The Impact of Data Re-Sampling on Learning Performance of Class Imbalanced Bankruptcy Prediction Models

Dilip Singh Sisodia and Upasana Verma

National Institute of Technology Raipur, Raipur, India,
dssisodia.cs@nitrr.ac.in

Abstract: The aim of this paper is to evaluate the effect of data sampling techniques on the performance of learners using real highly imbalanced Spanish bankruptcy dataset. The class imbalance problem refers to the highly uneven distribution of class instances where one class is having most of the instances than others. In the presence of highly skewed data distribution, the performance of classical learners is heavily biased in recognizing the majority class and consequently leads to the performance degradation of quantitative classifier or predictors models. In this paper, six sampling methods such as synthetic minority oversampling technique (SMOTE), Borderline-SMOTE, Safe-level-SMOTE, Random under sampling, random oversampling and condensed nearest neighbor are used with a different individual(SVM, C4.5, and Logistic regression) and ensemble learners(AdaBoostM1, DTBagging, and Random Forests). The different quantitative prediction models are designed by combination data sampling techniques and classical learners. The performance of quantitative prediction models are evaluated using G-Mean and area under the curve (AUC) measures on the real highly imbalanced data set. The result suggest that the performance of oversampling (with LR and DTBagging) and undersampling (with C4.5 and RF) methods are superior as compare to others on this data set.

Keywords: Class imbalance, Ensemble learners, Individual learners, Prediction, Sampling, Bankruptcy Prediction Model, and Performance Evaluation.

1. Introduction

Applications of machine learning (ML) techniques are growing very fast for solving diverse real world problems. Prominently ML techniques are very successful in classification and prediction problems where learning based on past data records and most of the classification algorithms are trained under the assumption of even distribution of classes' instances. The quality of solution heavily depends on underlying distribution of data points in problem space[1]. However, many real-world problems including oil spilling [2], network intrusion detection [3], weld flaw [4], financial fraud detection [5], churn prediction [6]and bankruptcy prediction [7] Are suffered from skewed distribution of class instances. Such problems are attributed to class imbalance problems where most of the example belongs to one class, and few of them belong to another class[8]. Therefore, the learning of standard classifier is biased for majority class by completely ignoring the minority class. The major challenge is to achieve the high accuracy in correctly classifying the minority class examples without any impact on majority class[9].

There are two fundamental reasons for performance are degradation of learners on class imbalance problems [10]. One is related to the objective function of the classification algorithm is based on arithmetic mean accuracy. The arithmetic means accuracy is the ratio of accurately classify instances to the total number of instances. However, in a case of class imbalance problem arithmetic accuracy is fully dependent on the majority class and abnormally very high due to

Received: June 10th, 2017. Accepted: July 23rd, 2018
DOI: 10.15676/ijeei.2018.10.3.2

skewed data distribution and learner always predicts the majority class. Furthermore, in [10] they focused on the accuracy which is based on geometric mean as it considers both majorities as well as the minority classes. The second reason is due to the distortion in resulting boundaries. The majority class decision limits prevail the minority class decision limit and significantly reduce the accuracy of a minority class. To address the issue of class imbalance different approaches are used, and Figure 1 shows the summary of these approaches [11].

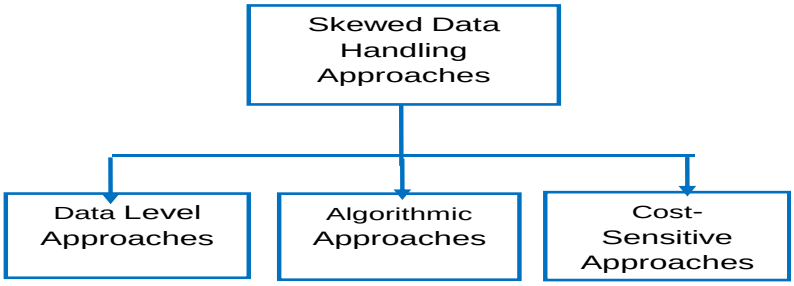

Figure 1. Classification of skewed data handling approaches

The Data-level approach involves some modification in the data before taking it for further processing. Some well-known methods are SMOTE [12], ADASYN[13], Random Undersampling, Random oversampling. In this paper, we are using data-level approach. In algorithmic approach, the ensemble methods are used to solve class imbalance are AdaBoost [14], Bagging [15], RUSBoost [16]. In cost-sensitive approach, data level, as well as an algorithmic approach or both, can be used individually.

The remaining work is organized as follows: Section II presents the related work done previously. Section III describes the methodology used in the experiment. The dataset used for this research, evaluation parameter and their results are discussed in section IV and lastly the conclusion, and future work is summarized under section V.

