International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 7 (2019) pp. 1516-1525
© Research India Publications. http://www.ripublication.com

# An Emperical Study on the Effect of Resampling Techniques in Imbalaced Datasets for Improving Consistency of Classifiers

**Shidha M V**
*Research Scholar*
*Bharathiar University,*
*Coimbatore, India*

**T Mahalekshmi**
*Professor & Principal*
*Sree Narayana Institute of Technology*
*Kollam, India*

**Abstract**

Classification problems may sometimes get into trouble when one class outnumbers the other one. This situation is termed as class imbalance problem and unfortunately it is often found with real life problems where the decision is crucial in the identification of rare class. This causes big challenges for machine learning algorithms as they assume that dataset has balanced class distributions. Several techniques have been suggested to tackle with class imbalance problem and rebalancing of data sets is the one among them. This paper conducts an experimental study on the performance of different classifiers after balancing their data using different sampling techniques like SMOTE, ROS, ADASYN, RUS, CUS and NearMiss.

**Keywords:** classification, resampling, SMOTE, ROS, ADASYN, RUS, CUS, NearMiss.

## I. INTRODUCTION

In data mining, identification of an event is a classification problem. The prediction on rare events is often difficult due to their infrequency and low variety of information. Many real life problems face this problem, so that misclassification of rare class can incur a heavy penalty [1]. As an example, misclassification of a negative sample in a cancer prediction event may lead to additional laboratory tests while the inability to identify a positive sample makes a huge loss. These sorts of occurrences force to find solutions for class imbalance problem. Formally, imbalanced classification is a supervised learning problem in which number of samples in given classes is highly uneven as shown in Fig. 1.

With imbalanced data sets, an algorithm does not get the relevant information about the rare class to make an accurate prediction. Hence, it is desirable to use machine learning algorithms with balanced data sets. Many machine learning approaches have been developed in the past decade to work with imbalanced data classification. These approaches can be categorized into two groups: internal and external. The internal/algorithm level approaches deal with the design of new algorithms or the modification of existing ones so as to handle imbalanced datasets [2,3]. The external/data level approaches use pre-processing techniques at the data level that reform the dataset to a balanced set [4,5]. Cost sensitive learning solutions are developed further with the combination of internal and external level approaches which imposes higher misclassification charges for the samples in the rare class and try to minimize the high cost errors [6,7]. Ensemble methods are also proposed to handle imbalance problem by modifying the ensemble learning algorithm at the data level approach [8] or by embedding cost-sensitive framework in the learning process [9,10].

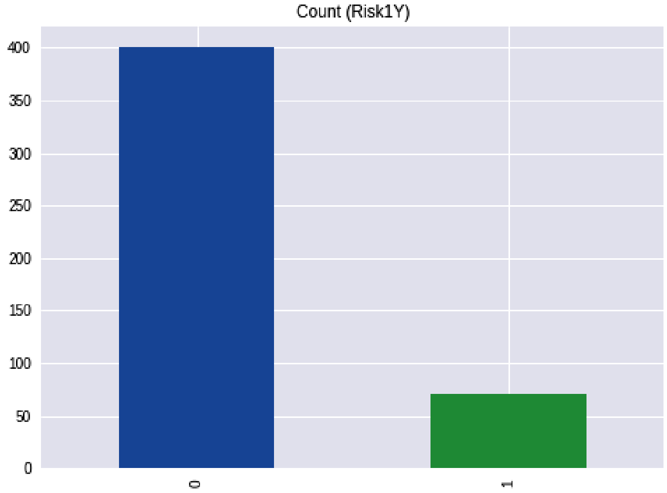


**Fig 1:** Distribution of Class 1 and Class 0 (ratio 1:6) in Thoracic cancer dataset

In recent years, the imbalanced learning problem has received much attention from machine learning community. In this paper we aim to provide the use of resampling techniques to alleviate the problem caused by imbalanced class distributions. The performance of classifiers on imbalanced datasets before and after sampling has been illustrated in detailed manner. The contents of this paper are organized as follows. Section 2 gives a brief note about the datasets. Section 3 describes the basic types of sampling techniques and the six resampling strategies that are adopted for this study. Section 4 presents the four classifiers which are used for classification, followed by section 5 with the discussion of evaluation measures used for imbalanced data classification. In section 6, details of the methodology and results of the study are presented and finally, section 7 presents the conclusion of the paper.

## II. DATASET DESCRIPTION

In this paper, the experiment has been done on four imbalanced datasets obtained from UCI and Kaggle. The characteristics of these data sets are given in Table1.

**Table 1:** Dataset description

| Dataset | # Samples | # Attributes | # Minority class samples | # Majority class samples | Imbalance Ratio |
|---|---|---|---|---|---|
| Thoracic | 470 | 17 | 70 | 400 | 5.71 |
| Abalone | 4177 | 9 | 1342 | 2835 | 2.11 |
| Diabetes | 768 | 9 | 286 | 500 | 1.75 |
| Hepatitis | 142 | 20 | 26 | 116 | 4.46 |

Thoracic cancer data provides the details of survival of cancer patients after surgery. It has 17 attributes including the class variable 'Risk1y'. The class value '1' indicates the death of patients within one year after surgery. There are 70 instances of class1, referred to as minority class and 400 majority class samples. Hence their imbalance ratio is 5.71. The second dataset 'Abalone' provides the classification of infant abalone (class va1ue 1) and matured abalones (class value 0). It has a total of 4177 samples among which the minority count is 1342 and majority count is 2835, finds an imbalance ratio of 2.11. Diabetes data contains the cases of 286 diabetes patients and 500 non-diabetes people. Its imbalance ratio is 1.75. The Hepatitis data has a total of 142 samples with 26 minority class samples and 116 majority class samples. The imbalance ratio of this dataset is 4.46. All datasets except Diabetes were subjected to pre-processing to convert all string valued attributes to numerical values prior to apply sampling techniques. Fig. 2 shows a data representation of samples of Thoracic cancer set.

