A Project Report

on

Statistical Analysis of Imbalanced Classification with Training Size Variation and Subsampling on Datasets of Research Papers in Biomedical Literature

College of Computing and Mathematic

جامعة الملك فهد للبترول والمعادن
King Fahd University of Petroleum & Minerals

Submitted By

Abd-r-Rahman

Jerry Joel

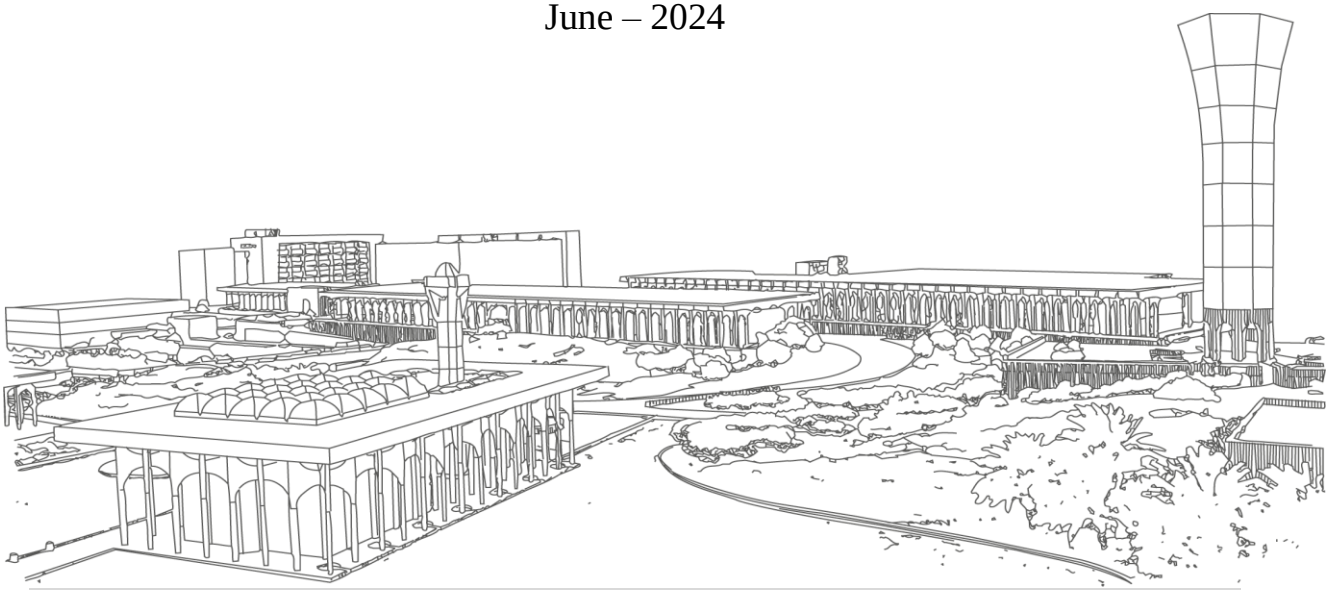
Submitted To

Dr. Jimoh Ajadi

College of Computing and Mathematic

Department of Mathematics, King Fahd University of Petrolium & Minerals, Dhahran

June – 2024



INTRODUCTION

Machine learning often faces the challenges of imbalanced classification, specifically when working with datasets that have much lower representation of certain classes than others. This challenge is common in divers fields of learning, such as biomedical writing, where the quantity of research articles addressing uncommon illnesses or ailments is sometimes outnumbered by those concentrating on more typical subjects. Biased models that perform well in the majority class but badly in the minority class, which is sometimes of more interest in research contexts, might result from such imbalances.

Customized methods are necessary to address the class imbalance to guarantee reliable and equitable model performance. Subsampling and different training sizes can have a big impact on how successful classification models behave. Using datasets from biomedical literature, this study investigates how different methods perform on tasks involving unbalanced classification. The goal of the research is to ascertain the best practices for managing unbalanced datasets in the context of biomedical research publications by carefully looking into the effects of various training sizes and subsampling techniques on model performance.

Text classification and unbalanced classification procedures have a rich and unique research landscape, with many articles providing information on the effectiveness of different sampling strategies, machine learning algorithms, and preprocessing techniques. Goudjil et al., (2018). present an active learning approach that reduces labeling work while preserving classification accuracy, Kadhim's study assesses the effect of text preprocessing tools on the categorization of English text and shows improvements in feature extraction technologies.

Mali et al., (2021) research on how preprocessing stages affect text classification, specifically in unstructured data, and find that different classifiers perform more accurately because of their efforts. Other studies exhibit the intricate relationship between data preparation, model selection, and classification accuracy. These works explore subsampling tactics, training size modifications, and the performance of various classifiers on raw and cleaned datasets. Research endeavors also include statistical approaches for evaluating performance, such as ANOVA, Student's t-test, and exploratory data analysis methods. This emphasizes the significance of strong statistical analysis in interpreting classification outcomes and directing future research paths.

Subsampling Techniques

Subsampling techniques are crucial strategies for addressing class imbalance in datasets, particularly in the context of classification tasks in machine learning. These techniques involve modifying the dataset to balance the representation of different classes, thereby improving the performance of classifiers. Here, we discuss various subsampling techniques used to mitigate the effects of class imbalance.

1. Random Under sampling

Random under sampling involves randomly removing instances from the majority class until the class distribution is balanced. This technique reduces the number of majority class samples to match the number of minority class samples.

2. Random Oversampling

Random oversampling involves randomly duplicating instances from the minority class until the class distribution is balanced. This technique increases the number of minority class samples to match the number of majority class samples.

3. SMOTE (Synthetic Minority Over-sampling Technique)

SMOTE generates synthetic samples for the minority class by interpolating between existing minority class instances. New samples are created along the line segments joining each minority class instance and its k-nearest neighbors.

4. Adaptive Synthetic Sampling (ADASYN)

ADASYN is an extension of SMOTE that focuses on generating synthetic samples for minority class instances that are harder to classify. It adaptively creates more synthetic data in regions where the minority class is underrepresented.

5. Cluster-Based Oversampling

Cluster-based oversampling involves clustering the minority class instances and then applying oversampling techniques within each cluster. This approach ensures that synthetic samples are generated based on local data distributions.

6. Ensemble Methods

Ensemble methods combine multiple classifiers trained on different balanced subsets of the data. Techniques such as Balanced Random Forests and Easy Ensemble are popular ensemble methods used to handle class imbalance.

Methodology

The dataset comprises 1000 PDF documents categorized into five machine learning labels: Immune, Problems in China, Risk Factors, Transmission, and Testing. Twenty-five percent of the documents are sourced from the World Health Organization (WHO) COVID-19 Downloadable Articles Database, representing the positive class, while the remaining 75% are non-related COVID-19 research papers from the PubMed Central database, representing the negative class. Conversion of PDF documents into text files using Python libraries Scikit-learn, NumPy, and Pandas facilitates feature engineering and text classification. The methodology involves two approaches: one utilizing combined text files for each label and another automatically annotating keywords based on regular expressions, with the positive class

documents processed for automatic labelling. Ultimately, the annotated CSV files are merged to form Train, Dev, and Test Subsets, facilitating comprehensive analysis and classification of the dataset.