2. Related Work

This section discusses the work reported in the literature for addressing the issues of learning of class imbalance data. The data sampling technique is more popular in handling the imbalance data distribution, because of versatile nature and independence from the underlying classifiers. Synthetic minority oversampling technique (SMOTE), proposed to increase the minority class instances by creating the synthetic data [12] and performance is evaluated using C4.5, Ripper and Naïve Bayes classifiers on a different dataset from UCI machine learning repository. Another popular sampling technique is adaptively synthetic sampling (ADASYN) suggested by [13]. ADASYN was applied on five data sets with different evolution parameter. The undersampling methods including random undersampling (RUS), removes the desired number of randomly selected majority class instances from the skewed data set to obtain the balanced result randomly [17]. The data sampling techniques are used in different application such as weld flaw detection, where 22 different data preprocessing technique were used in six different categories of weld flaws[4]. The bankruptcy prediction, of corporate firms, was discussed in details by Altman [18], Beaver [19], and Ohlson [20]. The main focus of these studies are ratio analysis and to give an empirical verification of the financial data. Their technique is very simple based on healthy and non-healthy companies. In[18] they used multivariate discriminate analysis (MDA) for classification of bankrupt or not bankrupt, where Bayes classification has been applied based on some assumption that covariance matrix is same for both bankrupt as well as non-bankrupt. The MDA is the most popular analysis for financial prediction but to some issues, as discussed in [21] related to MDA, the logistic analysis (LA) techniques are also used. There

are other classifiers used for bankruptcy prediction by many researchers. In[22], Neural Network(NN) based model is used with some novel indicator for the prediction. Another problem found in the literature survey is some of the prediction is based on the Genetic Algorithm (GA). In[23] genetic algorithm (GA) based learning model performs efficient financial failure prediction. In [17] seven sampling techniques has been used with five quantitative methods and solve the highly skewed data problem. In [24], Genetic programming approach is used to handle highly imbalanced data using cost modifying approach and fitness functions. Ensemble learners are a combination of multiple individual learners including DT (decision tree), ANN (Artificial neural network), and SVM (Support Vector Machine) and widely used in financial prediction. In [7] an ensemble of MDA, LR(Logistic regression), CRT(classification and regression tree), and ANNs are used to identify the bankruptcy prediction by using the financial indicators of Russian companies. In [24] [14]three AdaBoost models are used for corporate bankruptcy prediction with various imputation methods applied on various data size. In [10] Boosting algorithm are used to solve a data imbalance problem with geometric mean (GMBoost) which considered both majority class as well as the minority class.

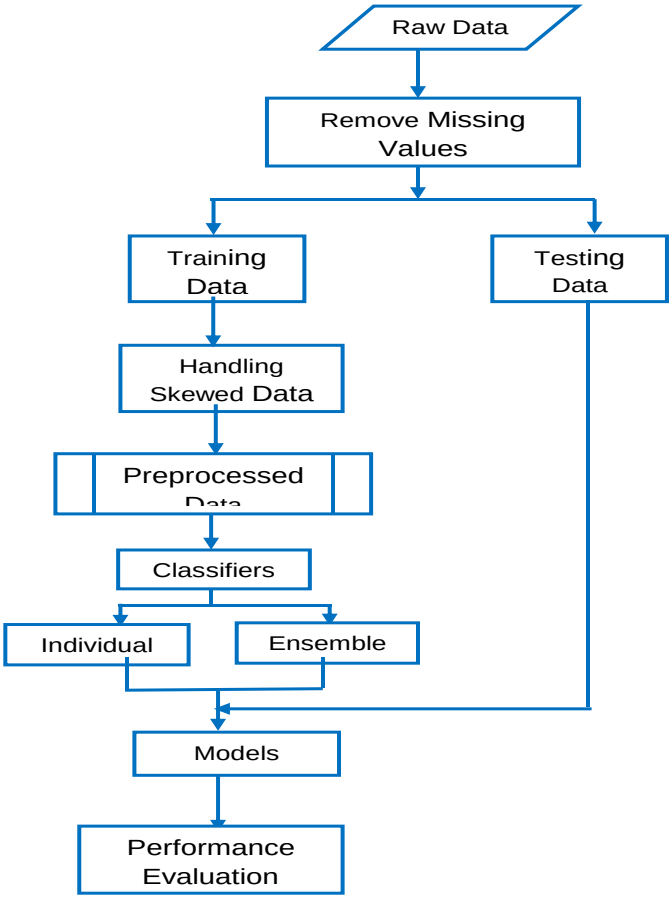

Figure 2. The Process flow of adopted methodology

3. Methodology

In this section, the adopted methodology to carry out the experimental work is described in details. The pictorial summary of the whole process is represented in Figure 2 as a process flow diagram.

A. Raw Dataset description

The Spanish bankruptcy dataset is used in this article for experimentation which collected from GitHub [25]. The original dataset consists of 38 financial and non-financial variables, but by considering only most useful and significant features, it has reduced to 30 independent attributes with categorical and numerical values. The Table1 and 2 briefly describe the various financial and non-financial variable used for bankruptcy. In Table 1, the description of 15 financial variables conveniently named as A1 to A15 is given along with their type of values such as an integer, real or binary. The financial variables are more useful for classification as compare to non-financial ones.

