challenges posed by missing data & class imbalance and might require some careful pre-processing of the data and tuning of the hyper-parameters.

When comparing the performance of SMOTE sampling and UNDER-sampling techniques, the general trend was that SMOTE sampling produced higher and more consistent accuracies across various models which was consistent with (Li et al., 2022) & (Hasan et al., 2023) findings. SMOTE oversamples the minority class by generating synthetic examples, which helps alleviate class imbalance and provides more training data for the models to learn from. This increased representation of the minority class contributes to better generalisation and higher accuracy & F1-score scores. In contrast, UNDER-sampling reduces the majority class by randomly removing instances, which can lead to a loss of information and potentially affect the models' ability to capture the underlying patterns in the data. However, despite the slight decrease in accuracy & F1-score compared to SMOTE sampling, the DT and RF models maintained their strong performance in both techniques, indicating their robustness and effectiveness in handling imbalanced datasets.

The work presented in this study makes significant contributions to the field of machine learning and predictive modelling. By delving into the performance of various machine learning models under different sampling techniques, this research sheds light on the intricate dynamics that influence model accuracy and F1-scores. The findings underscore the pivotal role of DT and RF models, which consistently outperformed other models across different scenarios. Their ability to capture complex relationships within datasets and generalise effectively to new data exemplifies their prowess in predictive tasks. Moreover, the study enriches the understanding of the performance nuances exhibited by LR, GB, XGB, and LGBM models. While these models demonstrated slightly lower accuracy and F1-scores compared to the leading DT and RF models, their competitive performance highlights their adaptability and robustness in handling diverse datasets. Drawing parallels with previous research, these findings echo the growing consensus on the efficacy of these models in predictive tasks.

9 Conclusion and Limitations

In conclusion, this analysis provided valuable insights into the performance of various machine learning models on a dataset with different levels of missing values, using different sampling techniques. The DT and RF models consistently stood out, achieving high accuracy & F1-score scores across both training and testing sets, regardless of the sampling technique employed. These models demonstrated their robustness in handling missing values and capturing intricate relationships between features. LR, GB, XGB, and LGBM models also showed competitive performance, showcasing their versatility in classification tasks.

However, NB and SGDC consistently underperformed compared to other models. These models exhibited limitations in capturing the complexities of the dataset in the presence

of missing values, resulting in lower accuracy & F1-score scores. Further investigation and improvement of these models, such as relaxing the independence assumptions in NB or optimising hyper-parameters in SGDC, could enhance their performance in similar tasks. The comparison between SMOTE sampling and UNDER-sampling techniques revealed that SMOTE sampling generally produced higher and more consistent accuracies across various models. SMOTE's ability to oversample the minority class and generate synthetic examples contributed to better generalisation and improved accuracy & F1-score. On the other hand, UNDER-sampling, by reducing the majority class, had a slight impact on the accuracy & F1-score of some models. However, the DT and RF models maintained their strong performance in both techniques, highlighting their effectiveness in handling missing values.

Future research could explore additional sampling techniques or alternative techniques to further enhance the performance of models on datasets with missing values. Furthermore, feature engineering techniques, dimensionality reduction methods, and model tuning could be employed to improve the overall performance of the models. Despite the valuable insights gained from this analysis, there are limitations to consider. The analysis did not explore the impact of different hyper-parameter settings on the models' performance and did not include the effect of the missing indicator. Optimising hyper-parameters could potentially lead to improved accuracy & F1-score scores for some models while the inclusion of a missing indicator has the potential to prevent biased imputations. Additionally, the analysis was based on a specific dataset with missing values, and the findings may not generalise to other datasets with different characteristics or imbalances. Conducting similar analyses on diverse datasets would provide a more comprehensive understanding of the models' capabilities and limitations.