Classification Tasks

The method uses five machine learning classifiers:

Decision Tree

A decision tree is a non-parametric supervised learning algorithm for classification and regression tasks. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression tasks, providing easy-to-understand models. A decision tree (DT) is structured with leaves at the base, resulting in an inverted tree-like appearance . Its components include:

• **Target or Class Variable:** This represents the outcome being predicted
• **Root Nodes**: The starting point of the splitting process, determined by the variable that best divides the class variable.
• **Branches:** Connect the nodes together, forming the paths through the tree.
• **Pure/Impure Nodes**: Nodes are considered pure if they contain only one class, and impure if they have a mix of classes.
• **Decision Nodes:** Impure nodes that necessitate further splitting based on a different variable.
• **Leaf Nodes (Terminal Nodes):** These are pure nodes, signifying the end of a branch where predictions are made.

Random Forest

Since Decision trees are highly sensitive to the training data, changing the training data slightly will yield a different result in the leaf node. RF algorithm relies on a combination of multiple decision trees and this helps in reducing bias, tolerate outliers, avoid overfitting and it is much less sensitive to the training data .In RF, each tree is grown using a bootstrap sample of the original data.Rather than splitting a tree node using all variables, RF selects at each node of each tree, a random subset of variables and only those variables are used as suitors to find the best split for the node.This two-step randomization will ensure the trees are not correlated and hence low variance, a bagging phenomenon.

XGBoost

XGBoost is a widely adopted supervised machine learning technique known for its high effciency and effectiveness. It is based on ensemble trees and utilizes a scalable gradient boosting algorithm . The approach of boosting is employed, where the predictions from a group of ”weak” learners are combined to create a powerful ”strong” learner through additive training techniques.XGBoost is designed to maximize a cost objective function, which consists of a regularization term and a loss function. Additionally, it incorporates a shrinkage hyper-parameter that controls the step size of the additive expansion, helping to mitigate overfitting.

Naïve Bayes

Naive Bayes classifier is a probabilistic machine learning model based on Bayes’ theorem.
It assumes independence between features and calculates the probability of a given input belonging to a particular class. it is worth noting that naive Bayes classifiers are among the

simplest Bayesian network models, yet they can achieve high accuracy levels when coupled with kernel density estimation.
This technique involves using a kernel function to estimate the probability density function of the input data, allowing the classifier to improve its performance in complex scenarios where the data distribution is not well-defined. One of drawbacks of this technique: If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". Another limitation of this algorithm is the assumption of independent predictors.

Logistic regression

Logistic regression uses Maximum likelihood to estimate the unknown logistic regression coefficients , where the estimates βˆ0 and βˆ1 are chosen to maximize this likelihood function. Consider N samples with labels either 0 or 1 For samples labeled as '1', we try to estimate β such that the product of all probability p(x) is as close to 1 as possible.
For samples labeled as '0', we try to estimate β such that the product of all probability is as close to 0 as possible in other words (1 — p(x)) should be as close to 1 as possible.
On combining the above conditions, we want to find β parameters such that the product.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

Subsampling and Training Size Variation

Subsampling plays a crucial role in enhancing the performance of classifiers, particularly in handling imbalanced datasets. The process involves selecting a specific number of samples based on predefined intervals, typically ranging from 5% to 100%. Two approaches are commonly employed, each with varying subsample sizes and iterations. In the first approach, performance fluctuates as subsampling progresses, with iterations 1 to 13 showcasing performance variations across classifiers and sampling techniques. However, at iteration 13, optimal or suboptimal performance is achieved depending on the specific technique and classifier used. Conversely, the second approach demonstrates significant performance improvements with increasing subsample sizes, particularly from 10% to nearly 100%, with a notable enhancement in classifier performance. Notably, a subsample size of 50% consistently yields the poorest performance across all sampling techniques and classifiers. This underscores the importance of carefully selecting subsample sizes to optimize classifier performance in handling imbalanced datasets.

Table 3. Labeling using Keyword Matching via Regular Expressions.

	Sentence	Label	Data	Regex
0	Research Letters	0	N/A	False
1	Table 1. Clinical characteristics of 604 patients with systemic lupus erythematosus with and without COVID-19 a Mean (standard deviation).	1	Clinical	True
2	SLE patients with COVID-19 reported a lower frequency of social isolation.	1	COVID-19	True
3	Secondary AA can be caused by infections, drugs, or various diseases.	1	Disease	True

Table 4. Number of Samples for Each Label Based on Subsampling.

Label	Percentage	Iteration	Number of Samples
Immune	5%	1	2336
Problems in China	10%	2	4715
Risk Factors	40%	8	9466
Testing	95%	19	94,398
Transmission	20%	4	17,580
Immune	25%	5	11,681
Transmission	30%	6	39,556

Statistical Analysis

The R programming language offers a comprehensive suite of packages for statistical analysis, with a particular emphasis on the principles of tidy data. Central to this approach is the tidy verse collection, which includes essential packages like dplyr, ggplot2, tidyr, readr, purrr, and Tibble. Leveraging these tools, statisticians can conduct exploratory data analysis efficiently, with tibble providing a more readable alternative to traditional data frames and ggplot2 facilitating the creation of diverse visualizations through a graphics grammar framework. Beyond data manipulation and visualization, key functions within packages like stats, rstatix, and ggpubr empower users to perform hypothesis testing, generate summary statistics, and produce visual representations of their analyses, such as box plots. Techniques like pairwise t-tests and ANOVA, supported by functions like pairwise_t_test and anova_test, enable rigorous evaluation of hypotheses, while tools like Bonferroni correction help mitigate the risks associated with multiple comparisons. Together, these packages and functions constitute a robust toolkit for statisticians to explore, analyse, and visualize data effectively within the R environment.

Experiment and Results

script is employed to organize data into structured CSV files, detailing key parameters such as labels, sampling methods, classifiers, and performance metrics like precision and recall.

Various sampling techniques, including SMOTE and RUS, are utilized to address imbalanced datasets, ensuring a more representative training set. The Scikit-learn library's modules, such as train test split, facilitate efficient data partitioning for model training and evaluation. Iteratively adjusting train split sizes allows for comprehensive exploration of different training/testing combinations, while fixed sizes provide a standardized approach for comparison. By recording evaluation metrics in CSV files for each model and iteration, the process enables thorough analysis and comparison of results, ultimately contributing to the development of robust machine learning models.

Results from MEDFULL Data

The MEDFULL data collection involves five labels: Immune, Problems in China, Risk Factors, Testing, and Transmission. This initial approach employs data preprocessing techniques and manually assigns documents to positive and negative classes based on content. Our experiment has shown that there is no issue with class imbalance.

Demonstrates the highest precision at 100% from an iteration of 13 or a 65% subsampling size. The precision scores range from 99.6% to 100%, while recall scores vary from 55% to 85%. This indicates that the optimal subsampling size is 65%, or iteration 13, when the test split size is 67% or 83%, and the train split size is between 17% and 33%. This configuration provides sufficient data to achieve the highest performance from the classifiers and imbalanced sampling techniques.

Highest Performance Metrics from COVID Iterations.