Table 1. Description of financial variable used for bankruptcy prediction

Variable	Financial Variable	Description	Type
A1	Current liability / Debt structure	long term liabilities	Real
A2	Debt cost	Total liabilities	Real
A3	Cash ratio	Current liability	Real
A4	Working capital	Working capital /Total Assets	Real
A5	Debt ratio	Total assets /Total liabilities	Real
A6	Operating income	Net sales	Real
A7	Leverage	Liabilities /equity	Real
A8	Debt-paying ability	operating cash flow /total liabilities	Real
A9	Return on operating assets	operating income / average operating assets	Real
A10	Return on equity	Net income / average total equity	Real
A11	Return on assets	Net income / average total assets	Real
A12	Receivable turnover	Net sales /average receivable	Real
A13	Stock turnover	costs of sales /average inventory	Real
A14	Current ratio	current assets /current liabilities	Real
A15	Acid test	(cash equivalent + marketable securities +net receivable)/current liabilities	Real

In Table 2, all non-financial variables named as A16 to A30 are described. These variables take a different range of values including integer, real and binary. Some features including the size of the company, auditor's opinion, type of company are categorical types. For example, the size of the company is a real type, but the company is categorized into three groups according to their size such as small, medium, large. These features are used to categorize the company into bankrupt or non-bankrupt.

Table 2. Description of non-financial variable used for bankruptcy prediction

Variable	Non-Financial Variable	Description	Type
A16	Size	Small/Medium/Large& Categorical	Categorical
A17	Age of the company		Integer
A18	Audited	If the company has been audited	Binary
A19	Type of company	Public Company/Limited Liability Company(Ltd)/Others	Categorical
A20	Historic amount of money spent on judicial incidences	Since the inception company	Real
A21	Amount of money spent on judicial incidences	Last year	Real

A22	Number of changes of location		Integer
A23	Number of employees		Integer
A24	Historic number of serious incidences	Such as strikes, accidents	Integer
A25	Historic number of judicial incidences	Since the inception of company	Integer
A26	Number of judicial incidences	Last year	Integer
A27	Number of partners		Integer
A28	Auditor's opinion	Favorable/Exceptions/ Unfavorable	Categorical
A29	Delay	If the company has submitted its annual accounts on time	Binary
A30	Linked to group	If the company is part of a group holding	Binary

The raw dataset contains some inconsistencies including missing values as well as the highly skewed distribution of class labels. Therefore, straightforward use of machine learning/prediction techniques may lead to an inaccurate result. Therefore, to address this inconsistency problem, some pre-processing techniques are used as described in following subsections:

B. Remove missing values

The missing values are handled by applying some imputation method which replaces the particular missing value with its mean of the numeric distribution. The replacement of missing value works well if you have a less amount of missing values in your data otherwise some different imputation techniques should be used.

The preprocessed data is divided into two parts one for the testing set and another for the training set using distribution optimally balances stratified-cross-validation (DOB-SCV) technique. DOB-SCV is a better scheme regarding both bias and variance as compared to regular cross validation [26]. In this work fivefold DOB-SCV is used.

C. Remove skewness from input data

The concept of skewness can be understood by assuming the total number of the sample set into majority class is denoted by T_{ma} on the other hand minority class sample set is denoted by T_{mi} from the original training data set. The size of the minority and majority class is denoted by $|T_{mi}|$ and $|T_{ma}|$ respectively and $|T_{mi}|<|T_{ma}|$. We use N_b and B_b notation as unique ID for non-bankrupt and bankrupt respectively. In our dataset, most of the sample belongs to the non-bankrupt class so it is majority class and bankrupt belong to the minority class. So, the set is $T_{ma} = \{N_{b_1}, N_{b_2}, N_{b_3}, N_{b_4}, N_{b_5}, N_{b_6}, N_{b_7}, N_{b_8}\}$ and $|T_{ma}| = 8$ for majority class, the set $T_{mi} = \{B_{b_1}, B_{b_2}\}$ and $|T_{mi}| = 2$ belongs to minority class.

The issue of class imbalance may heavily bias the performance of the classifier for majority class. Data sampling techniques including under sampling, oversampling[17] and hybridization of both are used to address the issue of skewed data.

C.1. Oversampling

It is one of the sampling strategies where the minority class instances are increased to balance the data. Some important oversampling technique used in this work is as follows:

Synthetic minority oversampling technique (SMOTE) [12], is one of the most widely used oversampling technique which oversamples the minority class by generating new synthetic instances. The main parameter of SMOTE is, the percentage by which minority class is oversampled, the number of nearest neighbor and the total number of minority class instances. The percentage of oversampling is chosen according to the imbalance ratio majority and minority class. Borderline-SMOTE(BSMOTE) [27] generates the synthetic data only for borderline minority instances. In this, firstly the borderline minority class are sampled from the training set and then for this extracted data new synthetic instances are generated, at last, data will be added to the original training set. Safe-level-SMOTE (SLS) generate the syntactic data for the minority class those are in the same line and whose weight degree is different and is computed by a minority class nearest neighbor. Random Oversampling (ROS) is a method to balance the minority class using class distribution through random replication.

