Towards Optimizing Performance of Machine Learning Algorithms on Unbalanced Dataset

Asitha Thumpati and Yan Zhang

School of Computer Science and Engineering
California State University San Bernardino
5500 University Pwy, San Bernardino 92407 USA

Abstract. Imbalanced data, a common occurrence in real-world datasets, presents a challenge for machine learning classification models. These models are typically designed with the assumption of balanced class distributions, leading to lower predictive performance when faced with imbalanced data. To address this issue, this paper employs data preprocessing techniques, including Synthetic Minority Oversampling Technique (SMOTE) for oversampling and random undersampling, on unbalanced datasets. Additionally, genetic programming is utilized for feature selection to enhance both performance and efficiency. In our experiment, we leverage an imbalanced bank marketing dataset sourced from the UCI Machine Learning Repository. To evaluate the effectiveness of our techniques, we implement it on four different classification algorithms: Decision Tree, Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). We compare various evaluation metrics, such as accuracy, balanced accuracy, recall, F-score, Receiver Operating Characteristics (ROC) curve, and Precision-Recall (PR) curve, across different scenarios: unbalanced data, oversampled data, undersampled data, and data cleaned with Tomek-Links. Our findings reveal that all four algorithms demonstrate improved performance when the minority class is oversampled to half the size of the majority class and the majority class is undersampled to match the minority class. Subsequently, applying Tomek-Links on the balanced dataset further enhances performance.

Keywords: Unbalanced Dataset, Oversampling, Undersampling, Feature Selection, Classification

1 INTRODUCTION

Classification tasks are fundamental in machine learning, where algorithms are trained to categorize new data into predefined classes or labels based on a given dataset or observations. These algorithms have evolved to excel in this task, assuming that the training data is well-balanced, with an equal number of instances across different classes and identical misclassification costs. The rapid advancements in science and technology have ushered in an era of unprecedented data generation and accessibility. However, a recurring challenge emerges—much of the data collected from diverse sources is inherently imbalanced. Imbalanced datasets exhibit a substantial disparity in the number of instances between different classes [1].

In a classification dataset, when the distribution of examples across known classes is skewed or biased, it qualifies as an imbalanced dataset. This class imbalance can manifest in binary classification problems or extend to multi-class scenarios. For the purpose of this paper, we focus on the binary imbalance problem, where one class (referred to as the majority class) significantly outnumbers the other (referred to as the minority class).

The extent of class imbalance is quantified by the class imbalance ratio, computed as the ratio of the sample size of the majority class to that of the minority class. A high class imbalance ratio indicates a skew towards the majority class. However, machine learning models are inherently designed to perform optimally on balanced datasets. When exposed to skewed datasets, these models tend to make biased predictions, favoring the majority class due to the limited exposure to the minority class during training. Furthermore, conventional classification models typically assign an equal misclassification error for both

David C. Wyld et al. (Eds): SIPR, NCWC, BIBC, SOFEA, DSML, ARIA, NLP, CSEN -2023
pp. 169-183, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131914

false negatives and false positives, which is not conducive to training on imbalanced data. When traditional metrics such as accuracy are used to evaluate model performance, they often exhibit a strong bias toward the majority class. In such cases, the majority class achieves accuracy close to 100%, while the minority class struggles to reach accuracy levels above 10%.

Despite intensive research efforts spanning the past two decades, addressing the challenge of imbalanced learning remains an ongoing and crucial endeavor [2] . Each dataset presents unique characteristics that demand tailored solutions. In this research, we aim to contribute to the ongoing efforts by harnessing a combination of advanced preprocessing techniques to rectify data imbalances. Additionally, we will leverage genetic programming as a means of feature selection, enhancing the model's capacity to navigate complex datasets and further improving its performance. To address this challenge of imbalanced dataset, two common strategies are employed: data-level preprocessing techniques and algorithm-level enhancements. Data preprocessing encompasses the task of addressing class imbalance by employing techniques such as undersampling the majority class, oversampling the minority class, or employing hybrid methods. Algorithm-level solutions incorporate advanced techniques like bagging and boosting to modify regular classifiers and adapt them to imbalanced datasets.

The paper is structured as follows: In Section 2, we delve into an examination of existing solutions addressing the class imbalance problem, offering perspectives on the historical context and prior work in this domain. In Section 3, we present our innovative approach to tackle this persistent issue. Section 4 constitutes the empirical evaluation of our proposed techniques. The effectiveness across various classification models by employing chosen evaluation metrics are assessed. Finally, in Section 5, we draw conclusions from our findings, summarizing the contributions and implications of our proposed system.

2 RELATED WORK

To learn more about this issue we explore the existing solutions for imbalanced data classification.

