Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 103–112

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

A Factor Based Multiple Imputation Approach to Handle Class Imbalance

Pranita Baro[a], Malaya Dutta Borah[a]

[a]*Dept. of Computer Science and Engineering, National Institute of Technology Silchar, Silchar-788010, India*

Abstract

Class imbalance and incompleteness are the two most serious problems faced in data science and machine learning when working on real-life datasets. Both of these cases have severe implications on the ability of classification algorithms to make accurate predictions. When a dataset used for training classifiers is both imbalanced as well as incomplete, the traditional approach is to address the missing data first and then handle class imbalance but it could lead to some issues such as overfitting as well as amplification of some errors due to random duplication. In this paper, an alternate factor-based multiple imputation oversampling method (FB-MIO) is proposed to handle class imbalance as well as missing values in the training dataset at the same time. First, a new factor is presented to evaluate the density of missing values belonging to the majority class with respect to the minority class in a particular region. With the help of this factor, an oscillator is developed to guide how imputation based oversampling should be carried out. Then the training set is divided into multiple smaller subsets and used the oscillator to determine whether missing values for the majority class belonging to that subsets should be imputed or not. This would help in preventing exaggerated duplication when not needed. Experiments were carried out on 27 imbalanced datasets after random addition of missing value and the F1 and AUROC scores of FB-MIO were compared to other dataset level resampling methods such as SMOTE, ADASYN, B-SMOTE etc. The effectiveness of the proposed method has been validated after experiments on benchmark datasets and the comparative results are presented in the form of average rank and number of wins.

Keywords: class imbalance; overfitting; oversampling; imputation; missing value

1. Introduction

Class imbalance is often a challenging problem in data science and machine learning. Combined with the incompleteness of data, class imbalance is one of the reasons for severely inhibiting the performance and effectiveness

* Corresponding author: Pranita Baro
E-mail address: baro.pranita92@gmail.com

1877-0509
10.1016/j.procs.2022.12.406

of classification algorithms. A dataset is considered imbalanced when the data samples of different classes vary in quantity. When working on real-life-based data or datasets, it is not rare to come across situations when data points belonging to the majority class comprise 99.99% of the total data, and the minority class data points only amount to 0.01%. Most of the existing classification methods are designed based on the assumption that there is an even distribution of the classes in a given training dataset [1], which is usually not the case. Moreover, they tend to underperform and make errors in their prediction when trained on an imbalanced dataset due to a bias towards the majority class. That is why it is important to properly balance classes in a training dataset before fetching it to a classifier. When working with real-life-based datasets, it is prevalent having to work with unbalanced datasets that also have chunks of missing data, which further increases the difficulty of handling class imbalance and accurate prediction by the classifiers. Some common reasons for the incompleteness of data are data collection issues, sampling measurement issues, human mistakes, and loss of information during transit. Like class imbalance, most classification methods are not adequately equipped to handle chunks of data missing in training or testing datasets. Moreover, depending on how much data is missing, it could significantly impact performance. So missing values in a dataset used for machine learning must be addressed appropriately before being fed to a classifier. The missing values in a dataset can be dealt with by either removing all the required missing values and even entire features if needed or by substituting missing values with statistically or algorithmically generated values. Oversampling and undersampling are some of the most popular data resampling methods which balance the distribution of the classes at the dataset level. Both approaches have their pros and cons. The approach depends on the usability, and whether it is pragmatic to be used depends on several factors such as data composition, data distribution, dataset size, etc. For handling missing values in a dataset, imputation techniques can be beneficial and are pretty much industry standard. A hybrid approach of imputation followed by oversampling might look appealing to handle incompleteness of data first and then class imbalance. However, since this entire process is carried out in sequence, imputation accuracy will impact performance on oversampling; also, any errors during imputation will be amplified during oversampling, which could cause significant data distortion.

This paper presents a systematic factor-based multiple imputation oversampling method (FB-MIO) for simultaneous handling of class imbalance and missing values in the training dataset. First, a factor is represented based on the number of missing values belonging to the majority class concerning the minority class. Based on that, a simple formula for an oscillator is introduced. The oscillator is cyclical, and its value ranges from 0 to 100. Then, multiple imputation-based oversampling approaches are applied to simultaneously handle class imbalance and missing values in the training dataset. The oversampling approach will take note of the current value of the oscillator to decide on the course of action depending on the current value of the oscillator. Our proposed method was tested on 27 real-life datasets [2] and performed above the benchmark. The main contributions of this paper are:

- A systematic factor-based multiple imputation oversampling techniques that incorporates oscillator as a guide to handle missing data and class imbalance.
- Performed better when a dataset is imbalanced as well as significantly observed loss of data.