C.2. Undersampling

In the under-sampling technique, the majority class instances are decreased to overcome the problem of imbalance. There are various undersampling techniques including random undersampling (RU), and condensed nearest neighbor (CNN) are used in this work. In random undersampling (RU) technique majority class examples are removed by random selection of data. In condensed nearest neighbor (CNN), data is reduced from the training set of majority class and find important observation to classify new observation. CNN used two bins S and T, initially the first training sample set is placed in S, and the remaining is placed in sample T. After that, in T first pass is performed and during scanning if a point of T is not classified using the content of S, it is removed from T and placed in S. The algorithm is iterated till there is no point which is placed from T to S in a complete pass of T.

D. Classification techniques

Classification is supervised machine learning technique, where the class labels of the data prior known to us. In this work three individual and three ensemble learners as summarized in Figure 3 are used.

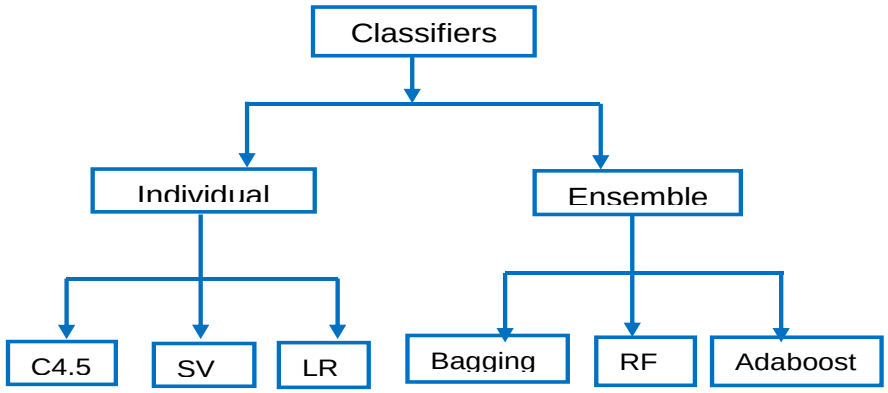

Figure 3. The summary of classifiers used

D.1. Individual learners

The C4.5 is developed by Ross Quinlan [28]and used to generate the decision trees. The C4.5 is one of the statistical classifiers where each node of the tree chooses the attribute of the data that most effectively split its set of sample into subset enriched in one or other class.

Logistic regression (LR) is a binary analysis of independent variables is and calculates the probability of response variable [29]. Support vector machine (SVM) used to classify the input

data into higher dimensions using the different kernel functions and finding the best hyper plane which separates the pattern of one class from another [30].

D.2. Ensemble learners

Ensemble learners are the group of multiple individual classifiers which work cooperatively to improve the generalization ability and increase the prediction performance [31]. Bagging [32] is a Bootstrap Aggregation ensemble method which creates individuals for its ensemble by training each classifier on the random distribution of training set. Each classifier train set is generated by random drawing, with replacement such as many of the original examples may be repeated in the resulting set while other may be left out. AdaBoostM1 is used to improve the simple boosting via an iterative process and creating the data subset for base classifier by resampling the training pattern [33]. Random Forest (RF) [34] is also known as random subspace and uses a large number of the individual unpruned decision trees at training time and outputting the class.

E. Performance evaluation measures

The Performance of binary classification problem is evaluated by generating the confusion matrix[10] for training and testing datasets as shown in Table 3.

Table 3. Confusion matrix

	Positive Predicted	Negative Predicted
Actual Positive	TP(True positive)	FN(False Negative)
Actual Negative	FP(False Positive)	TN(True Negative)

Where True Positive (TP) refers to the number of positive instances which are correctly predicted as positives by a learner. True Negative (TN) denotes the number of negative instances correctly classified as negatives by a learner. False Positive (FP) often referred to as false alarm; defines as the number of negative instances incorrectly classified as positives by a learner. False Negative (FN), sometimes known as Miss; is determined as the number of positive instances incorrectly assigned as negatives by a learner[35].

Traditionally, the accuracy rate has been the most commonly used empirical measure. However, in imbalanced datasets, accuracy is no longer a proper measure, since it does not distinguish between the numbers of correctly classified examples of different classes. Therefore, the other measures including sensitivity, specificity, and geometric mean are used when both classes are important and expected to be high simultaneously[36].

Sensitivity (also known as true positive rate/recall) refers to the ability of a classifier to correctly identifying the positive class as such and shown in Eq.(1). It ranges from 0 to 1 with 1 being the perfect score.

$$\text{Senstivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (1)$$

Specificity (also known as true negative rate) denotes the ability a classifier to correctly identifying the negative class as such shown in Eq.(2). The perfect score is 1 and 0 is the worst measure.

$$\text{Specificity} = \frac{\text{TN}}{\text{FP+TN}} \quad (2)$$

G-mean is a geometric mean of sensitivity (the accuracy of positive instances) and specificity (the accuracy of negative instances)[37] as shown in Eq.(3).

$$\text{G} - \text{mean} = \sqrt{\text{Senstivity} * \text{Specificity}} \quad (3)$$

The area under ROC(AUC), where ROC [38] stands for receiver operating characteristic that will be used to evaluate the performance of a binary classifier. It is a two-dimensional curve plotted between sensitivity in Y-axis and 1-specificity (FPR)in X-axis[39] and computed using Eq.(4).

$$\text{AUC} = \frac{1+\text{TPR}-\text{FPR}}{2} \quad (4)$$

True positive rate (TPR) is the percentage of positive instances correctly classified, and false positive rate (FPR) is the percentage of negative instances misclassified. The ROC curve shows that for any classifier TPR cannot increase without increasing the FPR. The larger the AUC, the better is the classifier performance.