Label	Technique [1]	Classifier [1]	Test Split Size	Train Split Size	Iteration	Precision	Recall	AUROC	Accuracy
Immune	NM	LR	0.67	0.33	13	1	0.557233	0.778617	0.602797
Problems in China	NM	RF	0.67	0.33	13	1	0.618273	0.809136	0.691296
Immune	NM	NB	0.67	0.33	13	0.999266	0.584622	0.791048	0.64464
Risk Factors	NM	XG	0.67	0.33	13	0.998252	0.620652	0.808094	0.69405
Problems in China	NM	DT	0.67	0.33	13	0.996917	0.578932	0.78391	0.635826

[1] The technique used is only relevant to Undersampling techniques. Full words for abbreviations: NM—NearMiss, LR—Logistic Regression, RF—Random Forest, NB—Naïve Bayes, XG—XGBoost, DT—Decision Tree.

Presents the minimum precision and recall scores observed during an iteration with a 13 or 65% subsampling size, where the test and train split sizes are both 50%. The findings indicate that iteration 13 yields inconsistent results.

Lowest Performance Metrics from COVID Iterations.

Label	Technique [1]	Classifier [1]	Test Split Size	Train Split Size	Iteration	Precision	Recall	AUROC	Accuracy
Testing	NM	LR	0.5	0.5	13	0.093478	0.014517	0.436867	0.436867
Testing	NM	XG	0.5	0.5	13	0.120879	0.00945	0.470361	0.470361
Testing	NM	NB	0.5	0.5	13	0.139394	0.013046	0.466251	0.466251
Transmission	NM	DT	0.5	0.5	13	0.188406	0.02054	0.466834	0.466834
Testing	NM	RF	0.5	0.5	13	0.142857	0.01	0.47495	0.474562

[1] The technique used is only relevant to Undersampling techniques. Full words for abbreviations: NM—NearMiss, LR—Logistic Regression, RF—Random Forest, NB—Naïve Bayes, XG—XGBoost, DT—Decision Tree.

Presents the average scores for each label, independent of the sampling method, technique, classifier, train split size, test split size, iteration, and subsampling size. Once again, the Risk Factors label yields the highest scores, whereas the Testing label results in the lowest scores.

Average Performance Metrics Based on Label for MEDFULL Data.

Label	Precision	Recall	AUROC	Accuracy
Immune	0.73670	0.70131	0.67559	0.66176
Problems in China	0.66540	0.63147	0.67900	0.66442
Risk Factors	0.73396	0.72982	0.68216	0.67627
Testing	0.72046	0.68546	0.67397	0.65945
Transmission	0.75239	0.69499	0.67528	0.66179

Presents the average precision, recall, AUROC, and accuracy scores for the sampling method, independent of the classifier, sampling technique, label, train split size, test split size, iteration, and subsampling size. The results indicate that imbalanced data outperforms other sampling methods such as oversampling and undersampling.

Provides the average precision, recall, AUROC, and accuracy scores for the sampling technique, regardless of the sampling method, label, classifier, train split size, test split size, iteration, and subsampling size. Machine learning models using imbalanced sampling techniques show slightly better performance compared to running models on the imbalanced dataset alone. Among the imbalanced sampling techniques, Tomek Links achieves the best scores, while Near Miss has the worst scores.

Table 10. Average Performance Metrics Based on Sampling for MEDFULL Data.

Sampling	Precision	Recall	AUROC	Accuracy
Imbalanced	0.73295	0.69637	0.66433	0.66619
Oversampling	0.71301	0.68496	0.68092	0.66251
Undersampling	0.68846	0.68846	0.67902	0.66574

Table 11. Average Performance Metrics Based on Technique for MEDFULL Data.

Technique	Precision	Recall	AUROC	Accuracy
Imbalanced	0.73295	0.69636	0.66432	0.66618
NearMiss	0.71225	0.68521	0.67541	0.65667
ROS	0.71283	0.68693	0.68151	0.66313
RUS	0.72433	0.68984	0.68834	0.66945
SMOTE	0.71319	0.68299	0.68026	0.66189
TomekLinks	0.73521	0.69033	0.67333	0.67109

Shows the ANOVA t-test results for classifiers based on precision. The p-values indicate the statistical significance of accuracy variations between algorithm pairings. Significant differences (****) in accuracy are observed between Decision Tree and Naive Bayes, Decision Tree and Random Forest, Decision Tree and Random Forest, Logistic Regression and XGBoost, and Logistic Regression and XGBoost. The adjusted p-values confirm that these significant differences remain even after correcting for multiple comparisons. Conversely, several comparisons, such as Random Forest with XGBoost, Naive Bayes with Random Forest, and Random Forest with XGBoost, have p-values above the standard cutoff of 0.05, indicating no significant changes.

Metric	Group 1 [1]	Group 2 [1]	p	p.signif [1]	p.adj [1]	p.adj.signif [1]
Precision	DT	LR	1.09×10^{-3}	**	1.09×10^{-2}	*
Precision	DT	NB	4.26×10^{-2}	*	4.26×10^{-1}	ns
Precision	LR	NB	1.21×10^{-7}	****	1.21×10^{-6}	****
Precision	DT	RF	2.89×10^{-5}	****	2.89×10^{-4}	***
Precision	LR	RF	9.85×10^{-14}	****	9.85×10^{-13}	****
Precision	NB	RF	3.11×10^{-2}	*	3.11×10^{-1}	ns
Precision	DT	XG	4.44×10^{-4}	***	4.44×10^{-3}	**
Precision	LR	XG	1.25×10^{-11}	****	1.25×10^{-10}	****
Precision	NB	XG	0.137	ns	1.00	ns
Precision	RF	XG	0.503	ns	1.00	ns

[1] The classifiers are compared based on 3000 observations. Full words for abbreviations: LR—Logistic Regression, RF—Random Forest, NB—Naïve Bayes, XG—XGBoost, DT—Decision Tree, ns—non-significance level. ns means $p > 0.05$, * means $p \leq 0.05$, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$. p.signf means the significance level of p-values, p.adj means adjusted p-value, and p.adj.signif means the significance level of the adjusted p-value.

The ANOVA t-test results for classifiers based on recall are presented in Table 13. The p-values indicate the statistical significance of differences in recall between pairs of algorithms. Several key findings are observed: significant differences (****) are noted in comparisons between Decision Tree and Logistic Regression, Decision Tree and Naive Bayes, Logistic Regression and Random Forest, Decision Tree and Random Forest, Logistic Regression and XGBoost, and Logistic Regression and XGBoost. These results remain significant even after adjusting for multiple comparisons, as indicated by the adjusted p-values. Conversely, some comparisons, such as Naive Bayes with Random Forest and Naive Bayes with XGBoost, show no significant differences, as evidenced by p-values above the conventional threshold of 0.05.

T-test Statistics of Recall Based on Classifiers for MEDFULL Data.

