International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Evaluation of Classification Algorithms with Solutions to Class Imbalance Problem on Bank Marketing Dataset using WEKA

Archit Verma[1]

[1]*M.Tech, Computer Science and Engineering, from United College of Engineering and Research, Prayagraj Uttar Pradesh, India*

Abstract – *Data Mining is the non-trivial extraction of information from data that are implicit, previously unknown, and potentially useful .Classification is a supervised learning technique in data mining. Classification algorithm predicts the categorical class label of a data instance, so as to classify it into a predetermined class. Here we will consider class imbalance problem and solutions for handling class imbalance problem, that is faced by the classification algorithm. Class imbalance problem occurs when one of the two classes in the dataset has a very small number of samples compared to the other class, in such datasets the prediction result is biased towards majority class, but the class of interest is the minority class. We will use Bank Marketing Dataset for our evaluation that is a class imbalance dataset. We will use WEKA for evaluation of various classification algorithms. WEKA is a machine learning tool written in Java. The algorithms we will consider are Decision Tree(C 4.5), Naïve Bayes, Multilayer Neural Network, Support Vector Machine(with SMO as optimization technique) , and Logistic Regression. The techniques to handle class imbalance problem discussed by us are sampling based technique, and algorithm based technique such as Random Forest. All these algorithms without and with solutions to class imbalance problems will be evaluated on evaluation metrics such as Precision ,Recall,F1-Score, ROC and AUCPR each of minority class. WEKA provides us facility to artificially balance class imbalanced dataset by applying sampling filters in preprocessing tab.*

Key Words: *Data Mining, Classification, Class Imbalance Problem ,Minority Class, Decision Tree(C 4.5) ,Naïve Bayes, Multilayer Neural Network, Support Vector Machine(SMO),Logistic Regression, Random Forest, Random Under-Sampling of majority class, Random Over-Sampling of minority class, SMOTE, Precision, Recall, F1-Score, ROC, AUCPR, WEKA, SpreadSubsample filter ,Resample filter and SMOTE filter*

1. INTRODUCTION

Data Mining is a non-trivial process of identifying patterns in data that are true on new data, previously unknown, potentially useful, and ultimately understandable.[11][12] It is also referred to as knowledge discovery from data. Classification is a supervised learning technique, it is used to predict the categorical class label of a given data instance, so as to classify it into a predetermined class. It is a two step process, in first step classification algorithm uses training data to build a classifier, and then in second step it uses this classifier to predict the class label of a given unlabeled data instance.

The classification algorithms we will consider here are Decision Tree (C 4.5), Naïve Bayes, Multilayer Neural Network, Support Vector Machine and Logistic Regression. We will analyze a research problem in data mining classification algorithm, that is class imbalance problem, which happens when one of the two classes has very less number of samples compared to the other class and the class of interest is the minority class. In this case the prediction is biased towards the majority class. We will discuss two approaches of handling class imbalance problem, namely sampling based approach and algorithm based approach.

We will evaluate classification algorithm without and with applying solutions to class imbalance problem using WEKA, on bank marketing dataset. The data in this dataset is related with direct marketing campaigns of a Portuguese banking institution. [6] The classification goal is to predict if the client will subscribe a term deposit. This is a class imbalance dataset, in which 'yes' is the label of the minority class.

Sampling based approach discussed and evaluated by us is, random under-sampling of majority class, random over-sampling of minority class, and Synthetic Minority Over-Sampling Technique (SMOTE).

Algorithm based approach discussed by us is Random Forest. We will also evaluate Random forest with sampling.