4. Experimental Results And Discussion

In this section, the extensive experiments are performed on individual learners and ensemble learners by applying it on the bankruptcy data. All the experiments are performed on a personal computer having 3.40GHz Core i7-4770 with 4.0 GB memory and running under the Microsoft Windows 8.1 Pro. The implementation of data sampling techniques is used from open source library of Keel software tool [40, 41] while The learners and performance measures are implemented using open source machine learning Weka API [42]. Spanish bankruptcy dataset is a collection of information from 470 companies during six successive years (from 1998 to 2008) and includes financial and non-financial features. In this dataset 2860 number of instances are present, where 2798 instances belong to the non-bankrupt class, and 62 instances are of bankrupt class. The Figure 4 shows that year wise bankrupt and non-bankrupt, also shows that the non-bankrupt data will increase yearly whereas bankrupt data will be decreased, so the total number of bankruptcy is very less as compare to non-bankruptcy.

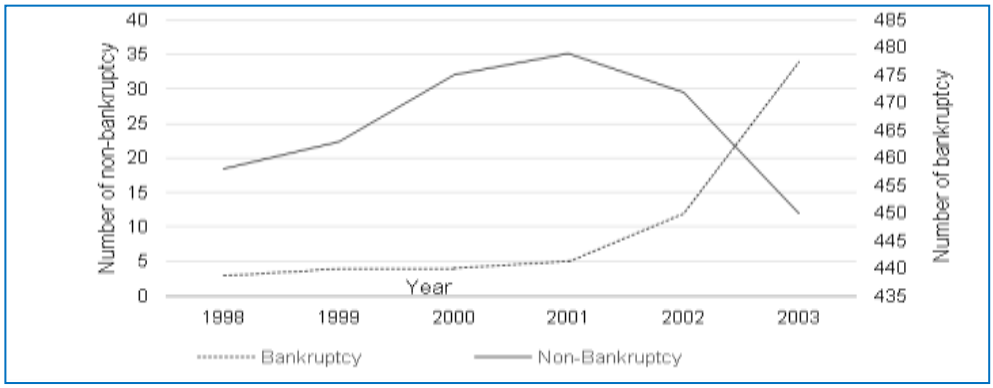

Figure 4. Year wise observation of bankruptcy in Spanish bankruptcy data

The number of instances in original training and test data before any preprocessing is shown in Table 4.

Table 4. Total Instances in Original Dataset

	Non-bankrupt	Bankrupt
Training data	2237	50
Test data	560	12

As shown in Table 4 the class distribution in original dataset is heavily skewed. Therefore, data sampling methods as discussed earlier are used to balance the class distribution. The number of class instances for training and testing data set after applying oversampling (SMOTE, BSMOTE, SLS, and ROS) and undersampling (RUS and CNN) methods are shown in Table 5 and Table 6 respectively.

Table 5. Number of instances after oversampling

		SMOTE	SLS	BSMOTE	ROS
Training data	Non-bankrupt	2237	2237	2237	2237
	Bankrupt	2237	2237	2237	2237
Test data	Non-Bankrupt	560	560	560	560
	Bankrupt	12	12	12	12

Table 6. Number of instances after under-sampling

		RUS	CNN
Training data	Non-bankrupt	50	114
	Bankrupt	50	50
Test data	Non-Bankrupt	560	560
	Bankrupt	12	12

The summary of empirically selected values of the various parameter used in individual and ensemble learners are shown in Table7.

Table 7. The summary of various parameter values set for learners

Learning Algorithms	Parameter Values							
	A	B	C	D	E	F	G	H
Bagging	100	100	REPTree	×	10	×	×	×
Bagging	100	100	c4.5	×	10	×	×	×
Bagging	50	100	REPTree	×	10	×	×	×
Bagging	50	100	c4.5	×	10	×	×	×
SVM	×	100	×	C-svc	×	RBF	×	×
SVM	×	100	×	C-svc	×	Linear	×	×
SVM	×	100	×	C-svc	×	Polynomial	×	×
SVM	×	100	×	C-svc	×	Sigmoid	×	×
Logistic	×	100	×	×	×	×	×	×
AdaBoostM1	×	100	DecisionStump	×	10	×	×	×
AdaBoostM1	×	100	c4.5	×	10	×	×	×
Random forest	50	×	CART	×	×	×	false	×
C4.5(J48)	×	100	×	×		×	×	2

Where, A is a Bag size, B Batch size, C classifier, D SVMtype, E Number of iteration, F Kernel type, G Cal. out of bag error, H Minimum number of the object, RBF stands for radial basis function and polySVM is SVM with kernel type polynomial, and DTBagging is bagging with C4.5.

The comparative performance of various data sampling techniques used in this work on individual learners and ensemble learners are recorded in Table 8 and 9.