This comparative analysis has yielded valuable insights into the performance of diverse machine learning models when handling missing values and employing different sampling techniques on the dataset. Notably, the decision tree and random forest models exhibited consistent excellence by achieving remarkable accuracy and F1-score scores across various scenarios. Their robustness in addressing missing values and capturing complex relationships underscores their reliability in real-world applications. The competitive performances of logistic regression, gradient boosting, extreme gradient boosting, and light gradient boosting machine models further emphasise their utility across classification tasks. The contrasting underperformance of Gaussian Naïve Bayes and stochastic gradient descent classifier serves as a vital reminder of the nuanced challenges posed by missing data. The findings signal potential areas for refinement, such as reevaluating assumptions in Gaussian naïve Bayes or optimising hyper-parameters in stochastic gradient descent classifier. These improvements could lead to more effective models in comparable scenarios. This work significantly benefits practitioners and decision-makers by offering a practical roadmap for selecting appropriate models and strategies to tackle missing data and class imbalance. The results highlight the pivotal role of decision trees and random forests as dependable options. The study's implications extend beyond academia, serving as a guide for professionals seeking accurate predictions and informed decision-making in scenarios where data quality is compromised. By shedding light on the strengths and weaknesses of various models, this analysis aids practitioners in making informed choices that align with their specific needs and resources, ultimately fostering more effective problem-solving and

strategic planning in real-world applications.

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.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Tanja Verster, for the unwavering 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.

References

Ahmed, A. (2021). Anti-money laundering recognition through the gradient boosting classifier. *Academy of Accounting and Financial Studies Journal*, 25(5).

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., Li, J., and Khushi, M. (2020). An investigation of credit card default prediction in the imbalanced datasets. *IEEE Access*, 8:201173–201198.

An, T.-K. and Kim, M.-H. (2010). A new diverse adaboost classifier. In *2010 International conference on artificial intelligence and computational intelligence*, volume 1, pages 359–363. IEEE.

Anand, M., Velu, A., and Whig, P. (2022). Prediction of loan behaviour with machine learning models for secure banking. *Journal of Computer Science and Engineering (JCSE)*, 3(1):1–13.

Aydın, Y., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S. M., and Geem, Z. W. (2023). Use of machine learning techniques in soil classification. *Sustainability*, 15(3):2374.

Bansal, A. and Kaur, S. (2018). Extreme gradient boosting based tuning for classification in intrusion detection systems. In *Advances in Computing and Data Sciences: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I 2*, pages 372–380. Springer.

Bittencourt, H. R., de Oliveira Moraes, D. A., and Haertel, V. (2007). A binary decision tree classifier implementing logistic regression as a feature selection and classification

method and its comparison with maximum likelihood. In *2007 IEEE international geoscience and remote sensing symposium*, pages 1755–1758. IEEE.

Breeden, J. (2021). A survey of machine learning in credit risk. *Journal of Credit Risk*, 17(3):1–62.

Chang, Y.-C., Chang, K.-H., Chu, H.-H., and Tong, L.-I. (2016). Establishing decision tree-based short-term default credit risk assessment models. *Communications in Statistics-Theory and Methods*, 45(23):6803–6815.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chicco, D. and 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.

Cohen, I., Sebe, N., Gozman, F., Cirelo, M. C., and Huang, T. S. (2003). Learning Bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE.

Dhieb, N., Ghazzai, H., Besbes, H., and Massoud, Y. (2019). Extreme gradient boosting machine learning algorithm for safe auto insurance operations. In *2019 IEEE international conference on vehicular electronics and safety (ICVES)*, pages 1–5. IEEE.

Du, G., Ma, L., Hu, J.-S., Zhang, J., Xiang, Y., Shao, D., and Wang, H. (2019). Prediction of 30-day readmission: an improved gradient boosting decision tree approach. *Journal of Medical Imaging and Health Informatics*, 9(3):620–627.

Du, W. and Zhan, Z. (2002). Building decision tree classifier on private data. *Electrical Engineering and Computer Science*, 8.

Dube, L. and Verster, T. (2023). Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. *Data Science in Finance and Economics*, 3(4):354–379.

Ferreira, L. E. B., Barddal, J. P., Gomes, H. M., and Enembreck, F. (2017). Improving credit risk prediction in online peer-to-peer (p2p) lending using imbalanced learning techniques. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 175–181. IEEE.