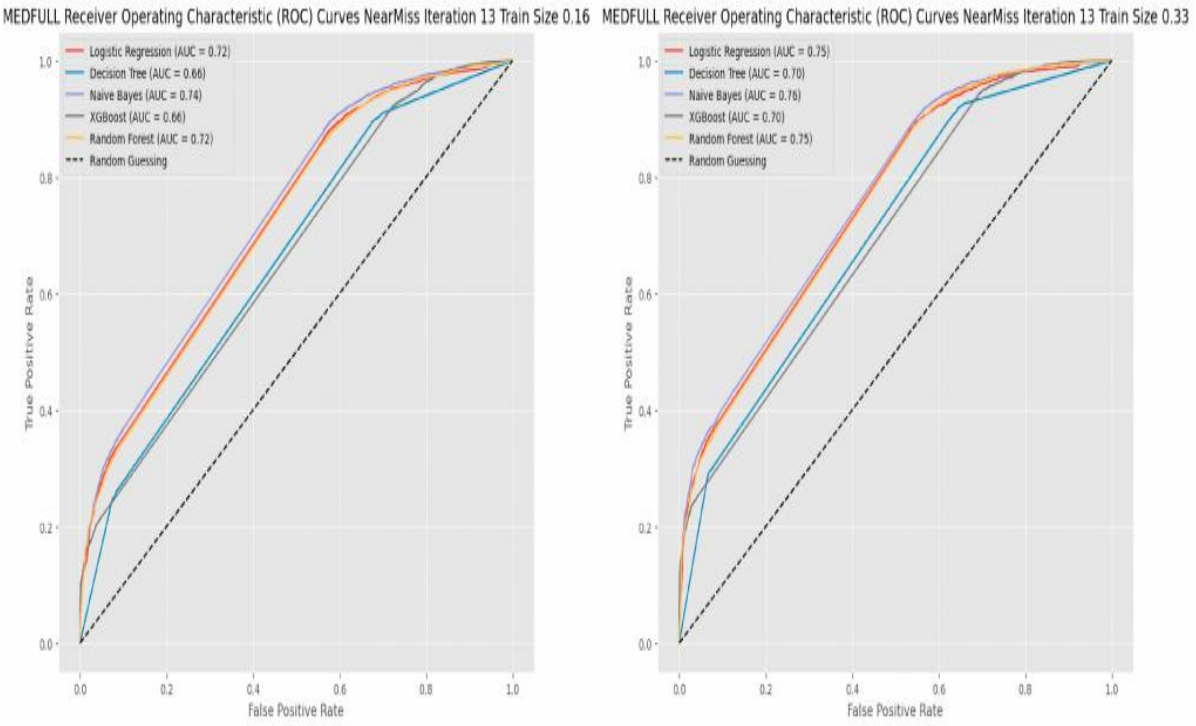
Metric	Group 1 [1]	Group 2 [1]	p	p.signif [1]	p.adj [1]	p.adj.signif [1]
Recall	DT	LR	6.33×10^{-8}	****	6.33×10^{-7}	****
Recall	DT	NB	3.65×10^{-1}	ns	1.00	ns
Recall	LR	NB	2.72×10^{-10}	****	2.72×10^{-9}	****
Recall	DT	RF	2.32×10^{-4}	***	2.32×10^{-3}	**
Recall	LR	RF	1.08×10^{-19}	****	1.08×10^{-18}	****
Recall	NB	RF	5.52×10^{-3}	**	5.52×10^{-2}	ns
Recall	DT	XG	1.30×10^{-2}	*	1.30×10^{-1}	ns
Recall	LR	XG	3.06×10^{-15}	****	3.06×10^{-14}	****
Recall	NB	XG	1.14×10^{-1}	ns	1.00	ns
Recall	RF	XG	2.31×10^{-1}	ns	1.00	ns

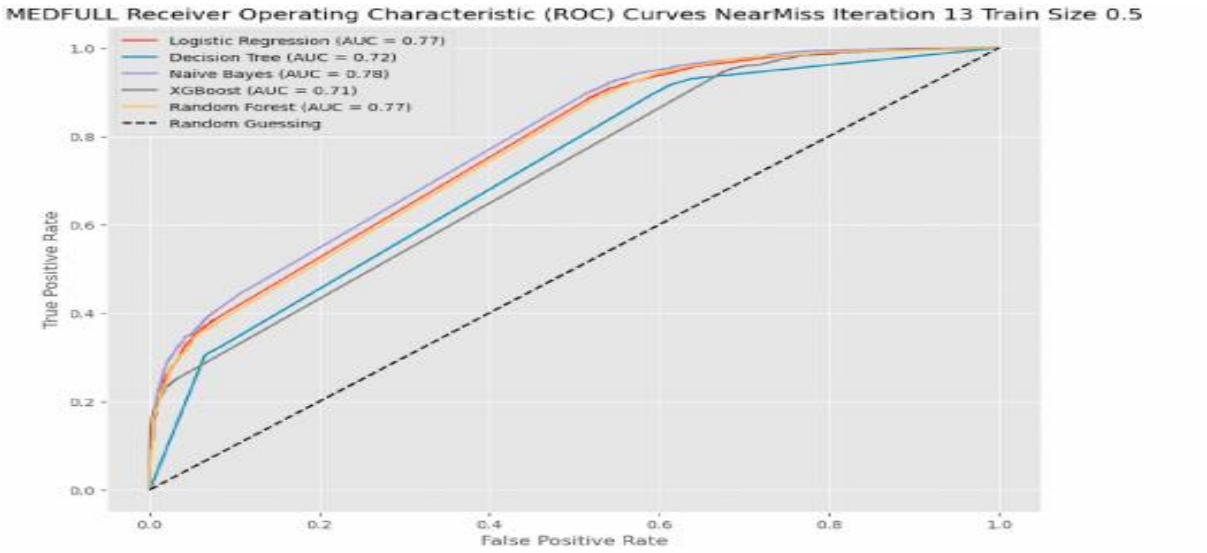
[1] The classifiers are compared based on 3000 observations. Full words for abbreviations: LR—Logistic Regression, XG—XGBoost, DT—Decision Tree, RF—Random Forest, ns—non-significance level. ns means $p > 0.05$, * means $p \leq 0.05$, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$. p.signf means the significance level of p-values, p.adj means adjusted p-value, and p.adj.signif means the significance level of the adjusted p-value.

The 'MEDFULL—Precision & Technique' graph shows an F-value score of 7.19 and an overall p-value less than 0.0001, indicating that all sampling techniques and imbalanced data will have partial differences in performance metrics compared to other groups. However, each sampling technique provides both significant and non-significant results compared to other groups.

The 'MEDFULL—Recall & Technique' graph shows an F-value of 1.25 and a p-value of 0.28, indicating no or slight variation among the sample means for recall scores. The only significant comparison of a sampling technique is between Imbalanced and SMOTE, with a p-value of 0.0245. Other comparisons are insignificant.

The ROC curves in Figure 4 illustrate the performance of different classifiers depending on the training size. Naïve Bayes has the highest AUC values in all graphs, ranging from 0.74 to 0.78. Logistic Regression and Random Forest have the second-highest values between 0.72 and 0.77. Decision Tree and XGBoost performed similarly, with values between 0.66 and 0.70 when the training size was 16% and 33%. Decision Tree has the second-lowest performance of 0.72, and XGBoost has the lowest performance of 0.71.





Presents the ROC curves derived from the 13th iteration of the NearMiss algorithm applied to the MEDFULL dataset.