Table 8 is used to show the average G-mean performance measure value evaluated for a different combination of classification algorithms and data sampling techniques on both

imbalance and balance dataset. The G-mean values suggested that in individual learners the performance of C4.5 (J48), and LR significantly improved with data sampling techniques. The best performance of LR and J48 is found with oversampling and undersampling techniques respectively. The SVM performs worst for both except with RUS. In the case of ensemble learners, AdaboostM1 and DTBagging perform well with oversampling techniques while RF performance is best with undersampling methods. The best performance of AdaboostM1, DTBagging, and RF is recorded with ROS, SLS, and RUS respectively. Overall the DTBagging with SLS recorded the highest value of average G-mean followed by RF with RUS.

Table 8. The G-mean values for different classifiers and sampling methods

		No Sampling	SMOTE	BSMOTE	SLS	ROS	RUS	CNN
Individual Learners	C4.5	0.055	0.739	0.638	0.675	0.612	0.823	0.755
	LR	0.233	0.838	0.769	0.805	0.817	0.752	0.708
	SVM	0.284	0.256	0.237	0.014	0.290	0.799	0.346
Ensemble Learners	AdaBoostM1	0.476	0.833	0.814	0.788	0.843	0.827	0.705
	DTBagging	0.330	0.885	0.760	0.906	0.686	0.845	0.652
	RF	0.562	0.741	0.594	0.694	0.582	0.902	0.804

Figure 5 shows the graphical representations of G-mean where x-axis represents sampling techniques and y-axis having G-mean of classifiers with the same interval as AUC. The graph shows that highest G-mean was found in Safe-level SMOTE sampling with DTBagging classifier. Table 9 is used to show the average value of the area under curve performance measure evaluated for a different combination of classification algorithms and data sampling techniques on both imbalance and balance dataset. The AUC values suggested that in individual learners LR with SLS outperformed the C4.5 and SVM except for C4.5 with CNN. Similarly, in the case of ensemble learners, DTBagging with SMOTE, BSMOTE, SLS and ROS outperform AdaboostM1 and RF while RF with undersampling methods RUS and CNN leads over AdaboostM1 and DTBagging. Overall DTBagging with oversampling methods and RF with undersampling methods perform superior while the performance of AdaboostM1 is moderate with both over and undersampling method regarding AUC performance measure values.

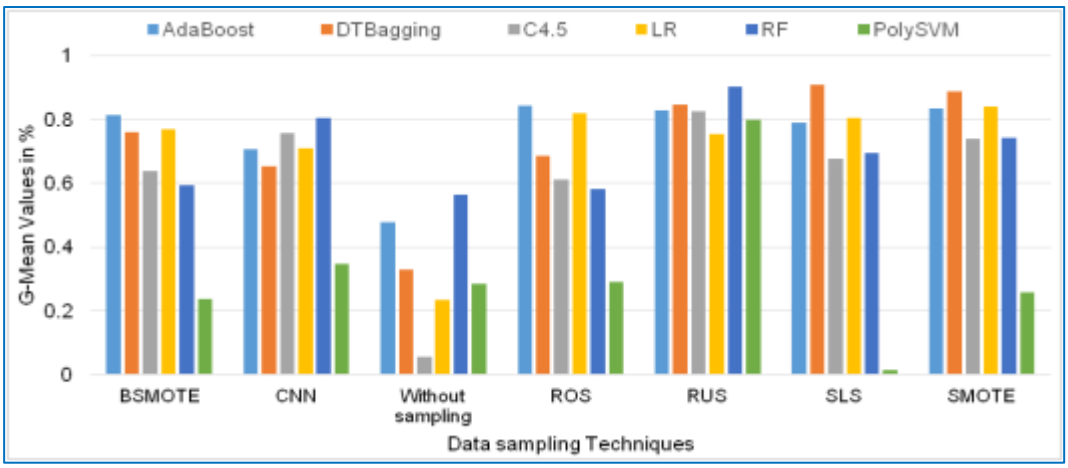

Figure 5. Graphical representation of G-mean values

The graphical representation of the AUC measure is shown in Figure 6 where x-axis shows the sampling techniques and the y-axis show the AUC values of different classifiers with the intervals of .1. The graph shows that DTBagging and RF are producing the best results with oversampling and undersampling techniques respectively.

Table 9. The AUC values for different classifiers and sampling methods

		No sampling	SMOTE	BSMOTE	SLS	ROS	RUS	CNN
Individual learners	C4.5	0.507	0.742	0.779	0.808	0.691	0.847	0.775
	LR	0.844	0.914	0.900	0.877	0.916	0.775	0.836
	SVM	0.552	0.526	0.478	0.492	0.516	0.801	0.511
Ensemble learners	AdaBoostM1	0.930	0.909	0.937	0.900	0.946	0.908	0.896
	DTBagging	0.913	0.990	0.998	0.988	0.998	0.930	0.913
	RF	0.685	0.960	0.966	0.948	0.977	0.959	0.945