1.1. Motivation

Class imbalance and incompleteness of data are often faced when working with real-life data in real-time, and they tend to have a noticeable impact on the accuracy of classification methods. It is always clear what approach should be taken to remedy the situation on the fly without a broad analysis of the incoming data. Hence there is a need for a systematic approach to the whole process where an indicator is used to determine the next course of action to clean, reshuffle and reshape the data without having much impact on class boundaries and information contained within the data. The objectives for this paper are:

- To handle missing values as well as class imbalance in a dataset at the same time.
- To develop an indicator-based approach to the class imbalance that considers the degree of incompleteness of data at a class level and provides guidance for an appropriate course of action.

1.2. Problem Statement

To design and develop a factor-based multiple imputation oversampling technique that is effective against class imbalance and incompleteness of data while respecting the class boundary in the original dataset. The rest of the paper is organized as follows:

- Section 2 represents the related study of the work.
- Section 3 explains the proposed method of the work.
- Section 4 & 5 discusses the dataset and the experimental analysis.
- Finally in Section 6 & 7 the paper is concluded, describes the discussion and conclusion.

2. Literature Survey

This section aims to give a brief overview of the various researches done by the researchers and institutions related to our work. Our first focus is on the oversampling technique over the years and our inferences based on it. Then move over the progress made over learning from incomplete data and implementing imputation methods.

2.1. Addressing class imbalance at dataset level

There are two common approaches to class imbalance on a training dataset [3]. The first is at the dataset level before feeding the data to the classifier, and the second is at the algorithm level. Dataset-level approaches such as oversampling tend to take the path of data manipulation, while algorithm-based approaches emphasize minority class during classification. To address class imbalance at the dataset level, we applied data point removal methods for the majority class, such as undersampling, as it could lead to further loss of information in a dataset with incomplete data. In [4] the authors have gone through different ensemble learning techniques and done a thorough analysis for handling class imbalance problem. [5] provided a brief overview of various synthetic data point generation-based oversampling approaches and derived the conclusion that the suitability of a method on a particular dataset depends on the dataset characteristics. There are no best methods for handling class imbalance. [6] improvised a new method for creating synthetic data from the original data pool by exporting essential features from the original data and generating synthetic data based on that. Synthetic minority oversampling technique (SMOTE) [7] is arguably not only in terms of oversampling but probably the most widely used technique to handle the class imbalance. SMOTE works by randomly selecting minority class data points and one of its k-nearest neighbor minority class instances and then generating a synthetic data point by a random convex combination of the two selected data points to boost the total count of data points belonging to the minority class. Many variants of SMOTE are in use today, such as ADASYN and B-SMOTE. Adaptive synthetic sampling (ADASYN) [8] is designed to consider the density of nearby majority class-based data points and generates more synthetic minority class data points for each minority class. It creates more synthetic minority class data points in the areas where the classifiers have trouble learning and respecting the boundary between classes. ADASYN is said to be an improvement over SMOTE as SMOTE is just random oversampling all over the training set. In contrast, ADASYN is more precise and only targets the areas with a high majority class data sample density. B-SMOTE (also known as Border-SMOTE) [9] is another popular oversampling technique based on SMOTE and is similar to ADASYN in that they target the decision boundary between classes and generate synthetic data belonging to minority classes. B-SMOTE reduces the difficulty of learning for the classifiers on the borders.