2. CLASSIFICATION ALGORITHM

2.1. DECISION TREE

Decision tree learning algorithm is a top-down recursive divide and conquer algorithm. Decision tree is a tree like structure in which internal nodes are labeled by attributes and outgoing edges denotes test condition on that attributes while leaf node denote classes. Nodes labeled by attributes divide the training dataset into two or more subsets. The attribute selection at each stage is done by an attribute selection measure that divides the training dataset best. Then decision tree algorithm works recursively on remaining subsets of data. The algorithm terminates until we get all tuples in same class in which we label leaf by that class, or other termination criteria that we get is attribute

list empty in which we label leaf by majority class of tuples in that subset. Here we will use decision tree C 4.5 for our evaluation purpose that is based on gain ratio. [7]

2.2. Naïve Bayes

Naive bayes classification algorithm is based on baye's theorem of posterior probability. This algorithm works by predicting probability that a given data instance belongs to a particular class .The probability that a given data vector is in class C , is called posterior probability and is denoted by P(C|X).Finally Maximum a posteriori hypothesis is applied.[7]

2.3. MULTILAYER NEURAL NETWORK

Neural network is a set of connected nodes that is an input/output units, and each connection has a weight associated with it .The network learns by adjusting its weight according to training data vectors. The networks adjusts its weight so as to minimize the mean squared error between predicted class label and actual class label, for this it employs gradient descent method. The weights are adjusted by backpropagation algorithm.

The network has three layers input layer, hidden layer and output layer. The nodes in hidden layer and output layer has sigmoid function. [7]

2.4. SUPPORT VECTOR MACHINE

Support vector machine SVM is a supervised learning based, non-probabilistic algorithm for classification. A support vector machine constructs a hyperplane which can be used for classification. This algorithm can also work on non-linear classification by mapping data points to a higher dimension by kernel technique. New data instance is then mapped into that same space and predicted to belong to a class based on which side of the gap they fall.

The weights vector in the hyperplane can be adjusted so that the hyperplanes defining the "sides" of the margin can written as:

$$H1: \vec{x}_i.\vec{w} + w_0 \geq +1 \quad \text{for } y_i = +1 \tag{1}$$

$$H2: \vec{x}_i.\vec{w} + w_0 \leq -1 \quad \text{for } y_i = -1 \tag{2}$$

Optimization technique is used for maximizing $\frac{2}{||w||}$ that is distance between $H_1$and H_2, this equation can be written in constrained (convex) quadratic optimization problem. Hyperplane with larger margin is found to be more accurate than hyperplane with lower margin.

WEKA uses Sequential minimal optimization (SMO) as optimization technique.[7]

2.5. LOGISTIC REGRESSION

[4]Logistic Regression is a classification technique that works on the association between categorical dependent variable and a set of independent variables. Dependent variable has only two values, such as 0/1 or Yes/No. It is used to predict this binary outcome. Logistic regression works by estimating the parameters of a logistic model. In the logistic model, the log-odds for the value labeled "1" is a linear combination of one or more independent variables.

$$logit(p) = log(\frac{p}{1-p}) \tag{3}$$

$$logit(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{4}$$

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{5}$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}} \tag{6}$$

To calculate the cost function of logistic regression of m samples we do the following

$$L = Cost(h_\beta, y) = \frac{1}{m}\sum_{i=1}^{i=m}(-y_i log(h_\beta(x_i)) - (1-y_i)log(1-h_\beta(x_i))) \tag{7}$$

where $$h_\beta(x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}} \tag{8}$$

Then β_i are updated by gradient descent rule.

3. DATESET

For our evaluation purpose we will use, a class imbalance dataset. Bank Marketing Dataset is one such dataset.

This dataset is available at https://www.openml.org/d/1461 the data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed. The classification goal is to predict if the client will subscribe a term deposit (variable y). [6]

The number of instance in minority class is 5289(yes) and number of instances in majority class is 39922(no).