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

Hasan, M., Zobair, M. J., Akter, S., Ashef, M., Akter, N., and Sadia, N. B. (2023). Ensemble based machine learning model for early detection of mother's delivery mode. In *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. IEEE.

He, F., Zhang, W., and Yan, Z. (2022). A novel multi-stage ensemble model for credit scoring based on synthetic sampling and feature transformation. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–16.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE.

Hu, W., Hu, W., and Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):577–583.

Islam, M. J., Wu, Q. J., Ahmadi, M., and Sid-Ahmed, M. A. (2007). Investigating the performance of naive-Bayes classifiers and k-nearest neighbor classifiers. In *2007 international conference on convergence information technology (ICCIT 2007)*, pages 1541–1546. IEEE.

Kabir, F., Siddique, S., Kotwal, M. R. A., and Huda, M. N. (2015). Bangla text document categorization using stochastic gradient descent (sgd) classifier. In *2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, pages 1–4. IEEE.

Kaggle (2023). Give me some credit. https://www.kaggle.com/competitions/givemesomecredit/data?select=cs-training.csv. Accessed: 2023-02-05.

Karvouniaris, M., Pontikis, K., Nitsotolis, T., and Poulakou, G. (2021). New perspectives in the antibiotic treatment of mechanically ventilated patients with infections from gram-negatives. *Expert Review of Anti-Infective Therapy*, 19(7):825–844.

Khafajeh, H. (2020). An efficient intrusion detection approach using light gradient boosting. *Journal of Theoretical and Applied Information Technology*, 98(5):825–835.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30.

Leung, K. M. et al. (2007). Naive Bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007:123–156.

Li, X., Ergu, D., Zhang, D., Qiu, D., Cai, Y., and Ma, B. (2022). Prediction of loan default based on multi-model fusion. *Procedia Computer Science*, 199:757–764.

Liu, A. Y.-C. (2004). *The effect of oversampling and undersampling on classifying imbalanced text datasets*. PhD thesis, Citeseer.

Liu, Y., Wang, Y., and Zhang, J. (2012). New machine learning algorithm: Random forest. In *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*, pages 246–252. Springer.

Machado, M. R., Karray, S., and de Sousa, I. T. (2019). Lightgbm: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry. In *2019 14th International Conference on Computer Science & Education (ICCSE)*, pages 1111–1116. IEEE.

Madaan, M., Kumar, A., Keshri, C., Jain, R., and Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. In *IOP Conference Series: Materials Science and Engineering*, volume 1022, page 012042. IOP Publishing.

Mahajan, S., Nayyar, A., Raina, A., Singh, S. J., Vashishtha, A., and Pandit, A. K. (2022). A Gaussian process-based approach toward credit risk modeling using stationary activations. *Concurrency and Computation: Practice and Experience*, 34(5):e6692.

Mitchell, T. M. and Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York.

Murphy, K. P. et al. (2006). Naive Bayes classifiers. *University of British Columbia*, 18(60):1–8.

Osho, O. and Hong, S. (2021). An overview: Stochastic gradient descent classifier, linear discriminant analysis, deep learning and naive Bayes classifier approaches to network intrusion detection. *International Journal of Engineering and Technical Research*, 10(4):294–308.

Padimi, V., Venkata, S., and Devarani, D. (2022). Applying machine learning techniques to maximize the performance of loan default prediction. *Journal of Neutrosophic and Fuzzy Systems (JNFS)*, 2(2):44–56.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.

Patro, S. and Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Ray, S. (2019). A quick review of machine learning algorithms. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 35–39. IEEE.

Rokach, L. (2016). Decision forest: Twenty years of research. *Information Fusion*, 27:111–125.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Schafer, J. L. and Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, 7(2):147.

Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68:235–265.

Swain, P. H. and Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147.

Taha, A. A. and Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. *IEEE Access*, 8:25579–25587.

Wang, K., Wan, J., Li, G., and Sun, H. (2022). A hybrid algorithm-level ensemble model for imbalanced credit default prediction in the energy industry. *Energies*, 15(14):5206.