2.2. Addressing incompleteness of data

Similar to class imbalance, the incompleteness of data is often seen in real-life-based datasets. There are various unavoidable situations for the incompleteness of data, for example, data collection error, human error, error in transmission, etc. It can be a lot more challenging to handle than class imbalance, and if a dataset is an imbalance and many data points belonging to a minority class are missing, the classifier's accuracy will go down a notch further; hence it is essential to treat this problem before the data is used for classifier's training, ideally during the data processing stage at the same time as handling of class imbalance. There are two standard methods of dealing with missing values in a

dataset. The first is removing all the missing value data points and even the containing features if necessary. However, this could lead to critical information loss and is unsuitable if the missing value percentage is very high, say 50%. The other method is to substitute the missing values with new data samples. New data samples could be generated either with statistical techniques such as simply replacing the missing data points with values from the data pool or within the dataset, similar or with algorithmic techniques that generate synthetic data points. To deal with missing values and incomplete data, imputation techniques are the most common technologies. Imputation is the method of substituting the missing values in a dataset with one or more synthetic values based on the features and characteristics of observed values. Depending on the number of substituted values, imputation divided into single and multiple imputations. Examples of single imputation are model-based imputation, mean imputation, and expectation-maximization imputation, to name a few [10, 11]. Multiple imputations generate many data samples for each missing value and consider the variability associated with imputation [12]. Most popular methods for multiple imputations are multiple imputations using chained equations (MICE) and multivariate normal imputation (MVNI). MVNI works under the assumption that all features fall under a multivariate normal distribution [13] whereas MICE works by assuming that the data is missing at random [14]. MICE focuses on one variable at a time it takes into account the observed values from a dataset or a set of prearranged values to come up with the missingness of that variable. MICE prediction is based on regression models, whether linear or logistic regression, based on the data's nature. The distribution of data has little impact on MICE which makes it flexible for a broad range of uses. In this paper [15, 16] we'll be working with and taking advantage of the flexibility of MICE. In [17] two single and multiple imputation based novel oversampling methods are proposed, that create valid synthetic minority class data points by estimating missing values already induced in the minority class samples. [18] developed a multiple imputation-based minority oversampling approach (MI-MOTE) to handle class imbalance and incompleteness of data. Most data points are once imputed, and without their observed values minority instances are oversampled using multiple imputations. In this paper, the MICE methodology is used to help in oversampling and substituting missing values.

3. Proposed method

The proposed method FB-MIO is multiple imputation-based oversampling techniques that aim to simultaneously handle class imbalance and incompleteness of data. It uses an oscillator derived from missing data points as an indicator to decide on the current course of action. The following will give an overview and describe the inner workings of FB-MIO.

3.1. Input

Let us assume that we have the training dataset is D with m numbers of samples in the form of $D = (X_i, y_i)_{i=1}^{N}$ where X_i is a data instance in the N-dimensional feature space X, and y_i is the class label for X_i data point instance. This paper is based on working with binary class datasets, hence $y_i \in 0, 1$ where 0 denotes data points belonging to the majority class and $y_i = 1$ for the minority class. Due to the incompleteness of data, $X_{i,j}$ for some features are not observable, hence categorized as a missing value. Let the user-defined oversampling ratio be λ, and multiple imputers are M which is based on M. The classifier in use will be denoted with f as output.

3.2. The incompleteness factor and the oscillator

When a dataset has both imbalances and some missing value, both are inextricably linked, and it is not wise to treat class imbalance without considering the incompleteness of data. The problem is that if all the missing values are substituted with new synthetic data points with the help of imputation techniques, it could further tilt the balance of classes and make more portions of the dataset hard to learn. Most of the time, the classifiers are more concerned about the minority class as that is the class we want to predict or target. Hence in addition to the number of minority class data samples than that of the majority class, several missing values of the majority class concerning a minority is also essential. The incompleteness factor, also called I-factor, is introduced, which would help find the middle ground between dealing with class imbalance as well as the incompleteness of data.

$$I = \frac{m_0 + m_1}{m_1} \tag{1}$$

where I is an incomplete factor. m_0 is several missing values belonging to the majority class in an area. m_1 is several missing values belonging to the minority class in the area. While the incompleteness (I) factor alone is of little use, expect to show how much missing data belongs to the majority class relative to the minority class. However, this information is of little use alone. The value of the I-factor can be put to better use when it is expressed on a scale from 0 to 100. That way, it would be easier to carry out some action depending on the resultant value. With that in mind, the incompleteness oscillator is represented below, which can be used as an indicator in the follow-up oversampling and imputation procedure.

$$I_0 = 100 - \frac{100}{1 + \frac{m_0}{m_1}} \tag{2}$$

or in terms of I-factor the above formula can be re-written as

$$I_0 = 100 - \frac{100}{I} \tag{3}$$

where I_0 is the incompleteness oscillator and $0 < I_0 < 100$

After repeated experimentations with the incompleteness oscillator and trying to come up with a combination of the oscillator and imputation-based minority sample oversampling techniques, and came up with the following rules. These rules will act as a guide in performing minority sample oversampling when a training dataset is imbalanced and missing some value.