Table -1: Attribute Description of Bank Marketing Dataset

Details of bank marketing dataset	
Attribute	Description
Age	(numeric)

Job	Type of job, "management" ,"housemaid", "entrepreneur" , "student", "blue-collar", "self-employed" , "retired", "technician", "services")
Marital	marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed
Education	(categorical: "unknown","secondary","primary","tertiary")
Default	Has credit in default? (binary: "yes", "no")
Balance	average yearly balance, in euros (numeric)
Housing	Has housing loan? (binary: "yes", "no")
Loan	Has personal loan? (binary: "yes", "no")
Contact	contact communication type (categorical: "unknown","telephone","cellular")
Day	last contact day of the month (numeric)
Month	last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")
Duration	last contact duration, in seconds (numeric)
Campaign	number of contacts performed during this campaign and for this client (numeric, includes last contact)
Pdays	number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
Previous	number of contacts performed before this campaign and for this client (numeric)
Poutcome	outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")
Class (Y) Has the client subscribed a term deposit? (binary: "yes", "no")	

4. CLASS IMBALANCE PROBLEM

[3][9]There can be an imbalance dataset provided for classification. By imbalance dataset we mean that one of the two class has very less number of samples compared to number of samples in the other class,$|C_2| << |C_1|$.The C_2 is called the minority class, and C_1 is called the majority class. The minority class is of our interest. The machine learning algorithm always performs well if it is given by balanced dataset, but this is not always the case , as an example the dataset for fraud detection ,will have less number of fraud transactions than genuine transaction. Anomaly detection, medical diagnostic and fault monitoring are other examples. The prediction in case of unbalanced dataset is biased towards majority class. The two approaches to solve this problem are, sampling based approach and algorithm based approach.

4.1. SAMPLING BASED APPROACH

[3][9]This is also known as data level approach. It works by artificially balancing the instances of class in the dataset. To artificially balance the class we apply resampling technique, such as random under sampling the majority class, random oversampling of minority class, and Synthetic Minority Over-Sampling Technique (SMOTE).

4.1.1. RANDOM UNDERSAMPLING OF MAJORITY CLASS

[3][9]In this approach we try to balance the class distribution in dataset by randomly throwing some data samples from majority class. Although it balances class distribution, but it leads to losing some important characteristics in dataset, due to removal of some samples, this is a disadvantage of this approach.

4.1.2. RANDOM OVERSAMPLING OF MINORITY CLASS

[3][9]In this approach we balance the class distribution by the replication of minority class samples. The problem with this approach is that it leads to overfitting.

4.1.3. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

[5]To reduce the problem of overfitting a method of creating synthetic instances is used. This techniques is known as the synthetic minority over-sampling technique (SMOTE). In this the training set is altered by adding synthetically generated minority class instances, causing the class distribution to become more balanced. The instances are said to be synthetic, as they are new minority instances that has being created out of existing minority class instances. In order to create the new synthetic minority class instances, SMOTE first selects an instance of minority class at random say 'x' and proceeds by finding its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors say 'y' at random and connecting 'x' and 'y' to form a line segment in the feature space. The synthetic instance say 'z' is generated as a convex combination of the two chosen instances 'x' and 'y'.

z.attribute=x.attribute+(y.attribute-x.attribute)*rand(0,1).

4.2. ALGORITHM BASED APPROACH

[1]An ensemble for classification is a composite model, made up of a combination of classifiers.

The individual classifiers vote, and a class label prediction is returned by the ensemble based on the collection of votes.

Ensembles tend to be more accurate than their component classifiers. The general ensemble methods are Bagging, boosting, and random forests.

The term bagging refers to bootstrap aggregation.
For a dataset, D, of d number of tuples, bagging proceeds as follows. For iteration I (i=1, 2,..., k), a training set, Di, is formed by sampling d tuples with replacement from the initial set of tuples, D.

Each training set is a bootstrap sample. Because sampling with replacement is used, some of the original tuples of D may not come in D_i, whereas some may occur more than once. A classifier model, M_i, is generated for each training set, D_i. Now if we want to classify an unknown tuple, X, each classifier, M_i, returns its predicted class, which gives one vote. The bagged classifier, M*, counts the votes and assigns the class with the majority votes to X.