Tllustrates the precision and recall metrics across various facets, comparing different sampling techniques for the MEDFULL dataset. The results indicate that all imbalanced sampling methods and the imbalanced data itself exhibit comparable performance, failing to effectively address the issue of data imbalance.

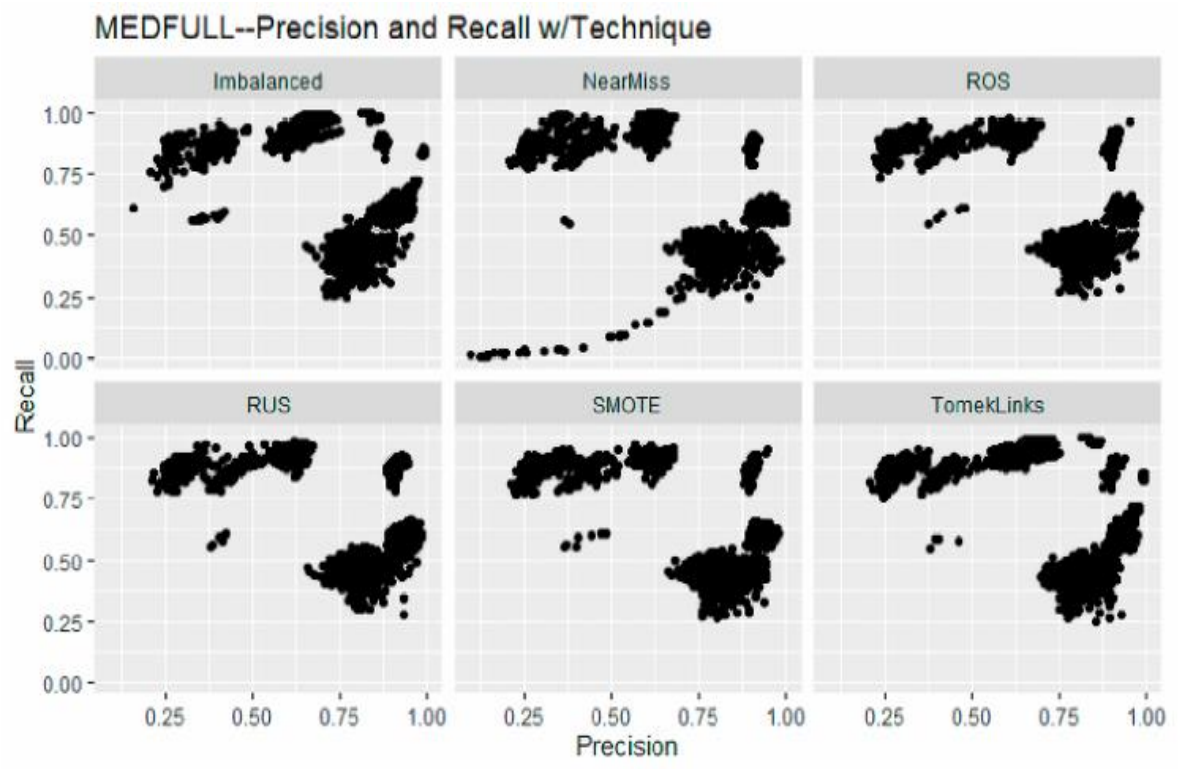
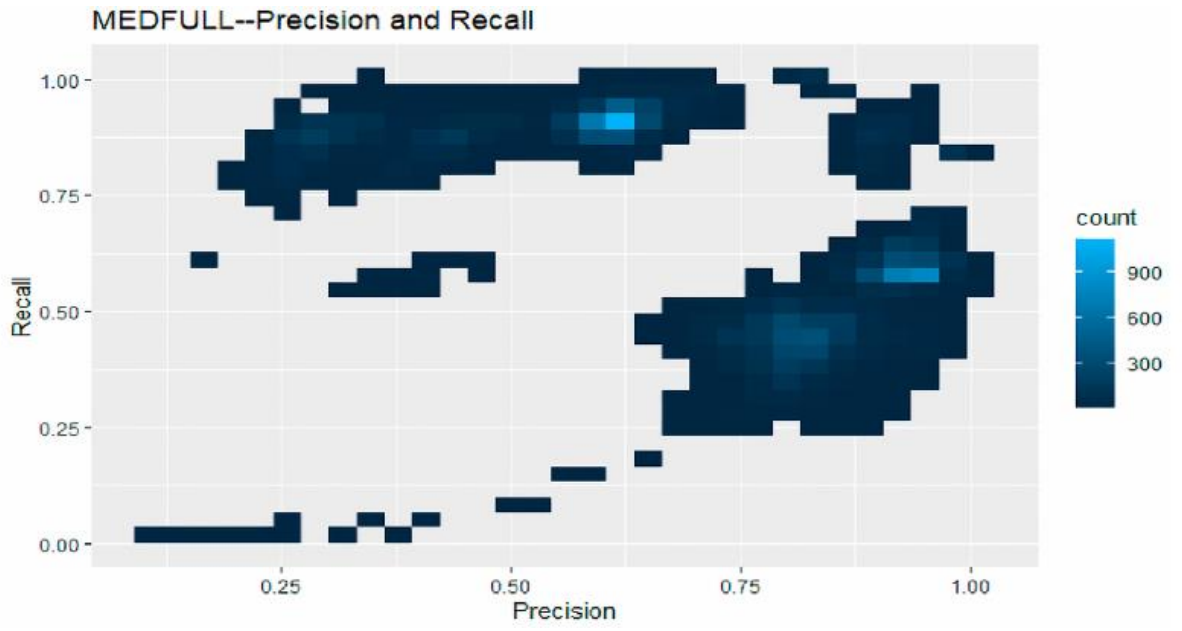


Figure 5. Facets of Recall and Precision Based on Technique for MEDFULL Data.

Presents a heatmap illustrating the precision and recall metrics for the MEDFULL dataset. The MEDFULL dataset exhibits significant variability due to the larger number of scores and observations. The heatmap indicates that the highest values are observed when both recall and precision are approximately 60%. Despite employing imbalanced sampling techniques, the classifiers yield the lowest recall and precision scores in comparison to the Subset data.



Results from Subset Data

The Subset data collection includes three subsets: TrainSet, DevSet, and TestSet. This method uses regular expressions to check if a sentence matches a keyword and labels it as either positive or negative. The Subset data collection has resolved the imbalance issue .

According to Table 14, the highest scores are possible when the iteration count is 9 or 10, or when the subsample size is 90% or 100%. Therefore, machine learning models achieve the best performance from Subset data by using 90% and 100% of the dataset.

Highest Performance Metrics from Subset Iterations.

Subset	Other Set	Technique [1]	Classifier [1]	Iteration	Precision	Recall	AUROC	Accuracy
Train_Set	Test_Set	ROS	XG	9	0.991202	0.936199	0.961843	0.961836
Train_Set	Test_Set	ROS	RF	9	0.991104	0.939023	0.963742	0.963584
Train_Set	Test_Set	ROS	LR	10	0.99074	0.935225	0.96111	0.961088
Train_Set	Test_Set	ROS	NB	10	0.990414	0.93441	0.960697	0.96059
Train Set	Test Set	ROS	DT	10	0.990303	0.9352	0.96712	0.96181

[1] This table only shows statistics related to Random Over Sampling. Abbreviations for Full terms: NM—NearMiss, XG—XGBoost, RF—Random Forest, LR—Logistic Regression, NB—Naïve Bayes, DT—Decision Tree, ROS—Random Over Sampling.