- If $I_0 >= 70$, it shows that in that particular area, the number of missing values belonging to the majority class outnumber those of the minority class. Hence this area is challenging to learn for the classifiers. The majority class missing values should be left alone, and minority class data points will be imputed λ times where λ is the user-provided oversampling rate.

- If $I_0 <= 30$, it shows that in that particular area, the number of missing values belonging to the minority class outnumber those of the majority class. In order to seek a balanced approach and prevent loss of information, both minority and majority class data points will be imputed with the help of multiple imputer λ times.

- If $30 < I_0 < 70$, all majority data points for a given neighborhood will be imputed once, and those belonging to the minority class will be imputed λ times.

- If $I_0 = 0$, it means there are no missing values belonging to the majority class. Hence, only minority class data points will be imputed $(\lambda - 1)$ times. There is no need to be concerned with the majority class as they have no missing values and want to seek a balanced approach to dealing with class imbalance as well as incompleteness of data.

- If $I_0 = 100$, it means there are no missing values belonging to the minor class. Hence only minority class data points will be imputed λ times, which in the absence of missing values will be the same as simple synthetic replication.

With the above rules in place, the method for FB-MIO is presented in Algorithm 1.

Algorithm 1: FB-MIO Algorithm

Inputs: The k-nearest neighborhood S which is subset of training dataset D that contain N number of data points, oversampling ratio λ, incompleteness oscillator I_0.

Output: MICE based multiple imputer M, classifier f.

Procedure:

$S_1 \leftarrow \{(X_i, y_i) \mid (X_i, y_i) \in S \text{ and } y_i = 1\}$

$S_0 \leftarrow \{(X_i, y_i) \mid (X_i, y_i) \in S \text{ and } y_i = 0\}$

if $I_o == 0$ **then**
- $X_i^0, X_i^1, ..., X_i^{\lambda-1}$ synthetic replication of $X_i \; \forall \; (X_i, y_i) \in S_1$
- $X_i^1 \leftarrow M(X_i^1) \; \forall (X_i, y_i) \in S_1$ where $I = 0, 1, ..., \lambda - 1$
- $S^1 \leftarrow (X_i^I, y_i) \mid (X_i, y_i) \in S_1$ and $I = 0, 1, ..., \lambda - 1$

else if $I_o <= 30$ **then**
- $X_i^0, X_i^1, \ldots, X_i^\lambda$, synthetic replication of $X_i \; \forall (X_i, y_i) \in S_1$
- $X_i^l \leftarrow M(X_i^l) \; \forall (X_i, y_i) \in S_1$ where $l = 0, 1, ..., \lambda$
- $S^1 \leftarrow (X_i^l, y_i) | (X_i, y_i) \in S_1$ and $l = 0, 1, ..., \lambda$
- $X_i^0, X_i^1, \ldots, X_i^\lambda$ synthetic replication of $X_i \; \forall \; (X_i, y_i) \in S_0$
- $X_i^l \leftarrow M(X_i^l) \; \forall (X_i, y_i) \in S_0$ where $l = 0, 1, ..., \lambda$
- $S^0 \leftarrow (X_i^l, y_i) \mid (X_i, y_i) \in S_0$ and $l = 0, 1, ..., \lambda$

else if $I_o >= 70$ **then**
- $X_i^0, X_i^1, \ldots, X_i^\lambda$ synthetic replication of $X_i \; \forall (X_i, y_i) \in S_1$
- $X_i^l \leftarrow M(X_i^l) \; \forall (X_i, y_i) \in S_1$ where $l = 0, 1, ..., \lambda$
- $S^1 \leftarrow (X_i^l, y_i) \mid (X_i, y_i) \in S_1$ and $l = 0, 1, ..., \lambda$

else
- $X^i \leftarrow M(X_i) \; \forall (X_i, y_i) \in S_0$
- $S^0 \leftarrow (X^i, y_i) \mid (X_i, y_i) \in S_0$
- $X_i^0, X_i^1, \ldots, X_i^\lambda$ synthetic replication of $X_i \; \forall (X_i, y_i) \in S_1$
- $X_i^l \leftarrow M(X_i^l) \; \forall (X_i, y_i) \in S_1$ where $l = 0, 1, ..., \lambda$
- $S^1 \leftarrow (X_i^l, y_i) \mid (X_i, y_i) \in S_1$ and $l = 0, 1, ..., \lambda$