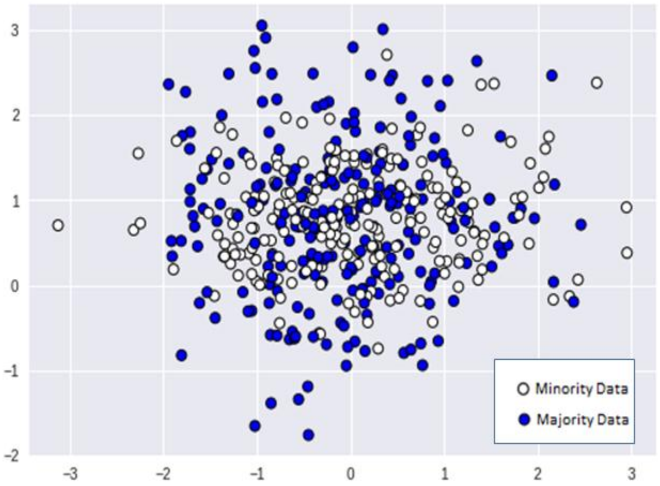


**Fig 2:** Original data distribution of training set of Thoracic cancer set

## III. RESAMPLING TECHNIQUES

Sampling technique is a useful pre-processing task [11] used to balance imbalanced data and it is done by either adding samples to minority class or dropping samples from majority class. The former process is known as oversampling and the latter is called undersampling.  Fig. 3 gives its schematic representation.

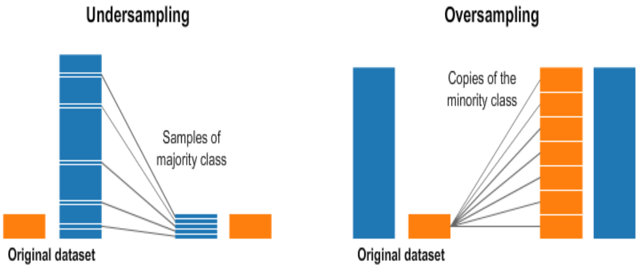


**Fig 3:** Undersampling and Oversampling process [12]

We have included three oversampling and three undersampling methods for the empirical study. The selected oversampling techniques are ROS(Random Over Sampling), SMOTE (Synthetic Minority Oversampling TEchnique) and ADASYN (ADAptive SYNthetic sampling). RUS(Random Under Sampling), CUS(Cluster based Under Sampling) and NearMiss are the other three under sampling techniques chosen for the study.

### 1. *Random Over Sampling*

In the groups of oversampling methods, the simplest preprocessing technique is random oversampling. Samples of the minority class are randomly selected and replicated, so that the count of rare class becomes equal to the count of majority class. ROS has applied to training sets of the four datasets and data distribution of Thoracic cancer set after ROS has shown in Fig. 4.

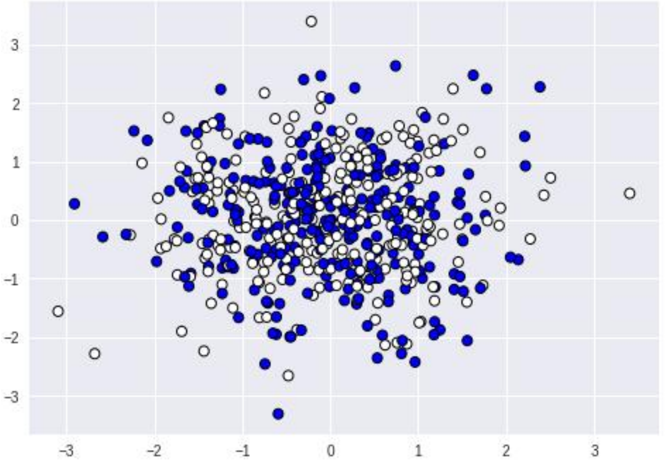


**Fig 4:** Data representation after ROS

In ROS, replication of same data can increase the likelihood of occurring overfitting. Usage of synthetic samples of minority class became popular due to this issue of random over sampling.  SMOTE and ADASYN are the two such synthetic sample generation oversampling methods.

### 2. *SMOTE*

SMOTE is a popular oversampling technique was proposed by Chawla et.al [13] uses synthetic samples instead of duplicating the same data. New samples are generated by taking k-nearest neighbours of minority class samples. The value set for 'k' depends on the number of new samples need to be generated. The difference between the featured vector and neighbouring sample is calculated and then this value is multiplied with a random number between 0 and 1. This new vector value is added to the selected featured vector to create a new sample of minority class.