In a study conducted by Mishra, a comprehensive approach to address imbalanced datasets was adopted [2].This approach involved a combination of preprocessing and ensemble techniques. Specifically, preprocessing techniques, namely Synthetic Minority Oversampling Technique (SMOTE) and random undersampling, were individually applied to the dataset. Each modified dataset was fed into ensemble classification models, including Random Forest and XGBoost. Through a meticulous comparison of diverse metrics, such as the Area Under Curve score, (AUC) sensitivity, and specificity, the author arrived at a significant conclusion that the utilization of SMOTE in conjunction with the Random Forest ensemble model yielded the highest sensitivity 79.19% [2].

Elhassan employed a multifaceted approach to address the challenge of imbalance data set [3]. Tomek-Links were utilized on the original imbalanced dataset to effectively eliminate noise from the data. Then, a combination of random undersampling and SMOTE was applied to this preprocessed dataset to assess the performance of various classification models. The author claimed that employing Tomek-Links in conjunction with random undersampling as a combined sampling method significantly enhanced model performance. The improvements were observed in terms of specificity, weighted accuracy, precision, G-mean, and F-statistics across multiple classification models, including Random Forest, Logistic Regression, Support Vector Machines, and Artificial Neural Networks [3].

In [4], the authors introduced two techniques aimed at addressing the issue of imbalanced datasets through majority class clustering. The first technique involves grouping the

majority class instances into clusters, each containing an equal number of data points as the minority class. K-Nearest Neighbors (KNN) algorithm is used to form these clusters and the centroid of each cluster is calculated and is used as a new instance within the majority class. The second technique selects an existing record from the original dataset that has the shortest euclidean distance to the center of the cluster. To evaluate the effectiveness of these two techniques, five classification models are chosen along with AdaBoost algorithm for ensemble learning. By conducting these two experiments on a diverse set of datasets, including 44 small-scale and 2 large-scale datasets, the authors concluded that employing the second technique for undersampling the majority class, in conjunction with using Multilayer Perceptron (MLP) as classifier, yielded better accuracy and Area Under the Receiver Operating Characteristic (ROC) Curve scores when compared to other tested combinations [4].

A preprocessing technique using Genetic Programming (GP) has been proposed for effective feature selection and construction to address imbalanced datasets [5] [6]. The proposed approach combines filters and genetic programming to select impactful features and also generate novel features from the initial dataset, known as as Feature Selection (FS) and Feature Construction and Modification (FCM) algorithms [6]. Considering FS and FCM as the two base algorithms, a combined technique called FCMFS is implemented, where multiple features are constructed from existing features using FCM and then the most effective features are selected with FS. Experimental results from 9 diverse datasets showed that FS and FCM demonstrated higher performance scores with comparison to the original feature sets. FCMFS surpassed both of these two algorithms by achieving higher classification accuracy with reduced number of features, thereby enhancing the overall model performance [6].

3 METHODOLOGY

In the proposed approach, we will be handling the previously discussed imbalance dataset problem through data-level interventions, specifically by combining two key data preprocessing techniques: data oversampling and undersampling. This section delves into the details of the data sources, the preprocessing techniques employed, and genetic programming algorithms used to implement the proposed approach.

3.1 Dataset

To perform the proposed technique, we use an imbalanced dataset sourced from the UCI Machine Learning Repository known as the "Bank Marketing" dataset [7]. This dataset was collected by a Portuguese banking institution during a direct marketing campaign through phone calls where its clients were recommended to subscribe for a bank term deposit. The classification problem involves predicting if a client would subscribe to the term deposit based on a set of input features. The original dataset, named "bank-full.csv" has a total of $45,211$ records with sixteen input attributes and one class variable. Table 1 provides a comprehensive breakdown of each client's attributes.

These attributes include demographics information such as age, education, marital status, and occupation, along with financial details like bank balance, home or personal loans, and data related to campaign call. The output variable, denoted as y, possesses two class values, with the value *yes* signifying the client subscribing to the term deposit and the value *no* signifying the client not opting for the term deposit. Among the 45,211 records, 39,922 instances fall into the *no* class, constituting the majority class, whereas 5,289 instances fall into the *yes* class constituting the minority class. The majority class

Table 1. Description of Attributes

Attribute name	Description	Variable type	Values
age	Age of the client	numeric	18-95
job	Type of job	categorical	"admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services"
marital	Marital status	categorical	"married", "divorced", "single"; note: "divorced" means divorced or widowed
education	Level of education	categorical	"unknown", "secondary", "primary", "tertiary"
default	Has credit in default?	categorical	"yes", "no"
balance	Average yearly balance	numeric	-8019 to 102127
housing	Has housing loan?	categorical	"yes", "no"
loan	Has personal loan?	categorical	"yes", "no"
contact	How contacted?	categorical	"unknown", "telephone", "cellular"
day	Last contact day	numeric	1-31
month	Last contact month	categorical	"jan", "feb", ..., "nov", "dec"
duration	Last contact duration	numeric	0 – 4918 sec
campaign	Number of contacts	numeric	1-63
pdays	Days since last contact	numeric	-1 to 871; -1 means not contacted
previous	Number of previous contacts	numeric	0-275
poutcome	Previous outcome	categorical	"unknown", "other", "failure", "success"
y	Subscribed to term deposit?	categorical	"yes", "no"

no accounts for approximately 89% of the dataset, indicating a significant class imbalance. The class imbalance ratio for this dataset stands at 7.5. Figure 1 depicts sample instances from the original dataset, offering an insight into the dataset's structure and attributes. This dataset serves as a representative foundation for our research, allowing

age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no

Fig. 1. Sample of Original Dataset

us to explore effective techniques to address class imbalance and optimize classification model performance