$S' = S^0 \cup S^1$

End procedure

The training dataset D is divided into multiple regions. Different actions will be taken depending on the oscillator's various outputs. Let S represent the neighborhood in which the missing value is present. S is a subset of original dataset D where $|S| = n$ and λ is the user-defined oversampling ratio. M denotes the multiple imputers based on the MICE methodology. S will be further divided into two more subsets of data points belonging to majority class S_0 and minority class S_1 respectively. One primary assumption of FB-MIO is that the dataset contains missing values. In the absence of missing value, the synthetic data generation process will be similar to simple replication, leading to overfitting. However, the oscillator ensures that the user stays within the limit and does not just blindly replicate all data points, causing the situation to deteriorate. When there are no missing values belonging to the majority class in

S, i.e., $I_o = 0$ and S_0 will remain as it is. By following the rules of the oscillator stated above, we will first create a $\lambda - 1$ copy of input vector X_i where $(X_i, y_i) \in S_0$, resulting in λ copies of the input vector. Then each of these replicated copies will be run through the imputer M, resulting in $X_i^l = M(X_i^l)$ where $l = 0, 1, ..., \lambda - 1$. If the input vector has a missing value, then because of randomness, multiple imputations will result in different synthetic data points. The same process above is carried out for $I_o >= 70$, except one more extra copy of the minority data point will be created, resulting in one more copy of the imputed value. The same is true for the extreme upper value of I_o, i.e., 100. This way, imputation based oversampling can be carried out without manipulating original data. When $I_o <= 30$ but not 0, it means missing values of the minority class outnumber those of the majority class. So we seek a balanced approach to handling both class imbalance and missing value and make lambda copies of data points belonging to both classes and then feed them to imputer M. The resulting minority class data points are stored in S^1 and those in majority class are stored in S^0. For all other cases, i.e., $30 < I_o < 70$, the above process would still be carried out. However, most instances will only be imputed once, and minority data points will be imputed as usual λ times by making lambda copies and running them through imputer M.

4. Dataset

The proposed method is designed for applying in various fields of study such as biology, banking, traffic, etc. When a dataset is based on a real-life situation, it is common to observe imbalance in class and incompleteness in data. As our approach is to minimize the effect of class imbalance in binary classification problems, it is necessary to choose appropriate datasets for experimentations and benchmarks deemed suitable for the given task. Hence, keeping the above observations in mind, 27 datasets have been chosen that are part of the sklearn imbalanced-learn package [2]. Imbalanced-learn is an open-source, MIT-licensed library that provides tools for classifying imbalanced classes, including the datasets used for this paper. These datasets are based on diverse research fields and are heavily imbalanced, i.e., data samples from the majority class vastly outnumber those from the minority class. The table below gives an excellent overview of the datasets used, their origin, features, and characteristics. These chosen datasets originally had no missing values, but to simulate the incompleteness of data one might encounter when working on real-life data. Some missing values were introduced, and based on each chosen dataset, three new ones with missing values were created, with missing value percentages of 10%, 30%, and 50%, respectively. For consistency, every continuous feature was scaled so that they had a mean value of 0 and a variance of 1. One hot-encoding was applied to each categorical feature as the resultant dimensions.

5. Experimental Analysis

The proposed FB-MIO approach to imbalanced classification and incomplete data uses the MICE method for fitting the imputer M. FB-MIO was compared to five approaches that do both oversampling and imputation in a sequential fashion: basic with no oversampling, random oversampling, SMOTE, B-SMOTE and ADASYN. The primary configurations used during the experiment were oversampling ratio as 5, modeling of conditional feature distribution by Bayesian ridge regression, round-robin iteration for multiple imputations was set to 10, and the remaining configurations regarding oversampling were kept at default in the imbalanced-learn package. Three methods were chosen from the scikit-learn package for the classification part following resampling. They are random forest [19], logistic regression [20] and neural network [21]. During the training stage, a five-fold cross-validation methodology was adopted. The used dataset is separated into five-folds, and one fold is chosen as testing set, the remainder are combined to form a training set. This process will be continued until every fold is used as a testing set.