Table indicates that the minimum scores are achieved with a dataset subsampling size of 50% or 60% and iterations of 5 or 6. The findings reveal that using only half of the samples from the Trainset, DevSet, and Test Set can yield the lowest scores from the dataset.

Table 15. Minimum Performance Metrics from Subset Iterations.

Subset	Other Set	Technique [1]	Classifier [1]	Iteration	Precision	Recall	AUROC	Accuracy
Train_Set	Test_Set	NM	LR	5	0.541509	0.988519	0.597201	0.58692
Train_Set	Test_Set	NM	DT	5	0.578896	0.989391	0.647256	0.641251
Train_Set	Test_Set	NM	XG	5	0.594941	0.993838	0.665787	0.662264
Train Set	Test Set	NM	RF	5	0.598006	0.982636	0.66621	0.663715
Train Set	Test Set	RUS	NB	6	0.676887	0.989792	0.762975	0.760883

[1] This table only shows statistics related to undersampling techniques. Abbreviations for Full terms: NM—NearMiss, LR—Logistic Regression, DT—Decision Tree, XG—XGBoost. RF—Random Forest, and NB—Naïve Bayes, RUS—Random Under Sampling.

Table provides a summary of the average precision, recall, AUROC, and accuracy scores for each classifier, independent of other categories. Logistic Regression shows the lowest performance among the classifiers, whereas Random Forest demonstrates the highest performance.

Average Performance Metrics by Classifier for Subset Data.

Classifier	Precision	Recall	AUROC	Accuracy
Logistic Regression	0.87255	0.90081	0.90472	0.92019
Decision Tree	0.87802	0.89721	0.90504	0.92357
Naïve Bayes	0.88158	0.90666	0.91178	0.92795
XGBoost	0.88811	0.90934	0.91281	0.92854
Random Forest	0.89632	0.91591	0.91861	0.93306

This Table presents a summary of the average precision, recall, AUROC, and accuracy scores for each sampling method, independent of other categories. Once again, the oversampling techniques show superior performance compared to both imbalanced data and undersampling methods.

Average Performance Metrics Based on Sampling for Subset Data.

Classifier	Precision	Recall	AUROC	Accuracy
Logistic Regression	0.87255	0.90081	0.90472	0.92019
Decision Tree	0.87802	0.89721	0.90504	0.92357
Naïve Bayes	0.88158	0.90666	0.91178	0.92795
XGBoost	0.88811	0.90934	0.91281	0.92854
Random Forest	0.89632	0.91591	0.91861	0.93306

This table presents a comparison between Logistic Regression and Random Forest in Group 1 and Group 2, resulting in a p-value of 0.0414, which is significant at the 0.05 level (indicated by an asterisk). However, this significance vanishes after adjusting for multiple comparisons (p.adj), implying that the observed difference might be due to chance. The results indicate no statistically significant differences in precision between Group 1 and Group 2.

t-test Statistics of Precision Based on Classifiers for Subset Data.

Metric	Group 1 [1]	Group 2 [1]	p	p.signif [1]	p.adj [1]	p.adj.signif [1]
Precision	DT	LR	0.6380	ns	1.000	ns
Precision	DT	NB	0.7600	ns	1.000	ns
Precision	LR	NB	0.4380	ns	1.000	ns
Precision	DT	RF	0.1160	ns	1.000	ns
Precision	LR	RF	0.0414	*	0.414	ns
Precision	NB	RF	0.2050	ns	1.000	ns
Precision	DT	XG	0.3860	ns	1.000	ns
Precision	LR	XG	0.1810	ns	1.000	ns
Precision	NB	XG	0.5740	ns	1.000	ns
Precision	RF	XG	0.4810	ns	1.000	ns

[1] The classifiers are compared based on 120 observations. Full words for abbreviations: LR—Logistic Regression, XG—XGBoost, DT—Decision Tree, RF—Random Forest, ns—non-significance level. ns means $p > 0.05$, * means $p \leq 0.05$, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$. p.signf means the significance level of p-values, p.adj means adjusted p-value, and p.adj.signif means the significance level of the adjusted p-value.

The next table indicates that all p-values exceed the typical significance threshold of 0.05. Consequently, the recall measure does not exhibit statistically significant differences between Group 1 and Group 2 for any of the classifiers used, including Decision Tree, Logistic Regression, Naive Bayes, Random Forest, and XGBoost. Even after accounting for multiple comparisons, the adjusted p-values show no significant changes. Based on the recall measure, the study suggests that there is no clear distinction in the performance of the machine learning methods between Group 1 and Group 2 with the given dataset.

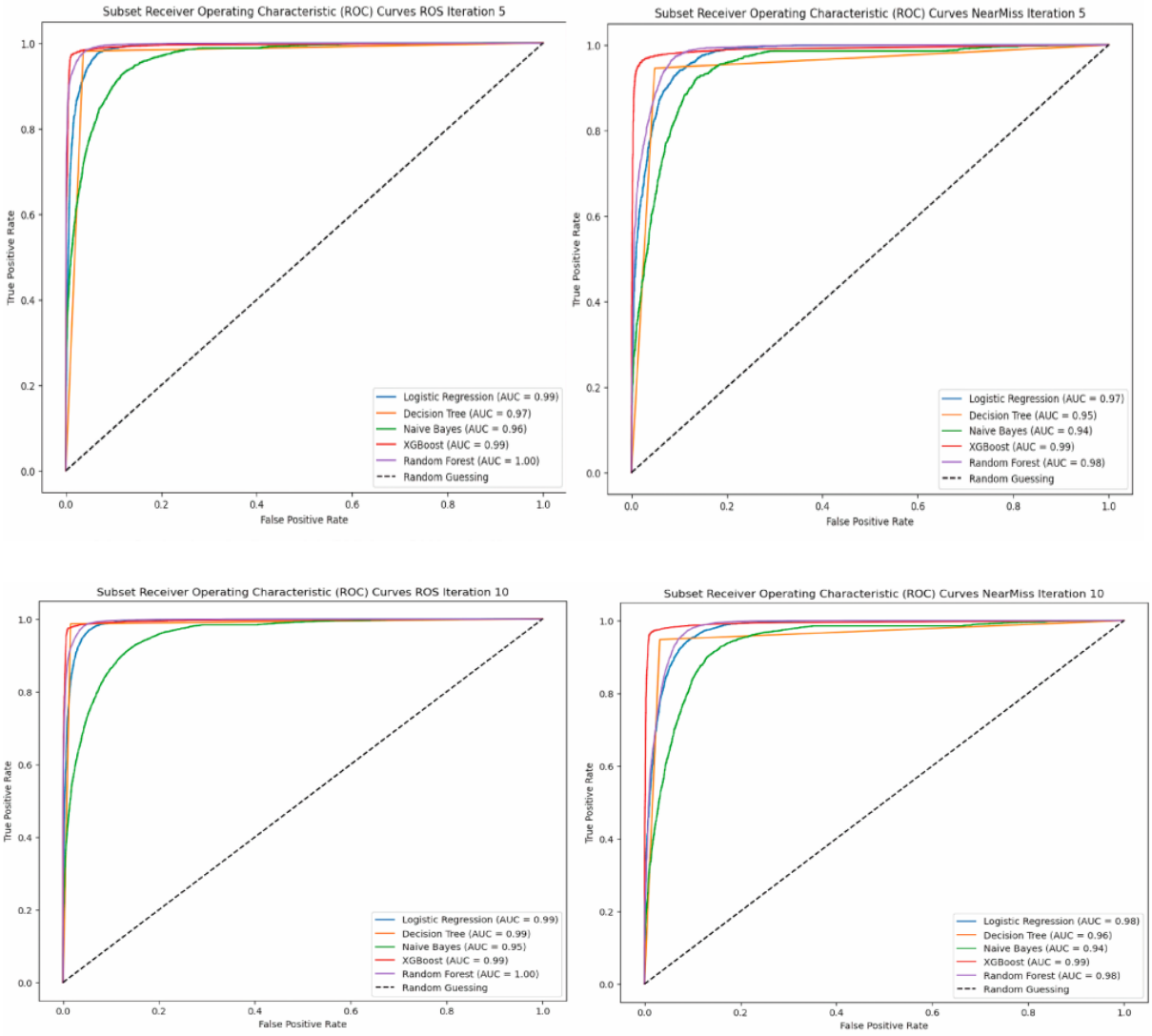
t-test Statistics of Recall Based on Classifiers for Subset Data.