3.2 Data Preprocessing

Data preprocessing stands as a fundamental and pivotal stage within the machine learning workflow. In the case of the Bank Marketing dataset, which comprises a mix of categorical, discrete, and continuous attributes, we employ label encoding to convert all attributes into a numeric type. This transformation reduces computational overhead. Additionally, for attributes like *duration*, originally measured in seconds, we convert the values into minutes to narrow the range. Considering the application of distance-based algorithms such as K-Nearest Neighbors, Support Vector Machines, and logistic regression, it becomes imperative to standardize the feature scales of *balance* and *duration* using standardizing technique standard scalar.

Our proposed approach handles the challenges posed imbalanced dataset at the data-level by oversampling the minority class using SMOTE and then passing this modified data to random undersampling. This combination ensures that the final class imbalance ratio reaches one. Then we further refine and clean this balanced dataset with Tomek-Links [3] [8][9]. To enhance model efficiency and effectiveness, we use Genetic Programming on top of this balanced and cleaned dataset to select the most efficient and relevant features for model training. In our comparative analysis, we assess the performance of our proposed approach against various scenarios, including the use of the original data, oversampled data, oversampled data cleaned with Tomek-Links, undersampled data, and undersampled data cleaned with Tomek-Links. This in-depth assessment enables us to see the effectiveness of our proposed methodology in addressing the challenges posed by class imbalance in the dataset.

Oversampling Oversampling is a common technique employed to mitigate class imbalance by duplicating existing minority class instances and appending them to the original dataset to reduce the imbalance ratio. Although this method successfully decreases the imbalance ratio, it introduces data redundancy and potentially conveys false information due to the inclusion of identical instances.

To address these limitations, an advanced oversampling technique, Synthetic Minority Oversampling Technique (SMOTE), was introduced [8]. SMOTE takes a distinctive approach by generating synthetic data points from the original dataset. Data augmentation adds minor changes to duplicated minority class instances aligning them with the direction of the original data points. A random space is generated from the minority class and K-Nearest Neighbors are calculated for the observations within this space. The distance between the observation and a randomly selected neighbor is multiplied by a number between 0 and 1, which is then added to the original observation. This leads to a new data vector in the direction of the chosen neighbor. This helps generate data points that are not same to those found in the initial dataset [10].

By adopting SMOTE in our data preprocessing, we enhance the representativeness of the minority class while avoiding the issues associated with simple oversampling.

Undersampling Undersampling is a data resampling techniqueaimed at reducing the number of records from the majority class while retaining the minority class examples [11]. The random undersampling technique characterizes by the stochastic removal of majority class samples. It is advised to use random undersampling when the number of majority class records significantly outweighs that of the minority class [9].

Another undersampling technique, known as Tomek-Links, not only reduces the data but can also be employed for data cleaning. It was developed by Ivan Tomek in 1976

by modifying Condensed Nearest Neighbors (CNN) algorithm [12]. The CNN algorithm aims to produce a training set consistency from the original dataset — a subset that can accurately classify data points present in the original data [13] [14]. One challenge with random undersampling is that it can result in the retention of unnecessary samples while excluding boundary samples. To address this issue, two modifications to the CNN algorithm have been proposed [15]. These modifications involve selecting a pair of examples from the original dataset, ensuring they belong to different classes and have the minimum distance between them. This pair of examples is referred to as Tomek-Links, which consists of boundary instances and noise instances, both of which affect the learning process. In the context of an imbalanced dataset, Tomek-Links can be used to identify all the majority class samples that serve as nearest neighbors to the minority class. This information can then be utilized to selectively remove these instances from the dataset [12] [15].

By leveraging these undersampling techniques, we aim to create a more balanced and representative dataset, thereby facilitating the training of machine learning models that can effectively address class imbalance.

3.3 Feature Selection

For optimal model performance, it is important that the classification model learns from data that is devoid of redundant and noise. Feature selection is a pivotal technique to handle this concern by identifying and selecting the most relevant and effective features from the entire dataset. Various methods have evolved over time for feature selection, including filter-based methods, wrapper methods and embedded methods [16].

Genetic programming, an evolutionary algorithm inspired by Charles Darwin's theory of evolution, is a versatile tool applicable to a wide array of optimization problems [17]. In the context of feature selection, genetic algorithms are employed to obtain a subset of features that best train the model [17]. The genetic algorithm comprises five fundamental stages: initial population, fitness function, selection, crossover, and mutation.