The experiments were carried out on the 27 datasets after randomly adding missing value percentages of 10%, 30%, and 50%, respectively. The following Tables 1 & 2 illustrate the performance of different methods and are calculated based on F1-score and AUROC. Inside the following tables, the performance summary is presented in the form of the average ranking and the number of wins. The results illustrated that FB-MIO successfully dealt with the class imbalance and missing data. Moreover, it maintains a small amount of data distortion at the same time. It was seen that when the missing value percent was not high, FB-MIO's performance was similar to other methods. However, it outperformed when the missing rate was higher, as using the imputer for oversampling and dealing with missing value aided in better classification. Figure 1 shows the impact of missing value on different performance metrics.

Table 1: **The performance measure of F1-score based on Average rank and No. of wins**

Miss_rate	classifier	statistics	base	random	SMOTE	B-SMOTE	ADASYN	FB-MIO
10%	RF	avg. rank	5.63	3.95	2.5	3.34	3.56	1.8
		No. wins	0	1	4	2	4	14
	LR	avg. rank	5.13	4.72	2.12	2.96	3.38	3.2
		No. wins	0	2	11	3	2	9
	NN	avg. rank	5.22	5.07	2.81	3.7	3.15	2.3
		No. wins	1	3	6	4	5	9
30%	RF	avg. rank	5.91	4.57	2.21	3.34	3.57	1.2
		No. wins	0	0	2	1	1	23
	LR	avg. rank	5.03	4.82	3.37	3.91	3.15	2.22
		No. wins	2	2	4	0	5	14
	NN	avg. rank	4.99	4.06	3.8	3.2	3.63	1.55
		No. of wins	1	1	3	2	3	17
50%	RF	avg. rank	5.58	4.75	3.4	2.68	2.97	1.32
		No. wins	0	0	0	3	1	23
	LR	avg. rank	4.55	5.1	3.25	3.08	2.97	2.32
		No. wins	2	0	3	3	5	14
	NN	avg. rank	4.29	3.47	4.14	3.2	3.32	2
		No. wins	0	2	0	4	2	19

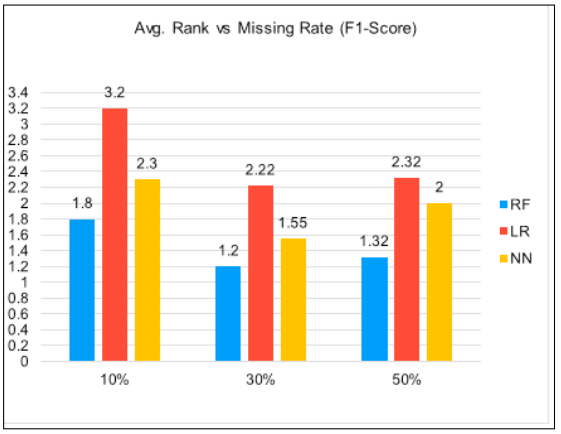
Table 2: **The performance measure of AUROC based on Average rank and No. of wins**

Miss_rate	classifier	statistics	base	random	SMOTE	B-SMOTE	ADASYN	FB-MIO
10%	RF	avg. rank	5.21	4.23	2.87	3.6	3.41	2.42
		No. wins	1	1	7	2	3	13
	LR	avg. rank	3.38	3.7	3.14	4.32	2.69	2.37
		No. wins	3	2	3	2	6	11
	NN	avg. rank	4.77	3.9	2.58	4.14	2.96	2.1
		No. wins	4	3	4	2	4	10
30%	RF	avg. rank	5.7	4.29	3.17	3.49	3.66	1.32
		No. wins	0	0	2	1	1	23
	LR	avg. rank	3.85	4.29	2.91	4.35	3.22	1.41
		No. wins	0	0	3	0	3	21
	NN	avg. rank	5.12	4.2	3.27	3.95	3.59	1.91
		No. wins	1	1	3	2	3	17
50%	RF	avg. rank	5.74	3.42	2.79	3.67	3.81	1.13
		No. wins	0	0	3	0	2	22
	LR	avg. rank	3.67	3.78	4.53	4.04	2.43	1.57
		No. wins	0	0	0	0	4	23
	NN	avg. rank	4.61	3.8	3.78	3.39	3.17	1.28
		No. wins	2	1	0	2	3	19