Xu, Z., Huang, G., Weinberger, K. Q., and Zheng, A. X. (2014). Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 522–531.

Yigit, H. (2013). A weighting approach for knn classifier. In *2013 international conference on electronics, computer and computation (ICECCO)*, pages 228–231. IEEE.

Zhou, J., Li, W., Wang, J., Ding, S., and Xia, C. (2019). Default prediction in p2p lending from high-dimensional data based on machine learning. *Physica A: Statistical Mechanics and its Applications*, 534:122370.

10 Appendix

This section displays additional results obtained ADASYN and OVER sampling techniques. Additionally, it also contains more ROC curves that were not presented in the results section.

10.1 ADASYN and OVER-sampling results by accuracy

Table 5 displays training and testing results for ADASYN and OVER sampling results as summarised by accuracy score.

10.2 ADASYN and OVER-sampling results by F1-score

Table 6 displays ADASYN and OVER sampling results as summarised by the F1-score.

10.3 ROC curves

Displayed below are the ROC curves for the testing results of SMOTE (Figure 4) and UNDER sampling (Figure 5).

Table 5: ADASYN and OVER-sampling results by accuracy

		Model	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample7	Sample 8	Sample 9	Sample 10
Training Sets	ADASYN Sampling	**LR**	0.7428	0.7361	0.7337	0.7267	0.7203	0.7153	0.7088	0.6910	0.6829	0.6783
		DT	**0.9999**	**0.9999**	**0.9999**	**0.9998**	**0.9997**	**0.9993**	**0.9983**	**0.9965**	**0.9943**	**0.9879**
		RF	**0.9998**	**0.9999**	**0.9998**	**0.9998**	**0.9997**	**0.9992**	**0.9983**	**0.9965**	**0.9942**	**0.9879**
		ADA	0.9325	0.9322	0.9281	0.9253	0.9245	0.9149	0.9069	0.8914	0.8785	0.8581
		GB	0.9481	0.9473	0.9458	0.9448	0.9393	0.9339	0.9279	0.9155	0.8999	0.8855
		XGB	0.9699	0.9698	0.9695	0.9686	0.9681	0.9648	0.9612	0.9530	0.9457	0.9323
		LGBM	0.9634	0.9631	0.9624	0.9618	0.9609	0.9572	0.9536	0.9446	0.9343	0.9203
		SGDC	0.7526	0.7378	0.7442	0.7308	0.7182	0.7117	0.6961	0.7023	0.6771	0.6820
		KNN	0.9156	0.9175	0.9163	0.9156	0.9149	0.9139	0.9136	0.9111	0.9042	0.8962
		NB	0.6352	0.6272	0.6459	0.6459	0.5878	0.6453	0.6223	0.5974	0.6254	0.5364
	OVER-Sampling	**LR**	0.7411	0.7403	0.7341	0.7272	0.7232	0.7090	0.6991	0.6981	0.6871	0.6810
		DT	**0.9998**	**0.9999**	**0.9999**	**0.9998**	**0.9997**	**0.9990**	**0.9985**	**0.9967**	**0.9941**	**0.9880**
		RF	**0.9998**	**0.9999**	**0.9998**	**0.9998**	**0.9997**	**0.9990**	**0.9985**	**0.9966**	**0.9941**	**0.9880**