Initially, each potential solution to the feature selection problem is organized into a set known as the population. Within this population, each individual is represented by a sequence of binary bits (0s and 1s) known as genes. A chromosome is a string of genes, representing a solution in the population. In the context of feature selection, the population is a power set of features derived from the original dataset which can be restricted by selecting the number of features in the optimal solution [17].

Next, a fitness function is defined and used to calculate the fitness value for each individual in the population [18]. This fitness function can be any performance metric used to evaluate the trained model, such as accuracy or error. Records with higher accuracy and lower error attain higher fitness values, while those with lower accuracy and higher error receive lower fitness values. Based on these fitness values, the individuals in the population are ranked from highest to lowest fitness values.

During the selection phase, the individuals with higher fitness are chosen for recombination to produce the next generation. Two individuals are selected from the result set of the selection phase as parents. Offspring are then generated from these parents through gene interchanges, a process known as crossover. These offspring are subsequently added to the population.

The mutation phase is the final step in the genetic algorithm, where the genes of the offspring are slightly altered to maintain diversity within the population and prevent early convergence [19]. The algorithm iteratively continues the last four steps — generating fitness values, selection, crossover, and mutation — until the optimal solution is achieved or the population exhibits signs of convergence [18].

By utilizing genetic programming for feature selection, we aim to identify the most informative features that facilitate enhanced model training and improved classification performance

Figure 2 illustrates the architecture of the proposed machine learning process, encompassing the pivotal stages involved in the transformation of raw and unprocessed data into a balanced training dataset ready for utilization by classification models.

Fig. 2. The proposed approach to balance imbalanced dataset

3.4 Classification Algorithms

In this paper, we will train four well-established classification models using our proposed preprocessing and feature selection techniques. These models include Decision Tree, Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Our objective is to examine the performance and functionality of each model when trained on our prepared dataset.

Figure 3 shows the process of splitting the balanced dataset into training and testing subsets for training the classifier and evaluating its performance. Prior to model training, each feature space undergoes individual normalization. The training data is subjected to the genetic programming algorithm, which selects the optimal subset of input features for each classifier to be trained. Once the classifiers are trained, they are deployed to predict the output labels of the test dataset. The performances of these classifiers are assessed using a range of performance metrics.

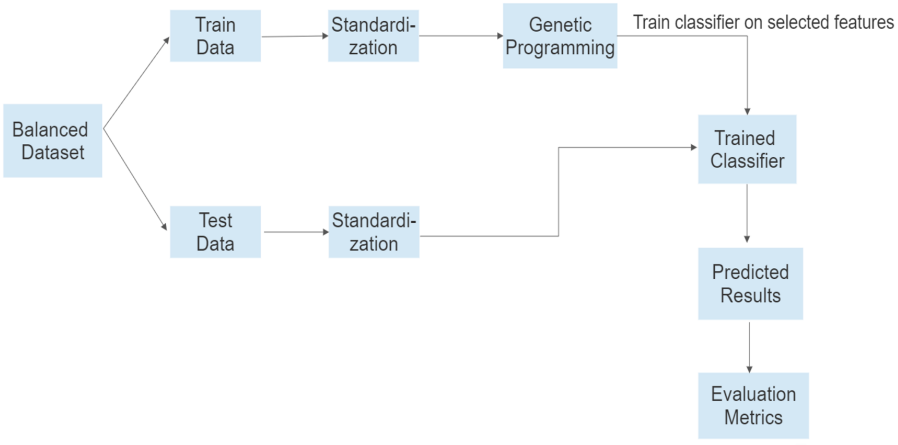

Fig. 3. The proposed approach to balance unbalanced dataset

Decision Trees Decision trees is a supervised, non-parametric and interpretable machine learning model that can be used for both classification and regression [20]. The decision

trees are designed to mimic human decision making processes by breaking down decision making tasks into a series of smaller and simpler choices. Each node in a decision tree represents a test on an attribute or feature, leading to branches that correspond to possible outcomes or values. As traveling the tree, it follows these branches, ultimately arriving at a terminal/leaf node that furnishes the ultimate prediction or decision. The iterative process aims to establish conditions on the input features to separate all the classes contained in the dataset for the fullest purity [20].