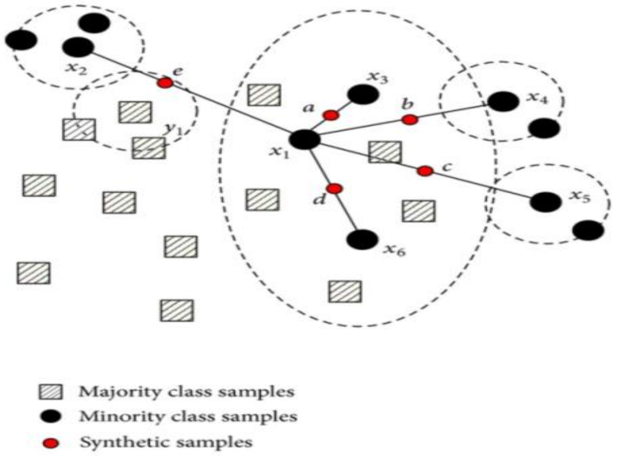
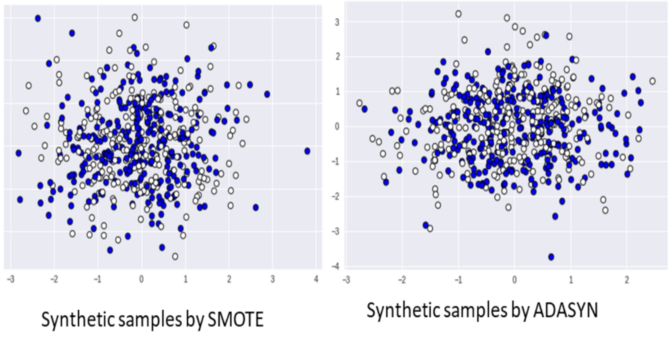


**Fig 5:** Synthetic sample generation using SMOTE for k=5 [14]

*3. ADASYN*

It is an improved version of SMOTE which uses a weighted distribution for different minority class samples depends on the toughness to learn them. More synthetic samples are generated for complicated minority samples than the minority samples which are less complex to learn. This adopted strategy reduces the bias due to the class imbalance [15]. Fig. 6 shows the new data distribution for thoracic cancer set when it is over sampled by SMOTE and ADASYN.



Synthetic samples by SMOTE

Synthetic samples by ADASYN

**Fig 6:** Balanced data after the application of SMOTE & ADASYN

*4. RUS*

RUS is an undersampling method randomly chooses instances from majority class and the selected observations are eliminated from the original set until it gets balanced. The training sets of the four data sets were undersampled using RUS to create a balanced datasets. Fig. 7 shows the status of positive and negative samples of Thoracic cancer set after applying RUS.

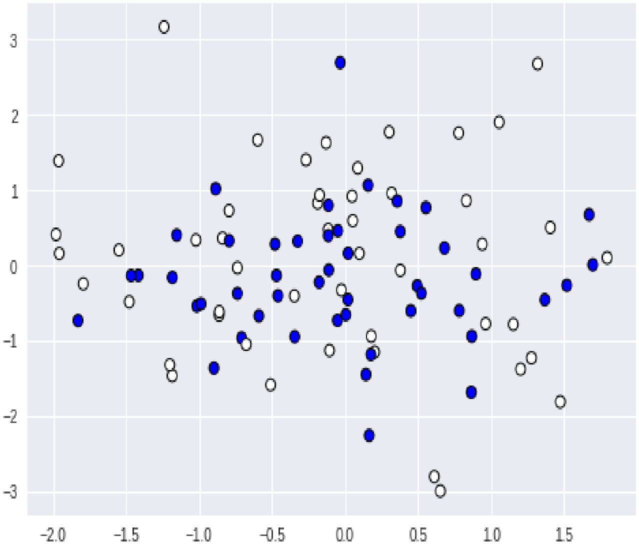


**Fig 7:** Data distribution after RUS

However RUS is very simple, its major drawback is that it can discard potentially useful data that could be important for learning process.

*5. CUS*

In order to overcome the drawback of RUS, Lin et.al [16] proposed a clustering based undersampling method. In this case, simple K-means algorithm is used to cluster the majority class. Each cluster is represented by a cluster centroid and this centroid will be added to the new sample list of majority class. In this way, the count of majority class instances will get reduced and the dataset gets balanced.

*6. NearMiss*

This method selects the majority class samples which are nearer to the minority class samples. i.e. it retains majority class samples whose distances are short to the minority class samples [17].

## IV. CLASSIFIERS SELECTED

Four different classifiers are used for performing classification. They are Support Vector Machine (SVM), AdaBoost (Adaptive Boosting), Random Forest (RF) and Multi-Layer Perceptron (MLP).

*1. SVM*

SVM is a very popular classifier for two-class problems, creates a hyper plane between two sets of data samples. The hyper plane that has the highest distance to the nearest data point is considered as the best hyper plane. Variant SVMs have also been developed for handling imbalanced datasets, GSVM-RU, was proposed by Tang et.al [18] shows high efficiency.

*2. AdaBoost*

AdaBoost is an ensemble method developed for binary classification. This creates a strong classifier from weak classifiers. This is done by creating subsequent models on correcting errors occurred in preceding models. The process is repeated until the training set is predicted perfectly or maximum number of models is added. Jie Song et.al proposed an improved AdaBoost algorithm, Balanced AdaBoost (BABoost) gives high misclassification weights for the samples from minority class [19].

*3.Random Forest*

RF is a powerful ensemble based decision tree classifier, well known for its classification accuracy. It is a combination of tree predictors in which each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large [20]. This error depends on the strength of individual trees in the forest and the correlation between them.

*4. MLP*

MLP is a neural network classifier uses back propagation method to train the network. The weight vector provided at the input layer can be adjusted with the back propagation of error to get a right match of the input and output. Bruzzone et.al proposed a multilayer perceptron to classify the imbalanced remote sensing data, succeeded to get more stable result [21]. Oh et.al modified the error function to update the weight value and thus reduced the skewness in the imbalanced datasets [22].

## V. PERFORMANCE MEASURES

Classification results are verified based on the confusion matrix. Confusion matrix provides the count of both correctly classified and misclassified samples. These values are TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). Python's sklearn.metrics.confusion_matrix is used to generate the values. The conventional method to evaluate the performance is accuracy. But in the case of imbalanced datasets, accuracy does not exhibit the actual performance on minority samples [23]. Hence other metrics such as Precision, Recall, F-Score, G-Mean and ROC are considered.