Metric	Group 1 [1]	Group 2 [1]	p	p.signif [1]	p.adj [1]	p.adj.signif [1]
Recall	DT	LR	0.7320	ns	1.000	ns
Recall	DT	NB	0.3700	ns	1.000	ns
Recall	LR	NB	0.5790	ns	1.000	ns
Recall	DT	RF	0.0765	ns	0.765	ns
Recall	LR	RF	0.1530	ns	1.000	ns
Recall	NB	RF	0.3800	ns	1.000	ns
Recall	DT	XG	0.2500	ns	1.000	ns
Recall	LR	XG	0.4190	ns	1.000	ns
Recall	NB	XG	0.8000	ns	1.000	ns
Recall	RF	XG	0.5330	ns	1.000	ns

[1] The classifiers are compared based on 120 observations. Full words for abbreviations: LR—Logistic Regression, XG—XGBoost, DT—Decision Tree, RF—Random Forest, ns—non—non-significance level. ns means $p > 0.05$, * means $p \leq 0.05$, ** means $p \leq 0.01$, *** means $p \leq 0.001$, **** means $p \leq 0.0001$. p.signf means the significance level of p-values, p.adj means adjusted p-value, and p.adj.signif means the significance level of the adjusted p-value.

The 'Subset—Recall & Technique' graph shows an F-value of 77.72 and a p-value less than 0.0001 for the sampling techniques. The Imbalanced data label and Tomek Links perform similarly to other sampling techniques such as NearMiss, ROS, RUS, and SMOTE.

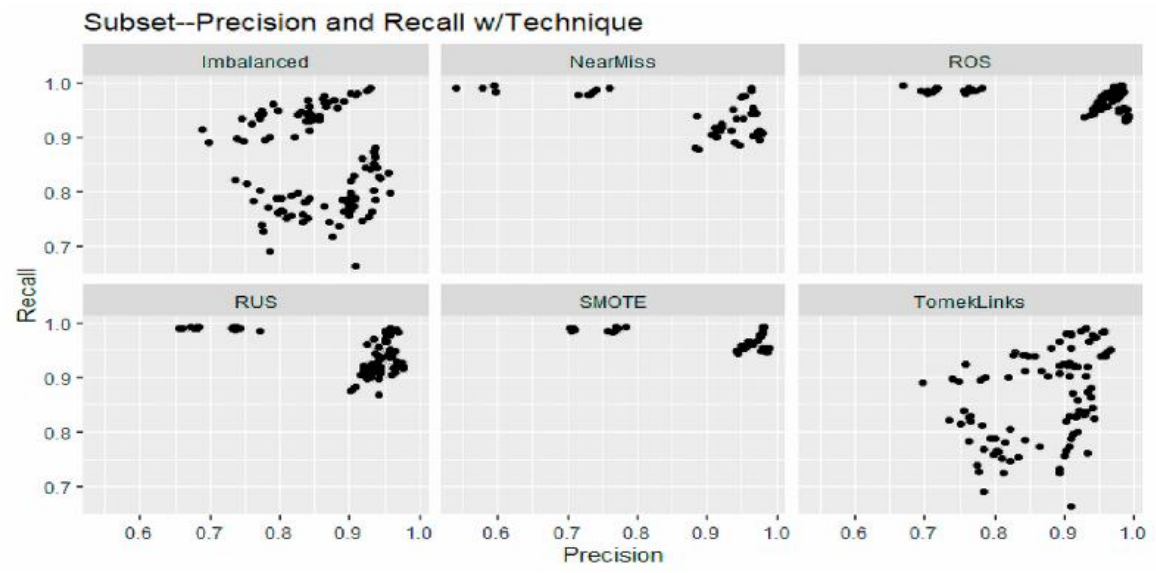
The ROC curves in Figure 8 depict the highest and lowest performances possible for the subset data. Performance is closely similar for all classifiers in Iterations 5 and 10. Iteration 10 shows a slight performance drop compared to Iteration 5. The highest performance comes from Iteration 5 of Random Oversampling, while Iteration 10 of NearMiss records the lowest performance. Random Forest and XGBoost classifiers exhibit the highest average performance at 0.99, whereas Naive Bayes has the lowest performance among all classifiers.



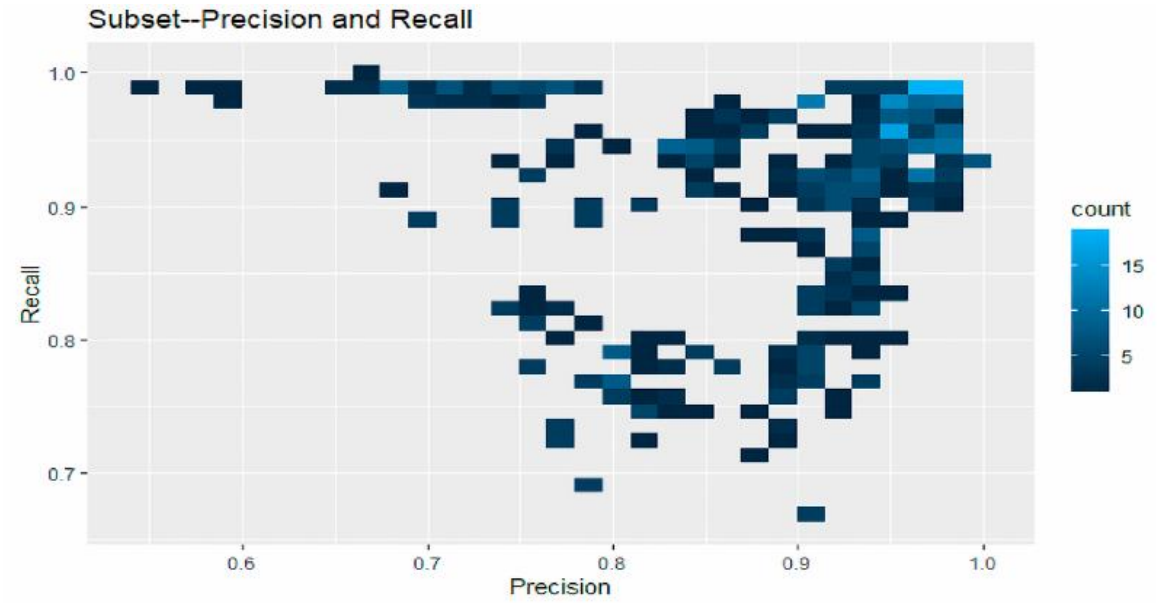
ROC Curves Based on Near Miss and ROS Iterations 5 and 10 for Subset Data.