To measure the impurity of the node and guide the decision tree's construction, Gini index and Entropy can be used [21]. The Gini index measures impurity of a node by quantifying the difference between one and the sum of squared probabilities for each class [22]. Equation 1 shows how to calculate the impurity of a given node N in a binary classification (with class labels C_1 and C_2) using Gini.

$$Gini(N) = 1 - (\frac{|C_1|}{|N|})^2 - (\frac{|C_2|}{|N|})^2 \tag{1}$$

The impurity after splitting $Node$ into child nodes N_1 and N_2 is the weighted impurities of child nodes N_1 and N_2 , and can be calculated by Equation 2.

$$Gini(Node) = \frac{|N_1|}{|Node|} \times Gini(N_1) + \frac{|N_2|}{|Node|} \times Gini(N_2) \tag{2}$$

Entropy is used to measure the impurity or randomness of a dataset. The optimal split is determined by selecting the one with the lowest entropy after splitting [23]. Decision tree algorithms CART uses Gini index to measure the node impurity and select splitting features; ID3 and C4.5 use Entropy and information gain to select features when building decision trees [21].

Logistic Regression Logistic Regression is a parametric statistical model tailored for binary classification tasks [24]. It captures the correlation between a set of independent variables or attributes and a binary dependent outcome. Logistic Regression leverages a logistic or sigmoid function to map real values to probabilities ranging from 0 to 1, as shown in Equation 3.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

Where z represents the linear combination of the input features and their corresponding weights. When z takes a positive value, $\sigma(z)$ approaches 1, indicating a high probability of belonging to the positive class. In contrast, when z is negative, $\sigma(z)$ tends toward 0, indicating a high probability of the negative class [25].

Its core functionality lies in estimating the parameters' weights by employing the likelihood function, which scrutinizes training data to predict the output labels of new instances. The key strength of Logistic Regression lies in its ability to provide probabilities and make predictions about the likelihood of an event occurring based on the input attributes' values [24]. The iterative techniques like gradient ascent are frequently employed in order to get the optimal weight vector that maximizes the likelihood function [24].

Logistic Regression models have been applied in various domains, including healthcare for disease prediction, finance for credit scoring, natural language processing, and marketing for customer churn analysis.

K-Nearest Neighbors K-Nearest Neighbors (KNN) is a lazy supervised learning algorithm and can be used for both classification and regression tasks. KNN is called lazy learning because it has no training process and just stores training data until test data is provided for prediction [26]. KNN classification classifies a new observation by assessing the classes of its K nearest neighbors. It is important to choose the optimal K value with respect to each data set. Smaller K values make the model sensitive to noise, potentially leading to overfitting. While larger K values lead to misclassification errors [26]. KNN relies on distance metrics such as Manhattan distance, Euclidean distance, and Minkowski distance, etc. to identify the nearest neighbors. Normally, the scaling of feature values is essential to reduce bias and maintain the relevance of each feature's contribution during the distance calculations [27]. KNN is particularly effective when dealing with extensive training datasets and scenarios where noise may be present, which makes it a useful tool in various domains, from recommendation systems to anomaly detection.

Support Vector Machines Support Vector Machine (SVM) is a sophisticated classification technique using linear models. SVM's core objective is to ascertain a decision boundary, often depicted as a hyperplane, that effectively segregates data points based on their respective classes. The essence of SVM revolves around the notion of margin, denoting the distance between this decision hyperplane and the nearest data point from each class. In pursuit of robust performance on test data, SVM aims to identify a hyperplane with the largest margin [28]. In cases where data instances cannot be linearly distinguished within lower dimensions, SVM adds additional features that facilitate separation of data points belonging to different classes in higher-dimensional spaces.

It is highly challenging to find a hyperplane that perfectly separates the data points belonging to different classes when dealing with data from real world. Therefore, a concept called "soft margin" is introduced, where a small misclassification is allowed in the training data while imposing penalties for misclassifying test data [29].

SVM models have been used in diverse applications, ranging from image recognition to sentiment analysis and bioinformatics.

4 EXPERIMENTS

In this section, we present the results of our comprehensive experimental analysis, designed to evaluate the effectiveness of the proposed preprocessing techniques and feature selection via genetic programming. Our experiments were conducted on the Bank Marketing dataset from the UCI Machine Learning Repository—known for its inherent class imbalance. We systematically applied our resampling methods to address this imbalance, which included oversampling, undersampling, and data cleaning. Additionally, we employed genetic programming to select and construct feature subsets aimed at enhancing the performance and efficiency of classification models. We evaluated the impact of these techniques on four widely-used classifiers: Decision Tree, Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). Through assessments and comparisons using various metrics, we aim to provide insights into the optimal strategies for addressing dataset imbalance within the realm of machine learning classification.

4.1 Evaluation Metrics

To ascertain the efficacy of the proposed preprocessing techniques in comparison to the original imbalanced dataset, it is essential to employ appropriate metrics that remain impartial and unbiased toward any single class. These metrics have a crucial role in assessing

the performance of each classifier at various stages of the resampling process. Different evaluation metrics are developed over time to assess the performance of machine learning classification models, out of which accuracy, balanced accuracy, recall, F1 score, area under ROC curve, Receiver Operating Characteristics(ROC) curve, and Precision-Recall(PR) Curve are used in this project.