In boosting, we assign weight to each training tuple. Then k classifiers is iteratively learned sequently one after the other. After a classifier, M_i, is learned, the weights are updated to allow the next classifier, Mi_{+1}, to be attentive towards the training tuples that were misclassified by M_i. The concluding boosted classifier, M*, combines the votes of each M_i's, where the weight of each classifier's vote is computed in terms of its accuracy.

Random forests or random decision forests are an ensemble learning method for classification, that operate by constructing a number of decision trees at training time and outputting the class that is the majority of the classes outputted from individual trees. Random decision forests correct the decision trees problem of overfitting to their training set. The random trees are created by random selection of attributes at each split, and random selection of tuples from the dataset. [7]

5. EVALUATION METRICES

[1]In holdout method for evaluation we split the data set into two sets, training dataset and test dataset. The splitting is generally two-third training and one-third test. We use training data set to build the classifier and then we use this classifier for prediction on test dataset.

When we are evaluating classification algorithm on class imbalance datasets, evaluation metrics like precision, recall, f1-score, ROC and AUCPR of minority class becomes important.

TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative

Precision: It is the measure of exactness, which is the percentage of tuples in test dataset that are labeled as positive, are actually positive.

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

Recall: It is the measure of exactness, which is the percentage of positive tuples in test dataset that the classifier labeled as positive.

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

F1-Score: It is the harmonic mean of precision and recall.

$$F - Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (11)$$

ROC: Receiver operating characteristic curves are a visual tool that helps us in comparison of two classification models. An ROC curve for a given model brings out the trade-off between the true positive rate (TPR) and the false positive rate (FPR).For a given test set and a model, TPR is the proportion of positive tuples that the models labels correctly, FPR is the proportion of negative tuples that are mislabeled as positive. So $TPR = \frac{TP}{TP+FN}$ which is sensitivity. Furthermore, $FPR = \frac{FP}{FP+TN}$. Any increase in TPR occurs at the cost of an increase in FPR. The area under the ROC curve is a measure of the accuracy of the model.

AUCPR: It is the area under precision recall curve. This measure is generally used to measure the accuracy of the classification model on class imbalance dataset.

6. WEKA

[10][13]Waikato Environment for Knowledge Analysis (WEKA) is software for machine learning written in Java, developed at the University of Waikato. This is free software and is licensed under the GNU General Public License.

WEKA provides us with many data mining algorithms and visualization tools for data analysis and predictive modeling. Its graphical user interfaces makes easy for user to run these algorithms on training datasets. WEKA supports several standard data mining tasks that are, data preprocessing, classification, regression, clustering, association, attribute selection and visualization. The classification algorithms that we will consider here are: Decision Tree (C 4.5) by J48 inside trees, Naive Bayes inside Bayes, Multilayer Neural Network by Multilayer Perceptron inside functions, SVM by SMO(PolyKernel) inside functions, Logistic Regression by Logistic inside functions and Random Forest inside trees. WEKA outputs precision, recall, f1-score, ROC, AUCPR of all classes separately and combined weighted average also. The

input datasets are provided in ARFF file format. ARFF stands for Attribute-Relation File Format.

For handling class imbalance problem we use filters in preprocess tab, we choose supervised instance filters.

For random under sampling of majority class we use (weka.filters.supervised.instance.SpreadSubsample) which produces a random subsample of a dataset. The original dataset must fit entirely in memory. This filter allows you to specify the maximum "spread" between the rarest and most common class. We set distributionSpread=1, which denotes uniform distribution spread. After applying the filter to dataset the produced dataset contains 5289 instances of minority class and 5289 instances of majority class.

For random over sampling of minority class we use (weka.filters.supervised.instance.Resample) which produces a random subsample of a dataset using either sampling with replacement or without replacement. The original dataset must fit entirely in memory. The number of instances in the generated dataset may be specified. We set biasToUniformClass=1, which denotes whether to use bias towards a uniform class. A value of 0 leaves the class distribution as-is, a value of 1 ensures the class distribution is uniform in the output data. For sampling with replacement we set noReplacement=false. After applying the filter the sampled dataset produced has 22605 instances of minority class and 22605 instances of majority class.