		ADA	0.9338	0.9342	0.9296	0.9290	0.9226	0.9184	0.9076	0.8948	0.8761	0.8565
		GB	0.9499	0.9485	0.9463	0.9464	0.9394	0.9346	0.9294	0.9149	0.9020	0.8830
		XGB	0.9705	0.9693	0.9693	0.9690	0.9670	0.9642	0.9618	0.9553	0.9449	0.9329
		LGBM	0.9641	0.9626	0.9628	0.9624	0.9608	0.9572	0.9539	0.9446	0.9316	0.9200
		SGDC	0.7510	0.7461	0.7372	0.7339	0.7261	0.7041	0.6923	0.6899	0.6680	0.6668
		KNN	0.9178	0.9169	0.9170	0.9155	0.9140	0.9122	0.9134	0.9119	0.9074	0.9012
		NB	0.6246	0.5721	0.5805	0.5291	0.5241	0.5528	0.5567	0.5417	0.5736	0.6060
Testing Sets	ADASYN Sampling	**LR**	0.7435	0.7362	0.7359	0.7297	0.7231	0.7161	0.7069	0.6932	0.6853	0.6766
		DT	**0.9348**	**0.9320**	**0.9332**	**0.9291**	**0.9286**	**0.9228**	**0.9218**	**0.9142**	**0.9064**	**0.8939**
		RF	**0.9597**	**0.9606**	**0.9585**	**0.9554**	**0.9575**	**0.9528**	**0.9488**	**0.9439**	**0.9330**	**0.9220**
		ADA	0.9325	0.9347	0.9279	0.9253	0.9255	0.9150	0.9065	0.8908	0.8767	0.8590
		GB	**0.9480**	**0.9498**	**0.9463**	**0.9432**	**0.9397**	**0.9340**	**0.9271**	**0.9151**	**0.8993**	**0.8837**
		XGB	**0.9648**	**0.9649**	**0.9642**	**0.9617**	**0.9620**	**0.9594**	**0.9549**	**0.9471**	**0.9394**	**0.9245**
		LGBM	**0.9630**	**0.9635**	**0.9627**	**0.9592**	**0.9589**	**0.9557**	**0.9525**	**0.9432**	**0.9333**	**0.9168**
		SGDC	0.7531	0.7391	0.7460	0.7344	0.7204	0.7141	0.6945	0.7019	0.6791	0.6815
		KNN	0.8734	0.8725	0.8706	0.8723	0.8715	0.8723	0.8706	0.8698	0.8602	0.8510
		NB	0.6336	0.6235	0.6484	0.6446	0.5900	0.6477	0.6228	0.5988	0.6271	0.5361
	OVER-Sampling	**LR**	0.7382	0.7380	0.7299	0.7260	0.7220	0.7088	0.6983	0.6975	0.6829	0.6755
		DT	**0.9366**	**0.9335**	**0.9295**	**0.9343**	**0.9309**	**0.9286**	**0.9221**	**0.9111**	**0.9067**	**0.8934**
		RF	**0.9591**	**0.9594**	**0.9569**	**0.9577**	**0.9549**	**0.9546**	**0.9505**	**0.9427**	**0.9359**	**0.9227**
		ADA	0.9326	0.9328	0.9294	0.9302	0.9212	0.9178	0.9084	0.8944	0.8724	0.8564
		GB	**0.9489**	**0.9474**	**0.9459**	**0.9443**	**0.9381**	**0.9344**	**0.9285**	**0.9162**	**0.9006**	**0.8821**
		XGB	**0.9644**	**0.9642**	**0.9631**	**0.9620**	**0.9614**	**0.9591**	**0.9560**	**0.9495**	**0.9389**	**0.9286**
		LGBM	**0.9633**	**0.9622**	**0.9620**	**0.9608**	**0.9593**	**0.9560**	**0.9522**	**0.9435**	**0.9306**	**0.9197**
		SGDC	0.7493	0.7448	0.7335	0.7329	0.7281	0.7025	0.6921	0.6912	0.6630	0.6626
		KNN	0.8728	0.8726	0.8701	0.8697	0.8717	0.8662	0.8720	0.8692	0.8628	0.8616
		NB	0.6230	0.5705	0.5774	0.5266	0.5228	0.5503	0.5556	0.5407	0.5728	0.6052