Precision tells us the rate of correctly classified samples over the predicted ones. In this study, precision over majority sample (-ve) is considered. The values can be derived through the equation, Precision $_{(\text{-ve instances})}$, $P_N = \dfrac{TN}{TN + FN}$

Nevertheless, imbalanced data classification gives more priority on the identification of positive instances, True Positive Rate (Recall (+ve) / Sensitivity) is also included in the study. Its values are obtained by Recall $_{(\text{+ve instances})}$, $R_P = \dfrac{TP}{TP + FN}$

Other metrics, F-Score reflects the goodness of classifier by considering the precision and recall together. This is one of the good methods to assess the performance of classifier. F-S core can be find out using the formula

$$\text{F-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Geometric mean (G-Mean) considers both true positive rate and true negative rate and hence it correlates both objectives. It can be defined as $\text{G-Mean} = \sqrt{\dfrac{TP}{TP + FN} * \dfrac{TN}{TN + FP}}$

Receiver Operating Characteristic (ROC) is a standard technique for exhibiting the trade-off between true positive rate and false positive rate. The Area Under the ROC curve provides a single measure of classifier's performance for evaluating which model is better on average. The AUC measure is computed by $AUC = \dfrac{1+ TP\%+FP\%}{2}$

## VI. EXPERIMENTAL STUDY AND RESULTS

The empirical study has been done with Python's Jupyter Notebook. The following methodology has adopted to conduct the prediction of class labels and for evaluating the effect of sampling in imbalanced datasets.

i) Selection of imbalanced datasets.
ii) Pre-processed the datasets to convert string valued attributes to integer type, for which we use Panda's data frame facility.
iii) Splitted the dataset into Training set and Test set in the ratio 70:30.
iv) Applied the resampling techniques to training set.
v) Learnt the model and fitted the data using four well known classifiers- SVM, AdaBoost, Random Forest and MLP.
vi) Evaluated the performances of classifiers before sampling and after sampling.

Schematic representation of this methodology has shown in Fig. 8.

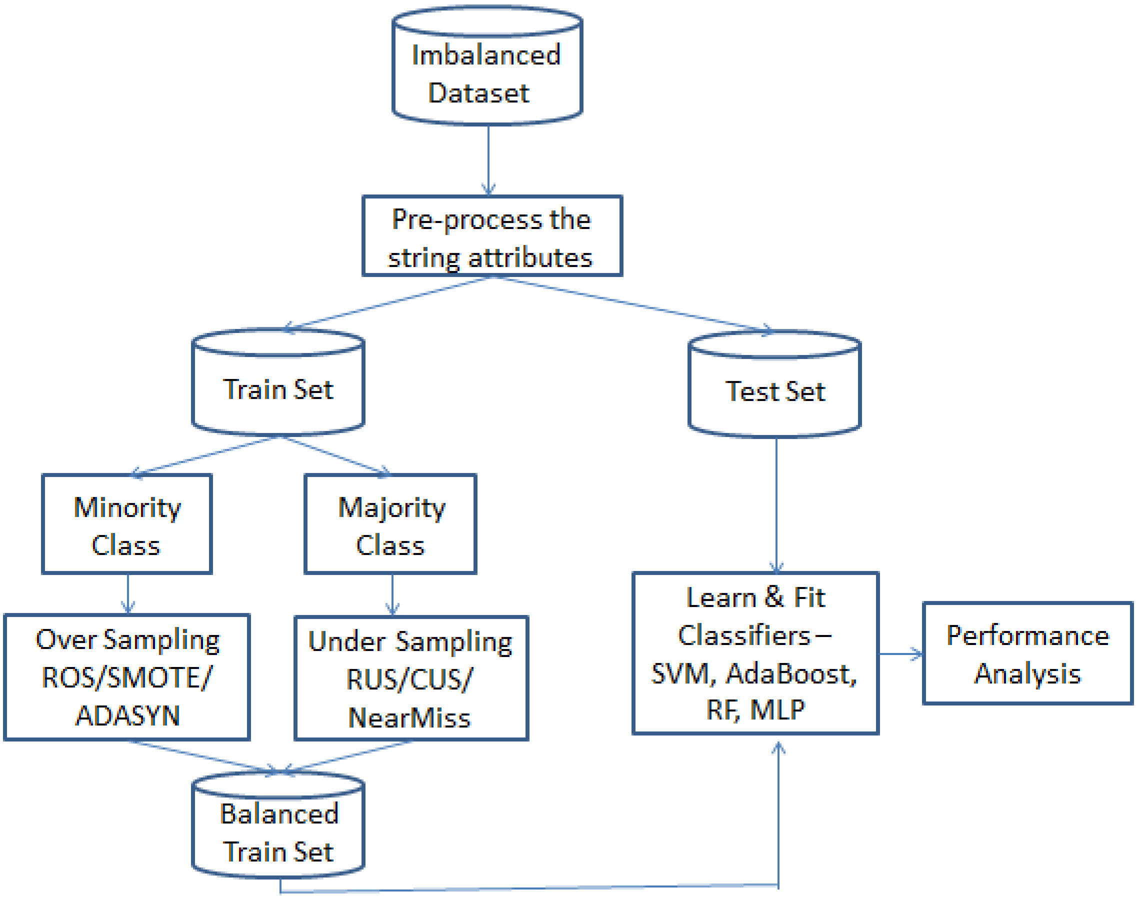

Imbalanced Dataset
Pre-process the string attributes
Train Set
Test Set
Minority Class
Majority Class
Over Sampling ROS/SMOTE/ ADASYN
Under Sampling RUS/CUS/ NearMiss
Balanced Train Set
Learn & Fit Classifiers – SVM, AdaBoost, RF, MLP
Performance Analysis

**Fig 8:** Methodology used

Initially, when the classifiers learnt with unbalanced data, the recall value on +ve class (sensitivity) found very low, sometimes it found as 0. Thereafter the training set was modified by the selected oversampling and undersampling strategies. Here we consider precision on –ve value and recall on +ve value with the intension of getting higher priority on the prediction of +ve(minority) samples. Since it is also necessary to ensure correctness in the classification of –ve (majority) samples, their precision is considered. However missing a +ve sample incurs high loss.