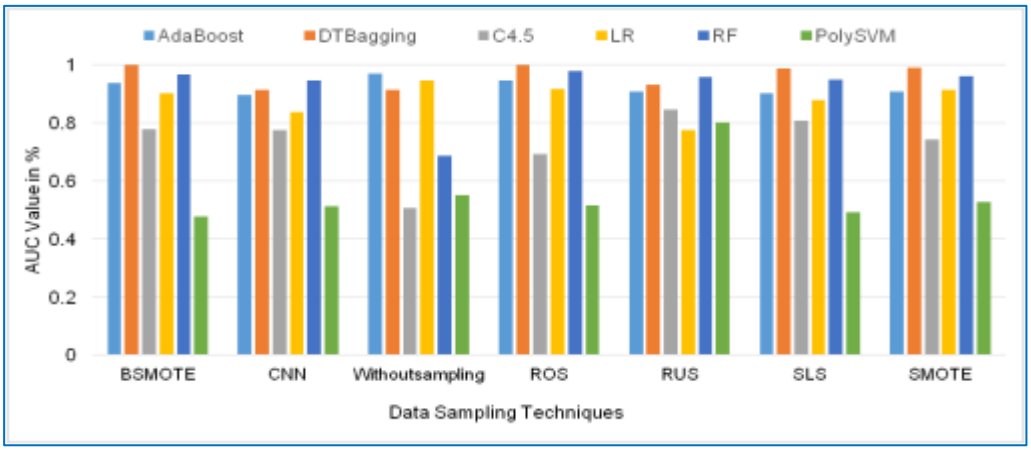

Figure 6. Graphical representation of AUC

5. Conclusion

In this paper, the effects of data sampling techniques are investigated on the performance of individual and ensemble learners using class imbalanced bankruptcy prediction dataset. Four oversampling including SMOTE, BSMOTE, SLS, and ROS and two undersampling techniques such as RUS and CNN are considered for investigation. Three individual (C4.5, LR, and SVM) and three ensembles (AdaboostM1, DTBagging, and RF) learners are used for experimentation. The extensive experiments are performed on highly imbalanced Spanish bankruptcy dataset using open source implementation of data sampling and learning techniques. The performance of prediction models is evaluated using G-mean and area under curve because other measures are not very useful for class imbalance problems. The G-mean values suggested that in individual learners logistic regression (LR) performs better with oversampling methods while C4.5 is better with undersampling methods. In the case of ensemble learners, AdaboostM1 and DTBagging perform well with oversampling techniques while RF performance is best with undersampling methods. The area under curve suggested that in individual learners LR with oversampling methods outperformed others. In ensemble learning DTBagging with oversampling methods and RF with undersampling methods perform superior while the performance of AdaboostM1 is moderate with both over and undersampling method regarding AUC performance measure

values. It can be concluded that universally no sampling method is superior, but the performance of oversampling (with LR and DTBagging) and undersampling (with C4.5 and RF) methods are superior as compare to others on this data set. The prediction model with oversampling and DTBagging outperformed the all other models.

6. References

[1]. Chen N, Chen A, Ribeiro B. Influence of class distribution on cost-sensitive learning: A case study of bankruptcy analysis. *Intell Data Anal* 2013; 17: 423–437

[2]. Kubat M, Holte RC, Matwin S. Machine learning for the detection of oil spills in satellite radar images. *Mach Learn* 1998; 30: 195–215

[3]. Rodda S, Erothi USR. Class Imbalance Problem in the Network Intrusion Detection Systems. In: *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016*. 2016, pp. 2685–2688

[4]. Liao TW. Classification of weld flaws with imbalanced class data. *Expert Syst Appl* 2008; 35: 1041–1052

[5]. Moepya SO. Applying Cost-Sensitive Classification for Financial Fraud Detection under High Class-Imbalance. 202. Epub ahead of print 2014. DOI: 10.1109/ICDMW.2014.141

[6]. Burez J, Van den Poel D. Handling class imbalance in customer churn prediction. *Expert Syst Appl* 2009; 36: 4626–4636

[7]. Fedorova E, Gilenko E, Dovzhenko S. Bankruptcy prediction for Russian companies: Application of combined classifiers. *Expert Syst Appl* 2013; 40: 7285–7293

[8]. Zhi W, Guo H, Fan M, et al. Instance-based ensemble pruning for imbalanced learning. *Intell Data Anal* 2015; 19: 779–794

[9]. Abdi L, Hashemi S. To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques. *IEEE Trans Knowl Data Eng* 2016; 28: 238–251

[10]. Kim MJ, Kang DK, Kim HB. Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Syst Appl* 2015; 42: 1074–1082

[11]. López V, Fernández A, García S, et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf Sci (Ny)* 2013; 250: 113–141

[12]. Chawla N, Bowyer K. SMOTE: Synthetic Minority Over-sampling Technique. *J Artif Intell Res* 2002; 16: 321–357

[13]. He H, Bai Y, Garcia EA, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proc Int Jt Conf Neural Networks* 2008; 1322–1328

[14]. Zhou L, Lai KK. AdaBoost Models for Corporate Bankruptcy Prediction with Missing Data. *Comput Econ DOI 101007/s10614-016-9581-4* 2016; 1–26

[15]. Quinlan JR. Bagging, boosting, and C4.5. *Proc Thirteen Natl Conf Artif Intell* 2006; 5: 725–730

[16]. Seiffert C, Khoshgoftaar TM, Hulse J Van, et al. RUSBoost: Improving classification performance when training data is skewed. In: *19th International Conference on Pattern Recognition*. 2008, pp. 8–11

[17]. L. Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Syst* 2013; 41: 16–25

[18]. Bellovary JL, Giacomino DE, Akers MD. A Review of Bankruptcy Prediction Studies: 1930 to Present. *J Financ Educ* 2007; 33: 1–42

[19]. William H . Beaver. Financial Ratios As Predictors of Failure. *J Account Res* 1966; 4: 71–111.