Table 6: ADASYN and OVER-sampling results by F1-score

	Model	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample7	Sample 8	Sample 9	Sample 10
ADASYN Sampling	**LR**	0.7270	0.7180	0.7157	0.7084	0.7025	0.6923	0.6764	0.6578	0.6414	0.6268
	DT	**0.9355**	**0.9326**	**0.9334**	**0.9293**	**0.9289**	**0.9231**	**0.9220**	**0.9140**	**0.9052**	**0.8921**
	RF	**0.9591**	**0.9599**	**0.9575**	**0.9543**	**0.9565**	**0.9516**	**0.9475**	**0.9423**	**0.9304**	**0.9192**
	ADA	0.9313	0.9332	0.9260	0.9232	0.9228	0.9117	0.9020	0.8850	0.8681	0.8462
	GB	0.9470	0.9484	0.9447	0.9413	0.9374	0.9311	0.9234	0.9100	0.8913	0.8738
	XGB	0.9642	0.9641	0.9633	0.9607	0.9609	0.9581	0.9534	0.9448	0.9358	0.9194
	LGBM	0.9624	0.9626	0.9616	0.9580	0.9576	0.9542	0.9507	0.9406	0.9291	0.9108
	SGDC	0.7086	0.6906	0.7040	0.6871	0.6648	0.6486	0.6223	0.6279	0.5794	0.5829
	KNN	0.8693	0.8670	0.8648	0.8666	0.8661	0.8664	0.8645	0.8636	0.8523	0.8436
	NB	0.5048	0.4549	0.5511	0.5763	0.3308	0.5717	0.6118	0.6433	0.5052	0.1353
OVER-Sampling	**LR**	0.7194	0.7184	0.7115	0.7039	0.7013	0.6848	0.6684	0.6659	0.6465	0.6355
	DT	**0.9369**	**0.9340**	**0.9299**	**0.9347**	**0.9313**	**0.9291**	**0.9225**	**0.9114**	**0.9069**	**0.8930**
	RF	**0.9584**	**0.9586**	**0.9560**	**0.9568**	**0.9540**	**0.9536**	**0.9494**	**0.9414**	**0.9344**	**0.9209**
	ADA	0.9309	0.9312	0.9276	0.9284	0.9185	0.9141	0.9045	0.8896	0.8647	0.8478
	GB	**0.9476**	**0.9460**	**0.9444**	**0.9426**	**0.9359**	**0.9317**	**0.9253**	**0.9118**	**0.8944**	**0.8736**
	XGB	**0.9636**	**0.9633**	**0.9622**	**0.9611**	**0.9604**	**0.9580**	**0.9547**	**0.9476**	**0.9362**	**0.9252**
	LGBM	**0.9624**	**0.9612**	**0.9611**	**0.9598**	**0.9581**	**0.9547**	**0.9506**	**0.9412**	**0.9271**	**0.9155**
	SGDC	0.6985	0.6984	0.6790	0.6811	0.6788	0.6276	0.6071	0.8692	0.5545	0.5462
	KNN	0.8675	0.8674	0.8642	0.8642	0.8660	0.8604	0.8661	0.8640	0.8568	0.8558
	NB	0.4602	0.2637	0.2891	0.1119	0.0971	0.1975	0.2219	0.1661	0.2849	0.5069