Table2 shows the details of training sets after resampling. We got better sensitivity after applying the resampling techniques. At the same time, precision over majority sample is also preserved. Similar improvement also found with F-Score, GMean and ROC values. These observations are tabulated in Table3 thru Table6. The results were obtained using Python libraries scikit-learn and imbalanced-learn.

**Table 2:** Data distribution of training set before and after resampling

| Dataset | Unbalanced Train set | | After Oversampling | | | | After Undersampling | |
|---|---|---|---|---|---|---|---|---|
| | | | ROS/SMOTE | | ADASYN | | RUS/CUS/ NearMiss | |
| | +'ve | -'ve | +'ve | -'ve | +'ve | -'ve | +'ve | -'ve |
| Thoracic | 48 | 281 | 281 | 281 | 298 | 281 | 48 | 48 |
| Abalone | 919 | 2004 | 2004 | 2004 | 2069 | 2004 | 919 | 919 |
| Diabetes | 194 | 343 | 343 | 343 | 343 | 343 | 194 | 194 |
| Hepatitis | 22 | 77 | 77 | 77 | 75 | 77 | 22 | 22 |

**Table3**: Performance evaluation using SVM

| Dataset | Sampling Method | Precision | Recall | F1-Score | G-Mean | ROC |
|---|---|---|---|---|---|---|
| | | -'ve | +' ve | +' ve | | |
| Thoracic | _ | 0.84 | 0 | 0 | 0.00 | 0.5 |
| | ROS | 0.89 | 0.55 | 0.34 | 0.62 | 0.621 |
| | SMOTE | 0.85 | 0.55 | 0.25 | 0.52 | 0.516 |
| | ADASYN | 0.82 | 0.55 | 0.22 | 0.46 | 0.462 |
| | RUS | 0.89 | 0.59 | 0.33 | 0.61 | 0.615 |
| | CUS | 0.86 | 0.55 | 0.26 | 0.52 | 0.525 |
| | NearMiss | 0.86 | 0.5 | 0.27 | 0.54 | 0.539 |
| Abalone | _ | 0.85 | 0.7 | 0.72 | 0.78 | 0.787 |
| | ROS | 0.91 | 0.86 | 0.73 | 0.80 | 0.803 |
| | SMOTE | 0.91 | 0.86 | 0.73 | 0.81 | 0.807 |
| | ADASYN | 0.93 | 0.9 | 0.71 | 0.79 | 0.791 |
| | RUS | 0.91 | 0.86 | 0.74 | 0.81 | 0.811 |
| | CUS | 0.92 | 0.88 | 0.72 | 0.79 | 0.793 |
| | NearMiss | 0.86 | 0.73 | 0.72 | 0.79 | 0.792 |
| Diabetes | _ | 0.81 | 0.54 | 0.62 | 0.70 | 0.72 |
| | ROS | 0.83 | 0.65 | 0.62 | 0.71 | 0.72 |
| | SMOTE | 0.91 | 0.86 | 0.73 | 0.80 | 0.803 |
| | ADASYN | 0.93 | 0.9 | 0.71 | 0.79 | 0.791 |
| | RUS | 0.91 | 0.86 | 0.74 | 0.81 | 0.811 |
| | CUS | 0.92 | 0.88 | 0.72 | 0.79 | 0.793 |
| | NearMiss | 0.8 | 0.74 | 0.56 | 0.63 | 0.739 |
| Hepatitis | _ | 0.97 | 0.75 | 0.6 | 0.83 | 0.837 |
| | ROS | 0.91 | 0.86 | 0.73 | 0.80 | 0.803 |
| | SMOTE | 0.97 | 0.75 | 0.46 | 0.80 | 0.798 |
| | ADASYN | 0.97 | 0.75 | 0.46 | 0.80 | 0.798 |
| | RUS | 0.96 | 0.75 | 0.32 | 0.72 | 0.721 |
| | CUS | 1 | 1 | 0.35 | 0.79 | 0.808 |
| | NearMiss | 0.97 | 0.75 | 0.33 | 0.73 | 0.73 |

Table 3 displays the results of SVM classifier on the four datasets- Thoracic, Abalone, Diabetes and Hepatitis. The recall and F-score values of Thoracic data were 0 before sampling. Same result found with RF and MLP also. It means that, the classifier was unable to identify a single minority sample due to the data skewness problem. Later, as a result of sampling, more than around 50% increase found with recall value. Values of other metrics like F-Score, G-Mean and ROC, are also showed improvements in their value. In the case of Hepatitis data, CUS gives 1 for both precision and recall which indicates that no positive data misclassified as negative. Similar sort of analysis can be performed in the tables 4, 5 and 6.