Presents a scatter plot of precision and recall in a two-dimensional graph, comparing imbalanced sampling techniques for subset data. Fewer observations from the subset data result in fewer plots from the MEDFULL data. Tomek Links and Imbalanced data show similar patterns with precision and recall ranging from 69% to 98%. ROS and SMOTE demonstrate high recall and precision compared to other techniques, including Imbalanced data.

Figure 11 features a heatmap of precision and recall in a two-dimensional graph for subset data. There is low variability due to the number of observations. The lighter the color, the higher the number of values in that graph area. The heatmap makes it easier to identify the highest scores for precision and recall.



Facets of precision and recall based on technique for subset data.



Model Execution

The classifiers were executed using Scikit-learn modules and the XGBoost library. Decision Tree, XGBoost, and Naive Bayes classifiers ran without additional parameters; Multinomial Naive Bayes was used. Logistic Regression employed an l2 penalty, random state of zero, lbfgs solver, automatic multi-class handling, and a maximum of 500 iterations. The Random Forest Classifier used 1000 estimators and a random state of zero.

Experiments were conducted on lab workstations with 16GB to 32GB of RAM and CPUs from 2.5GHz to 5GHz, running three to six models simultaneously. A Linux cluster with 200GB to 800GB of RAM is recommended for optimal efficiency, capable of running up to 75 models concurrently.

Conclusions and Discussion

This study analyzes the performance of ML classifiers and sampling techniques on document datasets. On Subset data, classifiers without sampling techniques achieved an average accuracy of 84.945% and recall of 85.076%, while those with sampling techniques reached 90.002% precision and 93.335% recall. On MEDFULL data, classifiers without sampling techniques averaged 73.296% precision and 69.636% recall, and those with sampling techniques achieved 71.954% precision and 68.706% recall.

Manual classification for MEDFULL data showed no class imbalance, unlike automatic classification for Subset data. Despite fewer observations in Subset data, it exhibited more variation and effectiveness in performance, as indicated by ANOVA scores, compared to MEDFULL data.

Classifiers generally perform better on unstructured text with imbalanced sampling techniques. However, varying training and testing sizes and subsampling data can unpredictably impact performance metrics. This paper addresses the shortcomings of previous studies by employing training size variation and subsampling.

Future work will involve deep learning algorithms with models like BERT and PyTorch, requiring larger datasets for fine-tuning and superior performance. Similar subsampling and training size variation methods, along with feature engineering techniques, will be applied, and statistical analysis will evaluate deep learning algorithm performance.

References

Büttcher, S., Clarke, C., & Cormack, G. V. (2010). Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press.

Belkin, N. J., & Croft, W. B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM, 35(12), 29–38.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text Classification Algorithms: A Survey. Information, 10(5), 150.

Zhou, Z.-H., & Liu, X.-Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18(1), 63–77.

Zhang, Z., Jasaitis, T., Freeman, R., Alfrjani, R., & Funk, A. (2023). Mining Healthcare Procurement Data Using Text Mining and Natural Language Processing—Reflection from an Industrial Project. arXiv, arXiv:2301.03458.

Borko, H., & Bernick, M. (1964). Automatic Document Classification Part II. Additional Experiments. Journal of the ACM, 11(2), 138–151.

Shakarami, A., Ghobaei-Arani, M., & Shahidinejad, A. (2020). A Survey on the Computation Offloading Approaches in Mobile Edge Computing: A Machine Learning-based Perspective. Computer Networks, 182, 107496.

Akritidis, L., & Bozanis, P. (2013). A Supervised Machine Learning Classification Algorithm for Research Articles. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC'13) (pp. 115–120). Association for Computing Machinery.

Goudjil, M., Koudil, M., Bedda, M., & Ghoggali, N. (2018). A Novel Active Learning Method Using SVM for Text Classification. International Journal of Automation and Computing, 15(3), 290–298.

Kadhim, A. I. (2018). An evaluation of preprocessing techniques for text classification. International Journal of Computer Science and Information Security, 16(1), 22–32.

Mali, M., & Atique, M. (2021). The Relevance of Preprocessing in Text Classification. In K. S. Mer, V. B. Semwal, V. Bijalwan, & R. G. Crespo (Eds.), Integrated Intelligence Enabled Networks and Computing (pp. 553–559). Springer.

Imberg, H., Yang, X., Flannagan, C., & Bärgman, J. (2022). Activesampling: A machine-learning-assisted framework for finite population inference with optimal subsamples. arXiv, arXiv:2212.10024.

Kumar, V., Balloccu, S., Wu, Z., Reiter, E., Helaoui, R., Recupero, D. R., & Riboni, D. (2023). Data Augmentation for Reliability and Fairness in Counselling Quality Classification. In Proceedings of the 1st Workshop on Scarce Data in Artificial Intelligence for Healthcare-SDAIH (pp. 23–28).

SciTePress Oyedare, T., & Park, M. J. (2019). Estimating the Required Training Dataset Size for Transmitter Classification Using Deep Learning. In Proceedings of the 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN).

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. (2022). A Survey on Text Classification: From Traditional to Deep Learning. ACM Transactions on Intelligent Systems and Technology, 13(1), 1–41.

Mujtaba, G., Shuib, L., Idris, N., Hoo, W. L., Raj, R. G., Khowaja, K., Shaikh, K., & Nweke, H. F. (2019). Clinical text classification research trends: Systematic literature review and open issues. Expert Systems with Applications, 116, 494–520.

Kamath, C. N., Bukhari, S. S., & Dengel, A. (2018). Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. In Proceedings of the ACM Symposium on Document Engineering 2018 (DocEng'18).

Kim, M., & Hwang, K.-B. (2022). An empirical evaluation of sampling methods for the classification of imbalanced data. PLoS ONE, 17(1), e0271260.

Agarwal, B., & Mittal, N. (2014). Text Classification Using Machine Learning Methods—A Survey. In Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS2012) (pp. 701–709). Springer.

Gaudreault, J.-G., Branco, P., & Gama, J. (2021). An Analysis of Performance Metrics for Imbalanced Classification. In Discovery Science (pp. 67–77).