For Synthetic Minority Over-sampling Technique we use (weka.filters.supervised.instance.SMOTE) which resamples a dataset by applying the Synthetic Minority Oversampling Technique (SMOTE). The original dataset must fit entirely in memory. The amount of SMOTE and number of nearest neighbors may be specified. In our dataset we set ClassValue=0 to auto-detect the non-empty minority class ,nearestNeighbors=5 and apply SMOTE filter three times in first time we put percentage=100 which increases number of instances of minority class from 5289 to 10578,second time we again apply SMOTE filter with percentage=100 which increase minority class instance from 10578 to 21156,third time we again apply SMOTE filter with percentage=88.7 which increases minority class instance from 21156 to 39921,and finally our dataset becomes balanced for classification with 39922 instances of majority class and 39921 instances of minority class.

For testing purpose we split the dataset as 66% training and 34% test dataset.

Finally the classification algorithms in WEKA is executed on these sampled dataset as input and finally readings of Precision, Recall, F1-Score, ROC and AUCPR of minority class outputted by WEKA are noted down, and graphically shown.

Table -2: Dataset preprocessed before evaluation

	Instances of Majority Class	Instances of Minority Class
Without Handling Class imbalance problem(No Filter Applied)	39922	5289
Random Under Sampling of Majority Class (Filter Applied: SpreadSubsample)	5289	5289
Random Over Sampling of Minority Class (Filter Applied: Resample)	22605	22605
SMOTE (Filter Applied: SMOTE)	39922	39921

7. RESULT OF EVALUATION FROM WEKA

Table -3: Precision of Minority class of Class Imbalance Datasets

Precision of Minority Class				
Algorithm	**Without Handling Class Imbalance Problem**	**Random Under Sampling of Majority Class**	**Random Over Sampling of Minority Class**	**SMOTE**
Decision Tree(C 4.5)	0.606	0.819	0.882	0.917
Naïve Bayes	0.493	0.766	0.797	0.81
Multilayer Neural Network	0.556	0.824	0.859	0.894
SVM(SMO)	0.626	0.833	0.837	0.882
Logistic Regression	0.641	0.833	0.841	0.885
Random Forest	0.623	0.838	0.922	0.935

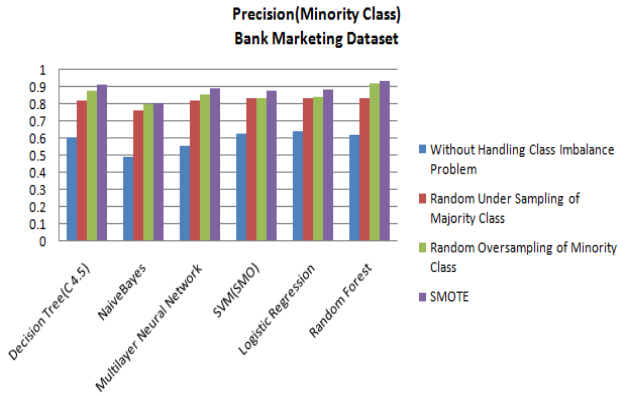

Chart -1: Precision of minority class of Bank Marketing Dataset

Table -4: Recall of Minority class of Class Imbalance Datasets

Recall of Minority Class				
Algorithm	**Without Handling Class Imbalance Problem**	**Random Under Sampling of Majority Class**	**Random Over Sampling of Minority Class**	**SMOTE**
Decision Tree(C 4.5)	0.477	0.892	0.958	0.935
Naïve Bayes	0.523	0.848	0.757	0.917
Multilayer Neural Network	0.421	0.811	0.921	0.924
SVM(SMO)	0.18	0.826	0.834	0.905
Logistic Regression	0.345	0.826	0.823	0.9
Random Forest	0.411	0.902	0.987	0.947