Accuracy measures the number of correct predictions made out of the total number of predictions by a trained model. Accuracy leads to misleading results when there is a high class imbalance in the dataset [30]. Balanced accuracy gives equal weight to each class, which is calculated by summing the recall for individual classes and dividing it by the number of classes [31].

Recall, also known as sensitivity or true positive rate, measures the ability to correctly identify positive or class of interest samples. It is calculated as the number of true positives divided by the sum of true positives and false negatives, ranging from 0 to 1. A classification model with a higher recall value can predict most of the positive samples correctly [32].

Precision measures the consistency and stability of a model, calculated for the positive class as the number of true positives divided by the sum of true positives and false positives [32].

The F1 score simultaneously considers both recall and precision, given as the harmonic mean of recall and precision [33]. A model with an F1 score of one has perfect recall and precision, which is highly impractical to achieve in real-world data.

The ROC Curve is a widely used evaluation metric when dealing with imbalanced datasets. It is a graphical representation of how well a binary classification model can perform across different classification thresholds. It is calculated by plotting recall/TPR on the Y-axis and the corresponding FPR on the X-axis at different classification thresholds [34]. A point where TPR=1 and FPR=0 on the graph represents an ideal classification model. The area under the ROC Curve measures the separability or ability of a classifier to distinguish between positive and negative classes [35]. The area under the curve is determined using a trapezoidal approximation. A model with a *roc_auc_score* of 1 can distinguish all the records perfectly correctly, whereas a model with a *roc_auc_score* of 0 misclassifies every input. Nevertheless, if there is a significant imbalance in the distribution of classes, ROC curves may provide an excessively positive assessment of a classifier's effectiveness.

In such cases, Precision-Recall (PR) curves are suggested to address the issue of class distribution skew [36]. The PR curve for a model is given by plotting precision on the Y-axis and corresponding recall on the X-axis at different thresholds. As both precision and recall consider true positives, the PR curve is more focused on how well a model predicts minority class samples correctly. The confusion matrix shows the overall performance of a model in tabular form for easy interpretation, especially for imbalanced datasets [37]. It uses an $n \times n$ matrix for n different class classification, which shows actual and predicted outcomes for test data as true positive and true negative for correct predictions, and false positive and false negative for incorrect predictions.

4.2 Performance Analysis

The efficacy of our proposed preprocessing techniques was evaluated across four distinct classification models, and the corresponding performance metrics were compared to estimate their impact. Tables 2 to 5 provide an in-depth insight into the performance of each classifier under various preprocessing scenarios.

In Table 2, the second row, labeled "Unbalanced Data", represents the evaluation metrics computed on the original imbalanced dataset. This dataset comprises 39,922 majority

class instances and 5,289 minority class instances. To address this imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE), as depicted in the third row. SMOTE effectively increased the minority class instances to 19,961, resulting in the associated metrics.

Table 2 includes "SMOTE + Tomek-Links" in the fourth row, which illustrates data that has been resampled using both SMOTE and Tomek-Links exclusively on the minority class. This resampling process culminated in 19,795 minority records.

For the fifth row of Table 2 , designated as "Undersampled", we randomly undersampled the majority class, yielding 12,900 majority and 5,289 minority class instances.

In Table 2, the sixth row, "Overall Balanced", showcases instances that underwent a two-step resampling process: oversampling the minority class first and then undersampling the majority class. This hybrid approach resulted in a balanced dataset comprising 19,961 instances in both classes. This balanced dataset underwent further refinement through Tomek-Links, as shown in the seventh row, labeled "Overall Balanced + Tomek-Links", resulting in 19,831 instances.

The last row of Table 2, labeled "Overall Balanced + Tomek-Links + GP", provides metrics for a classifier trained on the overall balanced and cleaned data, incorporating an optimal set of input features selected through the Genetic Programming (GP) algorithm.

Tables 3, 4, and 5 mirror the same resampled datasets as Table 2, presenting performance metrics for Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) classifiers, respectively. These comprehensive assessments aim to show the impact of our proposed techniques on diverse classification models and provide valuable insights into their efficacy.

All four classification algorithms showed improved performance when trained on balanced data compared to the original unbalanced dataset. It is observed that the Decision Tree model performed the best when trained on oversampled, undersampled, and cleaned data. Table 2 shows that decision tree classifier has highest balanced accuracy, recall, F1 score and area under ROC curve for the proposed technique, which is 0.8834, 0.8828, 0.8831 and 0.8834 respectively. Comparing the recall and F1 score of the proposed model to the original data reveals that these values nearly doubled, indicating better prediction by the model. Among the four classification models, the KNN classifier exhibited the highest recall and F1 score of 0.94 and 0.9, respectively.