**Table 4:** Performance evaluation using AdaBoost

| Dataset | Sampling Method | Precision | Recall | F1-Score | G-Mean | ROC |
|---|---|---|---|---|---|---|
| | | -'ve | +' ve | +' ve | | |
| Thoracic | _ | 0.84 | 0.05 | 0.07 | 0.22 | 0.498 |
| | ROS | 0.87 | 0.36 | 0.27 | 0.52 | 0.56 |
| | SMOTE | 0.87 | 0.27 | 0.29 | 0.49 | 0.578 |
| | ADASYN | 0.87 | 0.27 | 0.29 | 0.49 | 0.578 |
| | RUS | 0.88 | 0.55 | 0.3 | 0.58 | 0.584 |
| | CUS | 0.88 | 0.91 | 0.27 | 0.34 | 0.518 |
| | NearMiss | 0.87 | 0.59 | 0.27 | 0.54 | 0.606 |
| Abalone | _ | 0.86 | 0.72 | 0.73 | 0.79 | 0.794 |
| | ROS | 0.9 | 0.83 | 0.71 | 0.78 | 0.786 |
| | SMOTE | 0.9 | 0.83 | 0.72 | 0.79 | 0.795 |
| | ADASYN | 0.93 | 0.91 | 0.71 | 0.77 | 0.784 |
| | RUS | 0.91 | 0.86 | 0.73 | 0.80 | 0.799 |
| | CUS | 0.92 | 0.88 | 0.72 | 0.79 | 0.794 |
| | NearMiss | 0.8 | 0.74 | 0.56 | 0.63 | 0.739 |
| Diabetes | _ | 0.79 | 0.53 | 0.58 | 0.68 | 0.69 |
| | ROS | 0.84 | 0.65 | 0.66 | 0.74 | 0.75 |
| | SMOTE | 0.84 | 0.66 | 0.65 | 0.74 | 0.75 |
| | ADASYN | 0.83 | 0.65 | 0.63 | 0.72 | 0.73 |
| | RUS | 0.84 | 0.7 | 0.61 | 0.71 | 0.71 |
| | CUS | 0.82 | 0.62 | 0.63 | 0.72 | 0.72 |
| | NearMiss | 0.81 | 0.65 | 0.57 | 0.68 | 0.68 |
| Hepatitis | _ | 0.94 | 0.5 | 0.33 | 0.65 | 0.673 |
| | ROS | 0.94 | 0.5 | 0.29 | 0.63 | 0.647 |
| | SMOTE | 0.97 | 0.75 | 0.5 | 0.81 | 0.811 |
| | ADASYN | 0.97 | 0.75 | 0.43 | 0.78 | 0.785 |
| | RUS | 0.97 | 0.75 | 0.4 | 0.77 | 0.772 |
| | CUS | 0.95 | 0.75 | 0.23 | 0.62 | 0.631 |
| | NearMiss | 1 | 1 | 0.31 | 0.73 | 0.897 |

**Table 5:** Performance evaluation using RF

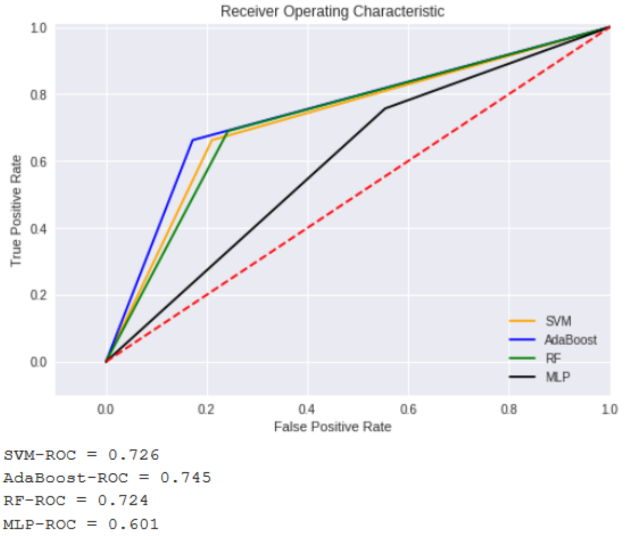
| Dataset | Sampling Method | Precision | Recall | F1-Score | G-Mean | ROC |
|---|---|---|---|---|---|---|
| | | -'ve | +' ve | +' ve | | |
| Thoracic | _ | 0.84 | 0 | 0 | 0.00 | 0.5 |
| | ROS | 0.86 | 0.5 | 0.26 | 0.53 | 0.536 |
| | SMOTE | 0.86 | 0.5 | 0.26 | 0.53 | 0.532 |
| | ADASYN | 0.87 | 0.5 | 0.28 | 0.55 | 0.557 |
| | RUS | 0.89 | 0.55 | 0.33 | 0.61 | 0.609 |
| | CUS | 0.87 | 0.73 | 0.27 | 0.49 | 0.528 |
| | NearMiss | 0.83 | 0.45 | 0.22 | 0.47 | 0.475 |
| Abalone | _ | 0.84 | 0.66 | 0.69 | 0.76 | 0.768 |
| | ROS | 0.9 | 0.84 | 0.71 | 0.78 | 0.785 |
| | SMOTE | 0.9 | 0.85 | 0.71 | 0.78 | 0.786 |
| | ADASYN | 0.92 | 0.89 | 0.69 | 0.75 | 0.766 |
| | RUS | 0.9 | 0.84 | 0.71 | 0.78 | 0.788 |
| | CUS | 0.92 | 0.88 | 0.71 | 0.78 | 0.785 |