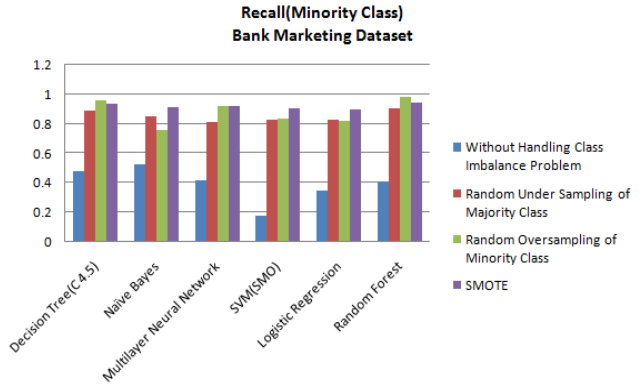

Chart -2: Recall of Bank Marketing Dataset

Table -5: F1-Score of Minority class of Class Imbalance Datasets

F1-Score of Minority Class				
Algorithm	**Without Handling Class Imbalance Problem**	**Random Under Sampling of Majority Class**	**Random Over Sampling of Minority Class**	**SMOTE**
Decision Tree(C 4.5)	0.534	0.854	0.918	0.926
Naïve Bayes	0.508	0.805	0.776	0.86
Multilayer Neural Network	0.479	0.817	0.889	0.908
SVM(SMO)	0.279	0.829	0.835	0.893
Logistic Regression	0.449	0.829	0.832	0.892
Random Forest	0.495	0.869	0.953	0.941

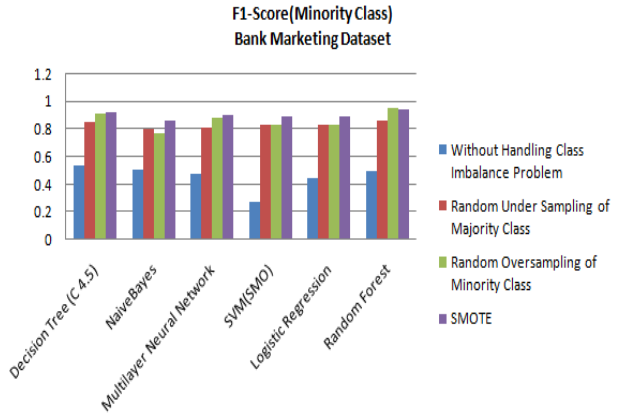

Chart -3: F1-Score of Bank Marketing Dataset

Table -6: ROC of Minority class of Class Imbalance Datasets

ROC of Minority Class				
Algorithm	**Without Handling Class Imbalance Problem**	**Random Under Sampling of Majority Class**	**Random Over Sampling of Minority Class**	**SMOTE**
Decision Tree(C 4.5)	0.839	0.879	0.938	0.953
Naïve Bayes	0.861	0.87	0.855	0.92
Multilayer Neural Network	0.868	0.89	0.932	0.959
SVM(SMO)	0.583	0.831	0.835	0.89
Logistic Regression	0.907	0.909	0.911	0.949
Random Forest	0.923	0.926	0.994	0.988