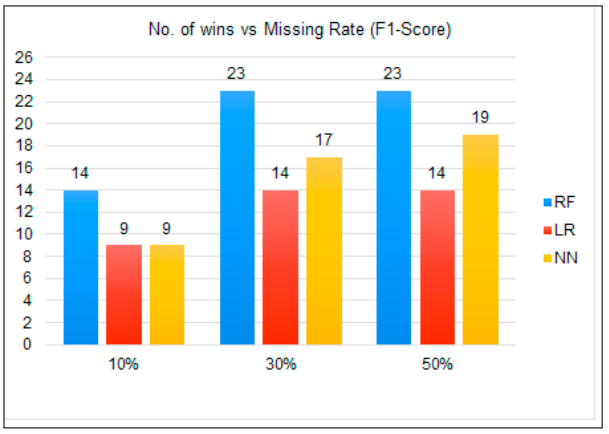
6. Discussion

It might seem that the simplest and most direct way to handle class imbalance and incompleteness of data simultaneously is first to substitute the missing values with imputation techniques and then use dataset-level methods such as SMOTE, ADASYN, B-SMOTE, etc. It violates the DRY principle by generating synthetic data at two stages, first

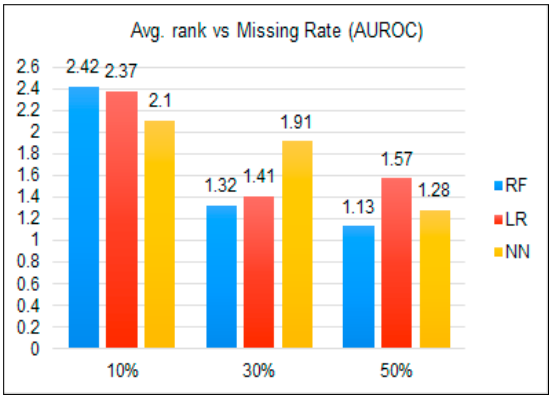
during imputation and second during oversampling. The approach mentioned above has a serious flaw. Let us say there was an error during imputation, resulting in some erroneous minority class synthetic data samples being generated. If those data points are used as a basis to generate more data points during minority class oversampling methods such as SMOTE, it could lead to amplification of those errors. At best, there are some errors in the new training dataset, and at worst, it could lead to data distortion, making the whole dataset unusable. Another demerit of this approach is that the combination of imputation and oversampling could lead to large numbers of synthetic data generation, which could cause overfitting. It is not easy to avoid this problem. The proposed FB-MIO approach does not handle original data directly. It handles imputation and oversampling simultaneously, thus, removing any chance of data distortion due to error amplification. Also, the oscillator used acts as a guide, trying to strive for a balance between dealing with the class imbalance and handling missing data. Since it guarantees that only required data points will be added, it dramatically reduces the chance of overfitting.



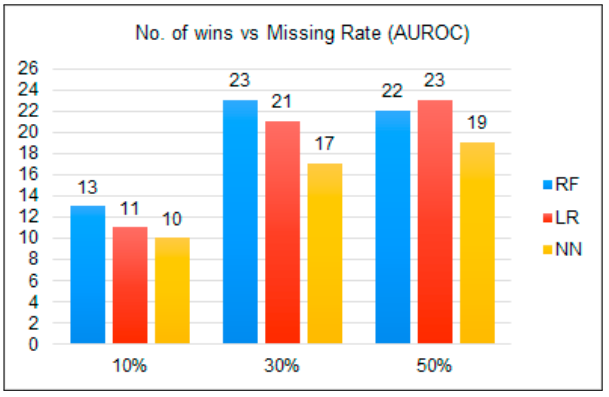
(a) F1_score_Avg.rank



(b) F1_score_no.of wins



(c) Auroc_Avg.rank



(d) Auroc_no.of wins

Fig. 1: The impact of missing value on several performance metrics: F1-score and AUROC