| Dataset | Sampling Method | Precision | Recall | F1-Score | G-Mean | ROC |
|---|---|---|---|---|---|---|
| | | -'ve | +' ve | +' ve | | |
| | NearMiss | 0.83 | 0.65 | 0.68 | 0.75 | 0.759 |
| Diabetis | _ | 0.74 | 0.31 | 0.43 | 0.54 | 0.62 |
| | ROS | 0.82 | 0.66 | 0.6 | 0.70 | 0.7 |
| | SMOTE | 0.84 | 0.69 | 0.63 | 0.72 | 0.72 |
| | ADASYN | 0.84 | 0.7 | 0.62 | 0.72 | 0.72 |
| | RUS | 0.84 | 0.69 | 0.62 | 0.72 | 0.72 |
| | CUS | 0.82 | 0.65 | 0.6 | 0.70 | 0.71 |
| | NearMiss | 0.81 | 0.61 | 0.59 | 0.69 | 0.7 |
| Hepatitis | _ | 0.95 | 0.5 | 0.67 | 0.71 | 0.75 |
| | ROS | 0.97 | 0.75 | 0.46 | 0.80 | 0.798 |
| | SMOTE | 0.97 | 0.75 | 0.5 | 0.81 | 0.811 |
| | ADASYN | 0.97 | 0.75 | 0.55 | 0.82 | 0.824 |
| | RUS | 0.97 | 0.75 | 0.35 | 0.74 | 0.747 |
| | CUS | 0.96 | 0.75 | 0.26 | 0.67 | 0.669 |
| | NearMiss | 0.96 | 0.75 | 0.32 | 0.72 | 0.721 |

**Table 6:** Performance evaluation using MLP

| Dataset | Sampling Method | Precision | Recall | F1-Score | G-Mean | ROC |
|---|---|---|---|---|---|---|
| | | -'ve | +' ve | +' ve | | |
| Thoracic | _ | 0.84 | 0 | 0 | 0.00 | 0.498 |
| | ROS | 0.91 | 0.68 | 0.37 | 0.65 | 0.652 |
| | SMOTE | 0.88 | 0.73 | 0.28 | 0.51 | 0.54 |
| | ADASYN | 0.85 | 0.73 | 0.26 | 0.45 | 0.502 |
| | RUS | 0.88 | 0.64 | 0.29 | 0.57 | 0.57 |
| | CUS | 0.84 | 0.77 | 0.26 | 0.42 | 0.499 |
| | NearMiss | 0.85 | 0.68 | 0.26 | 0.47 | 0.505 |
| Abalone | _ | 0.86 | 0.72 | 0.74 | 0.80 | 0.803 |
| | ROS | 0.91 | 0.85 | 0.73 | 0.80 | 0.803 |
| | SMOTE | 0.91 | 0.84 | 0.74 | 0.80 | 0.807 |
| | ADASYN | 0.93 | 0.89 | 0.71 | 0.78 | 0.79 |
| | RUS | 0.9 | 0.84 | 0.73 | 0.80 | 0.803 |
| | CUS | 0.92 | 0.89 | 0.72 | 0.79 | 0.793 |
| | NearMiss | 0.85 | 0.76 | 0.65 | 0.73 | 0.731 |
| Diabetis | _ | 0.68 | 0.15 | 0.2 | 0.35 | 0.49 |
| | ROS | 0.79 | 0.69 | 0.52 | 0.61 | 0.61 |
| | SMOTE | 0.8 | 0.76 | 0.52 | 0.58 | 0.6 |
| | ADASYN | 0.81 | 0.77 | 0.53 | 0.59 | 0.61 |
| | RUS | 0.73 | 0.36 | 0.42 | 0.54 | 0.59 |
| | CUS | 0.72 | 0.61 | 0.45 | 0.54 | 0.54 |
| | NearMiss | 0.7 | 0.32 | 0.35 | 0.49 | 0.53 |
| Hepatitis | _ | 0.93 | 0.25 | 0.4 | 0.50 | 0.625 |
| | ROS | 1 | 1 | 0.25 | 0.62 | 0.692 |
| | SMOTE | 1 | 1 | 0.23 | 0.56 | 0.654 |
| | ADASYN | 0.92 | 0.75 | 0.17 | 0.46 | 0.516 |
| | RUS | 0.96 | 0.75 | 0.3 | 0.71 | 0.708 |
| | CUS | 0.89 | 0.25 | 0.11 | 0.40 | 0.446 |
| | NearMiss | 0.96 | 0.75 | 0.32 | 0.72 | 0.721 |

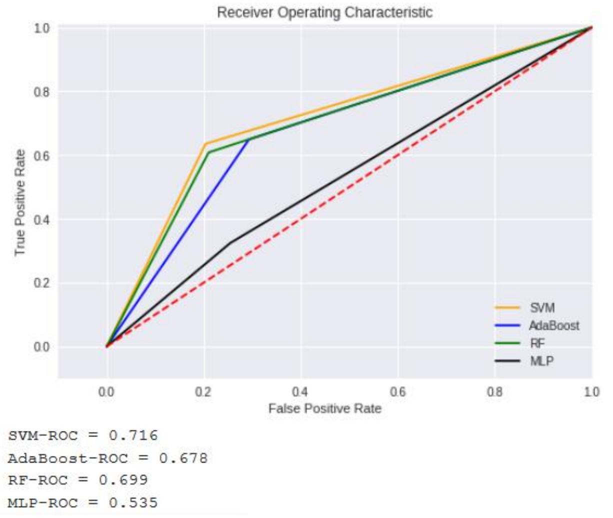
While analyzing the results in tables 3 to 6, SMOTE outperforms over the other sampling techniques. In some cases, ROS and RUS also lead in prediction. But when continue the experiment, we identified that the training dataset obtained through RUS and ROS for the same data changes for each invocation and hence produces different result for same dataset, which is sometimes good and sometimes bad. Hence use of ROS and RUS are not advisable. The goodness of ADASYN is also remarkable. Fig.9 and Fig.10 shows the performance of classifiers after applying the oversampling strategy SMOTE and undersampling strategy NearMiss. We observed that the ROC and G-Mean values obtained through oversampling are fairly good than the values obtained through undersamling.



```
SVM-ROC = 0.726
AdaBoost-ROC = 0.745
RF-ROC = 0.724
MLP-ROC = 0.601
```