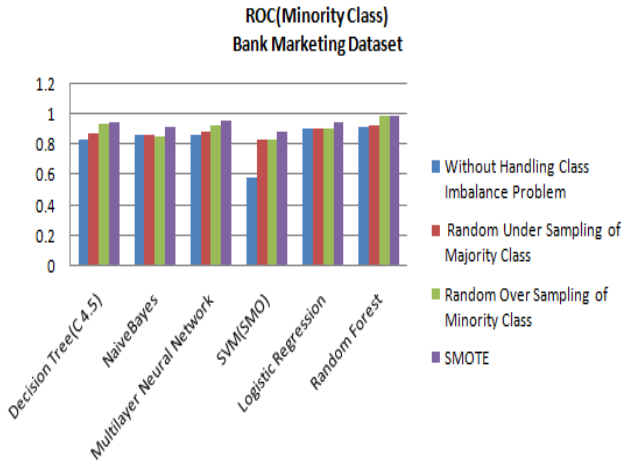

Chart -4: ROC of Bank Marketing Dataset

Table -7: AUCPR of Minority class of Class Imbalance Datasets

AUCPR of Minority Class				
Algorithm	**Without Handling Class Imbalance Problem**	**Random Under Sampling of Majority Class**	**Random Over Sampling of Minority Class**	**SMOTE**
Decision Tree(C 4.5)	0.492	0.813	0.9	0.938
Naïve Bayes	0.453	0.857	0.829	0.9
Multilayer Neural Network	0.506	0.87	0.91	0.951
SVM(SMO)	0.209	0.774	0.781	0.846
Logistic Regression	0.555	0.888	0.893	0.935
Random Forest	0.585	0.901	0.995	0.989

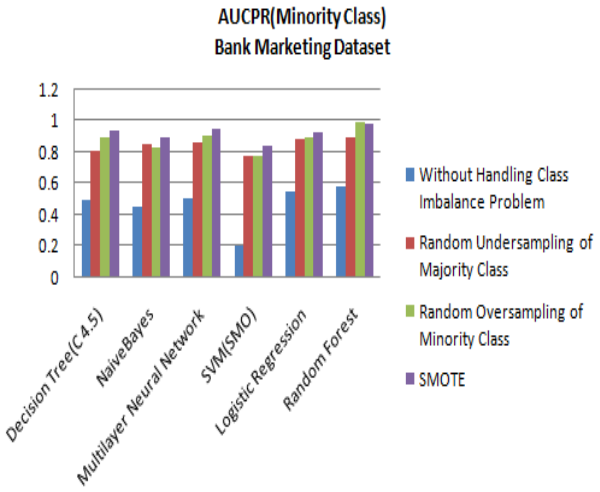

Chart -5: AUCPR of Bank Marketing Dataset

8. IMPORTANT POINTS FROM EVALUATION RESULTS OF WEKA

1.)From chart-1 of precision of minority class we see that sampling based technique improves precision of minority class, we see that Classification Algorithm(Decision Tree(C 4.5), Naïve Bayes , Multilayer Neural Network, Support Vector Machine (SMO), Logistic Regression and Random Forest) gives the highest precision with SMOTE compared to other sampling technique.

2.) From chart-2 of recall of minority class we see that sampling based technique improves recall of minority class , we see that Classification Algorithm(Naïve Bayes , Multilayer Neural Network, Support Vector Machine (SMO), and Logistic Regression) with SMOTE gives the highest recall while (Decision Tree(C 4.5), Random Forest) gives highest recall with random over sampling of minority class.

3.) From chart-3 of f1-score of minority class we see that sampling based technique improves f1-score of minority class , we see that Classification Algorithm(Decision Tree(C 4.5),Naïve Bayes , Multilayer Neural Network, Support Vector Machine (SMO), and Logistic Regression) with SMOTE gives highest f1-score,while Random Forest gives highest f1-score with random over sampling of minority class.

4.)From chart-4 of ROC of minority class we see that sampling based technique improves ROC of minority class , we see that Classification Algorithm(Decision Tree(C 4.5),Naïve Bayes , Multilayer Neural Network, Support Vector Machine (SMO), and Logistic Regression) with SMOTE gives highest ROC ,while Random Forest gives highest ROC with random over sampling of minority class.

5.)From chart-5 of AUCPR of minority class we see that sampling based technique improves AUCPR of minority class , we see that Classification Algorithm(Decision Tree(C 4.5),Naïve Bayes , Multilayer Neural Network, and Support Vector Machine (SMO) and Logistic Regression) with SMOTE gives highest AUCPR, while Random Forest gives highest AUCPR with random over sampling of minority class.