[20]. Ohlson J a. Financial Ratios and the Probabilistic Prediction of Bankruptcy. *J Account Res* 1980; 18: 109
[21]. Grice JS, Dugan MT. The Limitations of Bankruptcy Prediction Models : Some Cautions for the Researcher. *Rev Quant Financ Account* 2001; 17: 151–166
[22]. Atiya AF. Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Trans Neural Networks* 2001; 12: 929–935
[23]. K-S, Lee Y-J. A genetic algorithm application in bankruptcy prediction modeling. *Expert Syst Appl* 2002; 23: 321–328
[24]. Hutchison D, Mitchell JC. *Applications of Evolutionary Computing*. 2006. Epub ahead of print 2006. DOI: 10.1007/3-540-45365-2
[25]. Mora A. Spanish Bankruptcy Datasethttps://github.com/amorag/Bankruptcy_2016 (2016)
[26]. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Int Jt Conf Artif Intell* 1995; 14: 1137–1143.
[27]. Han H, Wang W-Y, Mao B-H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Adv Intell Comput* 2005; 17: 878–887
[28]. Quinlan JR. *C4. 5: programs for machine learning*. Elsevier, 2014 Jr H, W. D, Lemeshow S, et al. *Applied logistic regression*. John Wiley & Sons, 2013.
[29]. Jr H, W, D, Lemeshow S, et al. *Applied logistic regression*. John Wiley & Sons, 2013
[30]. Chang C-C, Lin C-J. LIBSVM : a library for support vector machines. *ACM Trans Intell Syst Technol* 2011; 2: 1–27.
[31]. Sisodia DS, Verma S, Vyas OP. Agglomerative Approach for Identification and Elimination of Web Robots from Web Server Logs to Extract Knowledge about Actual Visitors. *J Data Anal Inf Process* 2015; 3: 1–10
[32]. Breiman L. Bagging predictors. *Mach Learn 242, Springer* 1996; 24: 123–140
[33]. Freund Y, Schapire RRE. Experiments with a New Boosting Algorithm. In: *International Conference on Machine Learning*. 1996, pp. 148–156.
[34]. Breiman L. Random Forests. *Mach Learn* 2001; 45: 5–32
[35]. Ali A, Shamsuddin SM, Ralescu AL. Classification with class imbalance problem: A review. *Int J Adv Soft Comput its Appl* 2015; 7: 176–204.
[36]. Giang Hoang Nguyen ABSLP. Learning Pattern Classification Tasks with Imbalanced Data Sets, Pattern Recognition,. In: (Ed.) P-YY (ed) *Pattern Recognition*. InTech, China, 2009, p. 568.
[37]. Kubat M, Matwin S. Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In: *Icml*. 1997, pp. 179–186
[38]. Huang J, Ling CX. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans Knowl Data Eng* 2005; 17: 299–310
[39]. Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett* 2006; 27: 861–874
[40]. J A-F, L S, S G, et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput* 2009; 13: 307–318.
[41]. Alcalá-Fdez J, Fernández A, Luengo J, et al. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J Mult Log Soft Comput* 2011; 17: 255–287
[42]. Frank E, Mark A. Hall I, Witten an H. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques(Weka ML Tool)*. Morgan Kaufmann, 2016

Dilip Singh Sisodia received Ph.D. degree in computer science and engineering from the National Institute of Technology Raipur (An autonomous institute of national importance under the ministry of HRD, Govt. of India), India. He did his Master of Technology and Bachelor of Engineering degrees respectively in information technology (with specialization in artificial intelligence) and computer science & engineering from the Rajiv Gandhi Proudyogiki Vishwavidyalaya (A State Technological University of M.P.), Bhopal, India. Presently, Dr. Sisodia is working as an assistant professor in the department of computer science engineering, National Institute of Technology Raipur. He has over sixteen years of experience of various reputed engineering institutes in the field of academics & research. He has published over 50 refereed articles in journals, conference proceedings and books, published by reputed publishers including IEEE, Springer, Elsevier, SAGE, IOS press and IGI Global etc. He is also working as an active reviewer for many international journals and conferences. His current research interests include the web usage mining, machine learning, and computational intelligence. Dr. Sisodia is actively associated with various professional societies including IEEE, ACM, CSI, IETE, IE (India) etc.

Upasana Verma received her bachelor's degree in computer science & engineering from Swami Vivekananda Technical University (CSVTU) Chhattisgarh and Master of Technology from National Institute of Technology, Raipur.