**Fig9:** ROC comparison of classifiers in Diabetes dataset after SMOTE

Among the sampling techniques chosen, SMOTE performs well irresptive of nature of data and imbalance ratio. Through the experimental analysis, oversampling with synthetic sample generation is found to be a reliable and adoptable resampling strategy.



```
SVM-ROC = 0.716
AdaBoost-ROC = 0.678
RF-ROC = 0.699
MLP-ROC = 0.535
```

**Fig10:** ROC comparison of classifiers in Diabetes dataset after NearMiss

In this study, we give prime importance on how resampling techniques help the classifiers to improve their prediction on minority but relevant data. Performance comparison among the selected classifiers is secondary. Table 7 shows the average performance of each classifier based on G-Mean value. SVM performs best for each dataset with highest G-Mean and ROC values. AdaBoost and RF are also having remarkable performance in the classification process, placed just behind SVM. In each dataset except Abalone, MLP shows the lowest performance.

**Table7:** Average performance of classifiers

| Classifier | Dataset | | | |
|---|---|---|---|---|
| | Thoracic | Abalone | Diabetes | Hepatisis |
| SVM | 0.4677 | 0.7945 | 0.7466 | 0.781 |
| AdaBoost | 0.4552 | 0.7656 | 0.7123 | 0.7136 |
| RF | 0.45535 | 0.7703 | 0.6856 | 0.7521 |
| MLP | 0.43812 | 0.7886 | 0.5293 | 0.5657 |

## VII. CONCLUSION

Preprocessing of imbalanced datasets is essential as many real life problems are get affected by the skewness of certain class values. When compared the methods available for imbalanced problems, resampling is found more simple and flexible. By this study, we identified that oversampling method with synthetic sample generation shows best results for the entire datasets. Nevertheless undersampling is an alternate choice for rebalancing, oversampling is advisable to use, because undersampling often leads to the loss of vital information. This study shows that there is substantial improvement over the performance of classifiers after when the datasets were subjected to resampling process. The positive change in performance is more obvious with the data whose imbalance ratio is very high.

## REFERENCES

[1] Guo Haixian, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, Gong Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems With Applications 73 (2017), pp. 220–239.

[2] R. Barandela, J.S. Sanchez, V. Garcia, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (3) (2003) 849–851.

[3] W. Lin, J.J. Chen, Class-imbalanced classifiers for high-dimensional Briefings in Bioinformatics 14 (1) (2013) 13–26.

[4] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, SIGKDDExplorations 6 (1) (2004) 20–29.

[5] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced

data sets, Computational Intelligence 20(1) (2004) 18–36.

[6] R. Batuwita, V. Palade, Class imbalance learning methods for support vector machines, in: H. He, Y. Ma (Eds.), Imbalanced Learning: Foundations, Algorithms, and Applications, Wiley, 2013, pp. 83–96.

[7] P. Domingos, Metacost: a general method for making classifiers cost–sensitive, in: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99), 1999, pp. 155–164.

[8] Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (1) (2010) 1–39

[9] W. Fan, S.J. Stolfo, J. Zhang, P.K. Chan, Adacost: misclassification cost-sensitive boosting, in: Proceedings of the 16th International Conference on Machine Learning (ICML'96), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 97–105.

[10] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (12) (2007) 3358–3378.

[11] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, SIGKDD Explorations 6 (1) (2004) 20–29.

[12] Available at https://www.kaggle.com/rafjaa/resampling- strategies- for-imbalanced-datasets.

[13] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligent Research 16 (2002) 321–357.

[14] Available at https://medium.com/coinmonks/smote-and-adasyn- handling imbalanced-data-set-34f5223e167.

[15] Haibo He ; Yang Bai ; Edwardo A. Garcia ; Shutao Li, Adaptive Synthetic sampling approach for imbalanced learning, 2008, IEEE International Joint Conference on Neural Networks (IEEE world Congress on Computational Intelligence).

[16] Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S.: Clustering-based undersampling in class-imbalanced data. Information Sciences 409- 410 (Supplement C), 17 { 26 (2017).

[17] Inderjeet Mani and I Zhang. KNN approach to unbalanced data distributions: a case study involving information extraction. In Proceedings of workshop on learning from imbalanced datasets, 2003.

[18] Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S.: Modeling for highly imbalanced classification, J. latex class files. **1**(11) (2002).

[19] Jie Song, Xiaoling Lu, Xizhi Wu, An Improved AdaBoost Algorithm for Unbalanced Classification Data, 2009 Sixth Discovery **,** Volume7.

[20] Leo Breiman, Machine Learning, 45, 5–32, 2001c ©2001 Kluwer Academic Publishers.

[21] L.Bruzzone & S.B.Serpico Classification of imbalanced remote- sensing data by neural networks, Pattern Recognition Letters Volume 18, Issues 11–13, November 1997, Pages 1323-1328.

[22] Sang- Hoon Oh, Error back-propagation algorithm for classification of imbalanced data, Neurocomputing, Volume 74, Issue 6, February 2011, Pages 1058-1061.

[23] Shidha M.V, Mahalsekshmi T, Assessing the impact of Instance Imbalance in the prediction of Class labels, in: Proceedings of the National Conference on Advanced Computing (NCAC'19) in Bharathiar University, January 2019.