Table 2. .Evaluation Metrics for Decision Tree Classifier

Decision Tree	Accuracy	Balanced Accuracy	Recall	F1 Score	roc_auc_score
Unbalanced Data	0.8729	0.7007	0.4755	0.4684	0.7007
SMOTE	0.8869	0.8748	0.8386	0.8318	0.8748
SMOTE + TOMEK-Links	**0.8882**	0.8757	0.8384	0.8330	0.8757
Undersampling	0.8081	0.7676	0.6751	0.6644	0.7676
Undersampling + TOMEK-Links	0.8247	0.7887	0.7000	0.7034	0.7887
Overall Balanced	0.8737	0.8737	0.8754	0.8740	0.8737
Overall Balanced + TOMEK-Links	0.8771	0.8771	0.8760	0.8767	0.8771
Overall Balanced + TOMEK-Links + GP	0.8834	**0.8834**	**0.8828**	**0.8831**	**0.8834**

5 CONCLUSIONS

In this research, we have introduced a robust data preprocessing technique designed to effectively address the challenges posed by imbalanced datasets. This innovative approach

Table 3. Evaluation Metrics for Logistic Regression Classifier

Logistic Regression	Accuracy	Balanced Accuracy	Recall	F1 Score	roc_auc_score
Unbalanced Data	**0.8984**	0.6419	0.3110	0.4105	0.6419
SMOTE	0.8196	0.7817	0.6680	0.7118	0.7817
SMOTE + TOMEK-Links	0.8236	0.7842	0.6667	0.7154	0.7842
Undersampling	0.8287	0.7580	0.5964	0.6621	0.7580
Undersampling + TOMEK-Links	0.8425	0.7861	0.6475	0.7093	0.7861
Overall Balanced	0.8308	**0.8308**	**0.8274**	**0.8303**	**0.8308**
Overall Balanced + TOMEK-Links	0.8305	0.8304	0.8234	0.8289	0.8304
Overall Balanced + TOMEK-Links + GP	0.8301	0.8301	0.8229	0.8285	0.8301

Table 4. Evaluation Metrics for KNN Classifier

KNN	Accuracy	Balanced Accuracy	Recall	F1 Score	roc_auc_score
Unbalanced Data	0.8899	0.6551	0.3482	0.4267	0.6551
SMOTE	0.9023	0.9019	0.9007	0.8601	0.9019
SMOTE + TOMEK-Links	**0.9054**	**0.9056**	0.9062	0.8643	**0.9056**
Undersampling	0.8196	0.7516	0.5969	0.6501	0.7516
Undersampling + TOMEK-Links	0.8358	0.7789	0.6388	0.6979	0.7789
Overall Balanced	0.8968	0.8968	**0.9410**	0.9013	0.8968
Overall Balanced + TOMEK-Links	0.8974	0.8975	0.9408	**0.9014**	0.8975
Overall Balanced + TOMEK-Links + GP	0.8930	0.8931	0.9261	0.8962	0.8931

Table 5. Evaluation Metrics for SVM Classifier

SVM	Accuracy	Balanced Accuracy	Recall	F1 Score	roc_auc_score
Unbalanced Data	**0.8956**	0.6195	0.2584	0.3682	0.6195
SMOTE	0.8168	0.7736	0.6438	0.7011	0.7736
SMOTE + TOMEK-Links	0.8203	0.7747	0.6384	0.7026	0.7747
Undersampling	0.8261	0.7466	0.5647	0.6463	0.7466
Undersampling + TOMEK-Links	0.8380	0.7748	0.6194	0.6942	0.7748
Overall Balanced	0.8294	0.8294	**0.8195**	0.8278	0.8294
Overall Balanced + TOMEK-Links	0.8298	0.8298	0.8179	0.8274	0.8298
Overall Balanced + TOMEK-Links + GP	0.8302	**0.8302**	0.8181	**0.8279**	**0.8302**

combines oversampling, undersampling, data cleaning, and feature selection to mitigate class imbalance, providing an effective solution for improved model performance.

To evaluate the efficacy of our proposed method, we conducted extensive experiments on a real-world Bank Marketing dataset. This involved training four classification models on various balanced datasets generated through our preprocessing technique. We employed a suite of metrics tailored for imbalanced datasets, including recall, F-1 score, balanced accuracy, and the area under the ROC curve, to comprehensively assess the performance of both existing preprocessing methods and our proposed approach.

Our results illustrate that employing both oversampling and undersampling at the same time as part of our data preprocessing technique consistently enhances the performance of all four classifiers. The Decision Tree and K-Nearest Neighbors (KNN) algorithms emerged as top performers, achieving the highest balanced accuracy rates of 88.34% and 89.31%, respectively. These discoveries emphasize the efficacy of our approach on addressing the class imbalance challenge.

There are many compelling paths for further investigation. One potential direction involves combining our data-level preprocessing technique with algorithmic-level approaches, such as ensemble machine learning classifiers, to leverage their complementary strengths. Additionally, the integration of genetic programming, which we used for feature selection, could be extended to data-level preprocessing tasks, offering the potential to further enhance model performance. This research lays the foundation for continued advancements in the field of imbalanced dataset handling, with the aim of improving the robustness and reliability of machine learning classification models in practical applications.