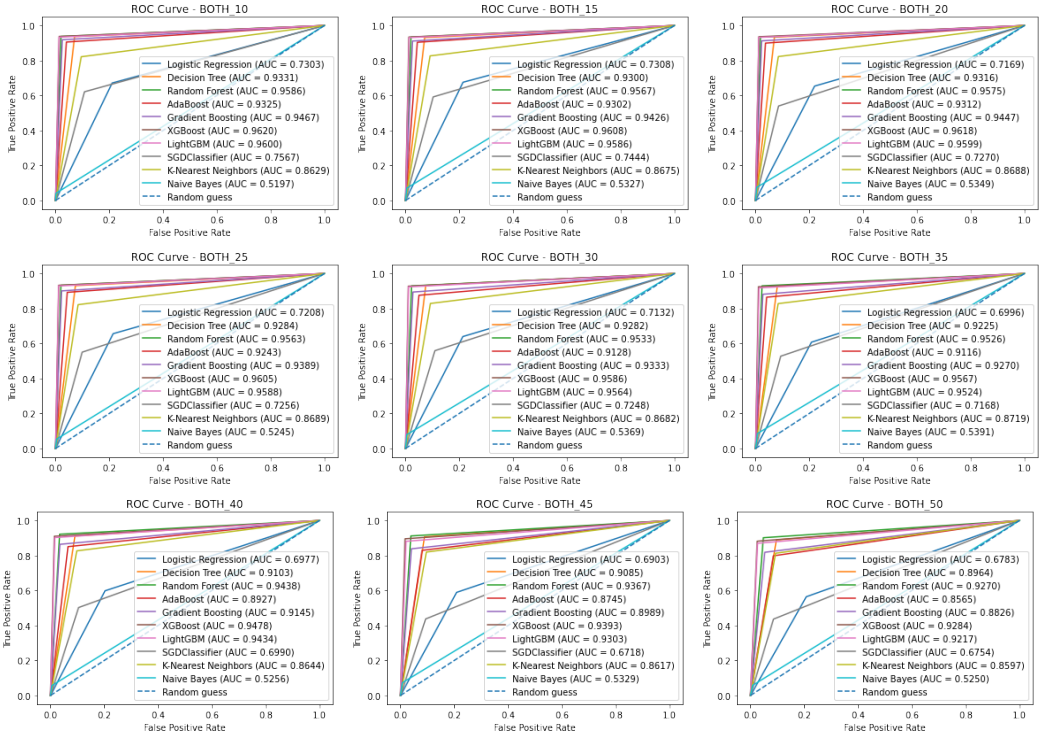

Figure 4: SMOTE-sampling ROC curve results

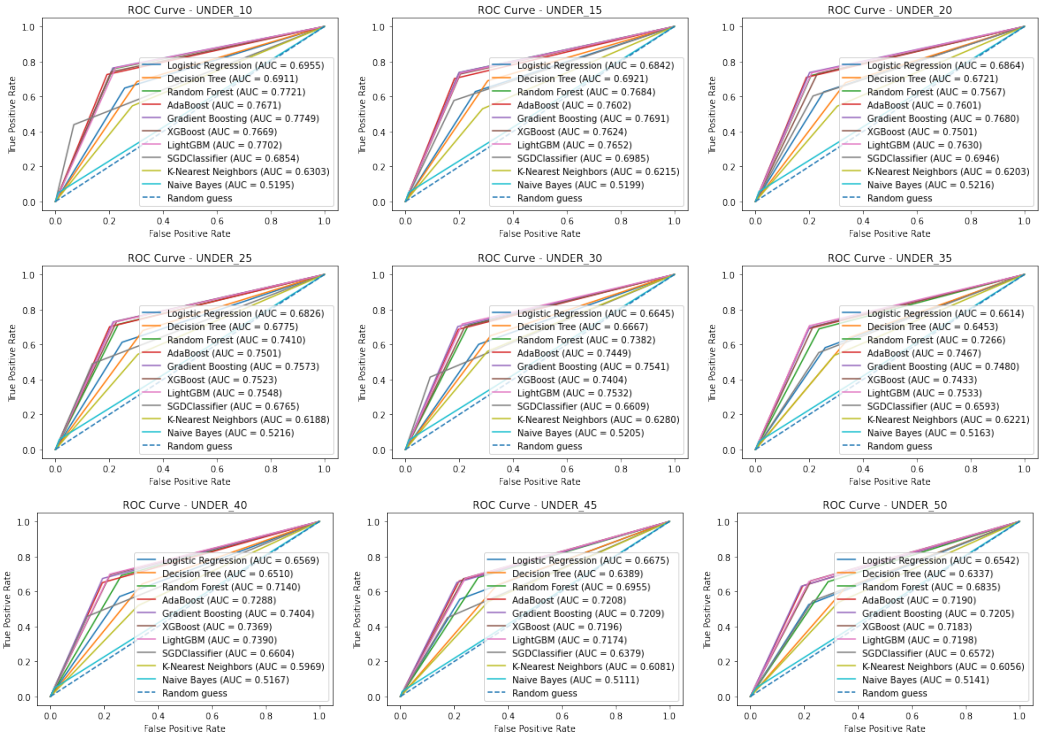

Figure 5: UNDER-sampling ROC curve results