7. Conclusion and Future Work

Class imbalance, as well as incompleteness of data, takes the difficulty of training classification algorithms to another level. There has been lots of work done on treating both of these problems in a dataset. However, most of those works focus on first handling missing value with imputation and then dealing with class imbalance with resampling or algorithmic techniques. In this paper, a factor-based multiple imputation oversampling technique, FB-MIO, is presented that uses an oscillator as a guide for follow-up directions. The oscillator is based on the incompleteness factor (I-factor), also introduced in this paper. Then five rules are proposed regarding oscillators that act as a guide

on what approach to take for handling class imbalance and incompleteness of data based on the value shown on the oscillator. To verify the effectiveness of the proposed approach, we selected the set of 27 datasets that come with the imbalanced-learn package. These datasets are originally highly imbalanced and to further emulate real-life scenarios, three new datasets were created based on each of the 27 datasets and these new datasets contain random missing value rates of 10%, 30% and 50% respectively. Finally F1 score and AUROC were chosen as evaluation metrics and a five-fold cross validation approach was adopted to separate the datasets into training and testing sets. The subsequent benchmarking and experiment results proved our approach to be a better method for handling class imbalance and missing values in a dataset. Compared to traditional resampling methods such as SMOTE, ADASYN or B-SMOTE, FB-MIO performed a lot better when a dataset was imbalanced as well as significantly observed loss of data. For future work, we aim to use the oscillator method that is proposed in this paper for developing a hybrid resampling method. This hybrid method will combine two resampling methods and will use the oscillator as an indicator to come up with follow-through actions.

References

[1] Kumar, P., Bhatnagar, R., Gaur, K., & Bhatnagar, A. (2021, March). Classification of imbalanced data: review of methods and applications. In IOP Conference Series: Materials Science and Engineering (Vol. 1099, No. 1, p. 012077). IOP Publishing.
[2] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
[3] Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. Information Sciences, 513, 429-441.
[4] Singh, S., Kumar, P., Borah, M. D., Agrahari, A., & Baro, P. (2021). Ensemble Methods for Learning: An approach towards handling Class Imbalance and Class Overlapping Problems. In Interdisciplinary Research in Technology and Management (pp. 174-180). CRC Press.
[5] Santoso, B., Wijayanto, H., Notodiputro, K. A., & Sartono, B. (2017, March). Synthetic over sampling methods for handling class imbalanced problems: A review. In IOP conference series: earth and environmental science (Vol. 58, No. 1, p. 012031). IOP Publishing.
[6] Lundin, E., Kvarnström, H., & Jonsson, E. (2002, December). A synthetic fraud data generation methodology. In International Conference on Information and Communications Security (pp. 265-277). Springer, Berlin, Heidelberg.
[7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
[8] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE.
[9] Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Springer, Berlin, Heidelberg.
[10] Van Buuren, S. (2018). Flexible imputation of missing data. CRC press.
[11] García-Laencina, P. J., Sancho-Gómez, J. L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. Neural Computing and Applications, 19(2), 263-282.
[12] Hayati Rezvan, P., Lee, K. J., & Simpson, J. A. (2015). The rise of multiple imputation: a review of the reporting and implementation of the method in medical research. BMC medical research methodology, 15(1), 1-14.
[13] Schafer, J. L. (1997). Analysis of incomplete multivariate data. CRC press.
[14] Mera-Gaona, M., Neumann, U., Vargas-Canas, R., & López, D. M. (2021). Evaluating the impact of multivariate imputation by MICE in feature selection. Plos one, 16(7), e0254720.
[15] Murray, J. S. (2018). Multiple imputation: a review of practical and theoretical findings. Statistical Science, 33(2), 142-159.
[16] Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work?. International journal of methods in psychiatric research, 20(1), 40-49.
[17] Razavi-Far, R., Farajzadeh-Zanajni, M., Wang, B., Saif, M., & Chakrabarti, S. (2019). Imputation-based ensemble techniques for class imbalance learning. IEEE Transactions on Knowledge and Data Engineering, 33(5), 1988-2001.
[18] Shin, K., Han, J., & Kang, S. (2021). MI-MOTE: Multiple imputation-based minority oversampling technique for imbalanced and incomplete data classification. Information Sciences, 575, 80-89.
[19] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
[20] Maalouf, M. (2011). Logistic regression in data analysis: an overview. International Journal of Data Analysis Techniques and Strategies, 3(3), 281-299.
[21] Wan, E. A. (1990). Neural network classification: A Bayesian interpretation. IEEE Transactions on Neural Networks, 1(4), 303-305.