References

1. Shuo Wang, Leandro L. Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4802–4821, 2018.
2. Satwik Mishra. Handling imbalanced data: Smote vs. random undersampling. *International Research Journal of Engineering and Technology*, 4(8):317–320, 2017.
3. T Elhassan and M Aljurf. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Global Journal of Technology and Optimization*, 1:1–11, 2016.
4. Wei Chao Lin, Chih Fong Tsai, Ya Han Hu, and Jing Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26, 2017.
5. William B Langdon and Riccardo Poli. *Foundations of genetic programming*. Springer Science & Business Media, 2013.
6. Jianbin Ma and Xiaoying Gao. A filter-based feature construction and feature selection approach for classification using genetic programming. *Knowledge-Based Systems*, 196:105806, 2020.
7. S. Moro, P. Rita, and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.
8. Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
9. Tawfiq Hasanin and Taghi Khoshgoftaar. The effects of random undersampling with simulated class imbalance for big data. In *2018 IEEE International Conference on Information Reuse and Integration*, pages 70–79. IEEE, 2018.
10. Jason Brownlee. Smote for imbalanced classification with python. *Machine Learning Mastery*, 16, 2020.
11. Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(2):539–550, 2008.
12. I Tomek. Two modifications of condensed nearest neighbors. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:769–772, 1976.
13. Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

14. Chien Hsing Chou, Bo Han Kuo, and Fu Chang. The generalized condensed nearest neighbor rule as a data reduction method. In *The 18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 556–559. IEEE, 2006.
15. Jason Brownlee. Undersampling algorithms for imbalanced classification. *Machine Learning Mastrey*, 27, 2020.
16. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.
17. Natalia Grafeeva, Lyudmila Grigorieva, and Natalia Kalinina-Shuvalova. Genetic algorithms and genetic programming. *International Journal of Advanced Computer Science and Applications*, 3(9):465–480, 2013.
18. Il Seok Oh, Jin Seon Lee, and Byung Ro Moon. Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1424–1437, 2004.
19. Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT press, 1998.
20. Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39:261–283, 2013.
21. Yifan Lu, Tianle Ye, and Jiali Zheng. Decision tree algorithm in machine learning. In *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pages 1014–1017. IEEE, 2022.
22. Vikas Jain, Ashish Phophalia, and Jignesh S Bhatt. Investigation of a joint splitting criteria for decision tree classifier use of information gain and gini index. In *TENCON 2018 - IEEE Region 10 Conference*, pages 2187–2192. IEEE, 2018.
23. Ming Yih Lee and Chi Shih Yang. Entropy-based feature extraction and decision tree induction for breast cancer diagnosis with standardized thermograph images. *Computer Methods and Programs in Biomedicine*, 100(3):269–282, 2010.
24. R. E. Wright. Logistic regression. *Reading and Understanding Multivariate Statistics*, pages 217–244, 1995.
25. Sonia Domínguez-Almendros, Nicolás Benítez-Parejo, and Amanda Rocío Gonzalez-Ramirez. Logistic regression models. *Allergologia et immunopathologia*, 39(5):295–305, 2011.
26. Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2003)*, pages 986–996. Springer, 2003.
27. Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, and Nittaya Kerdprasop. An empirical study of distance metrics for k-nearest neighbor algorithm. In *Proceedings of the 3rd International Conference on Industrial Application Engineering*, 2015.
28. Yong Li Zhang. Support vector machine cclassification algorithm and its application. *Information Computing and Applications*, pages 179–186, 2012.
29. Yukihiro Hamasuna, Yasunori Endo, and Sadaaki Miyamoto. Support vector machine for data with tolerance based on hard-margin and soft-margin. In *2008 IEEE International Conference on Fuzzy Systems*, pages 750–755. IEEE, 2008.
30. Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining Knowledge Discovery*, 28(1):92–122, 2014.
31. Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *The 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE, 2010.
32. Hongwei Shang, Jean-Marc Langlois, Kostas Tsioutsiouliklis, and Changsung Kang. Precision/recall on imbalanced test data. In *International Conference on Artificial Intelligence and Statistics*, pages 9879–9891. PMLR, 2023.
33. Antonio Maratea, Alfredo Petrosino, and Mario Manzo. Adjusted f-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257:331–341, 2014.
34. Christopher D Brown and Herbert T Davis. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1):24–38, 2006.
35. Caren Marzban. The roc curve and the area under it as performance measures. *Weather and Forecasting*, 19(6):1106–1114, 2004.
36. Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, 2006.
37. Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de Las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

Authors

Asitha Thumpati received her Master degree in Computer Science from California State University San Bernardino in the Spring 2023.

Yan Zhang currently is an associate professor at School of Computer Science and Engineering, California State University San Bernardino. She received her Ph.D. degreed in Computer Science from University of Regina, Canada. Dr. Zhang's research interests include machine learning, uncertain data analysis, and game theory.