9. CONCLUSIONS

Classification is a supervised learning technique in data mining. It predicts the categorical class label of a data instance so as to classify it into a predetermined class. Class imbalance problem is one of the most important research problem in classification, in which the class of interest is the minority class, and has very less samples compared to the majority class. This leads classifier prediction to be biased towards majority class, so solutions needs to be found out to handle this problem. Here we have evaluated solutions to class imbalance problem on bank marketing dataset using WEKA.

We have used WEKA to preprocess dataset so as to balance class distribution, using filters.

For random under-sampling of majority class we have used “SpreadSubsample” filter, for random over-sampling of minority class we have used “Resample” filter and for synthetic minority over-sampling technique we have used “SMOTE” filter. From evaluation results of WEKA we see that sampling based technique increases precision, recall, f1-score, ROC and AUCPR of minority class of various classification algorithms.

We have also read that that random oversampling of minority class suffers from overfitting, that is corrected by SMOTE. In case of random undersampling of majority class

in spite of the fact that it leads to the loss of some important characteristics in dataset, due to removal of some samples, it has also increased accuracy at a considerable level.

Our evaluation results of WEKA for precision metric shows that Classification Algorithm (Decision Tree(C 4.5), Naïve Bayes, Multilayer Neural Network, and Support Vector Machine (SMO), Logistic Regression and Random Forest) with SMOTE gives highest value.

Our evaluation results of WEKA for recall metric shows that Classification Algorithm(Naïve Bayes , Multilayer Neural Network, and Support Vector Machine (SMO) ,Logistic Regression) with SMOTE gives highest value, while (Decision Tree(C 4.5) and Random Forest) gives highest value with random over sampling of minority class.

Our evaluation results of WEKA for metrics F1-Score, ROC ,and AUCPR of minority class shows that for Classification Algorithm(Decision Tree(C 4.5),Naïve Bayes , Multilayer Neural Network, and Support Vector Machine (SMO) and Logistic Regression) with SMOTE gives highest value, while Random Forest gives highest value with random over sampling of minority class.

REFERENCES

1) Book, Jiawei Han, Micheline Kamber Data Mining Concepts and Techniques 2nd edition
2) Mr.Rushi Longadge, Ms. Snehlata S. Dongre, Dr. Latesh Malik, "Class Imbalance Problem in Data Mining: Review", International Journal of Computer Science and Network (IJCSN) Volume 2, Issue 1, February 2013
3) http://www.chioka.in/class-imbalance-problem/
4) https://medium.com/deep-math-machine-learning-ai/chapter-2-0-logistic-regression-with-math-e9cbb3ec6077
5) Nitesh V. Chawla "Data Mining For Imbalanced Datasets"
6) https://www.openml.org/d/1461
7) Archit Verma, "Study and Evaluation of Classification Algorithms in Data Mining", International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 08, August 2018
8) Justice Asare-Frempong,Manoj Jayabalan, "Predicting Customer Response to Bank Direct Telemarketing Campaign", 2017 IEEE The International Conference on Engineering Technologies and Technopreneurship
9) https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/
10) https://en.wikipedia.org/wiki/Weka_(machine_learning)
11) Book,Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar ,"Introduction to Data Mining, Pearson Education", 2016
12) http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html
13) Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, David Scuse WEKA "Manual for Version 3-8-2".
14) M. Purnachary,B. Srinivasa S P Kumar, Humera Shaziya "Performance Analysis of Bayes Classification Algorithms in WEKA Tool using Bank Marketing Dataset" International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 2, February 2018

BIOGRAPHY


Archit Verma

I am M.Tech (C.S.E.) from United College of Engineering and Research, Prayagraj (Allahabad) in 2017, B.Tech (C.S.E.) from Institute of Engineering and Rural Technology, Prayagraj (Allahabad). I am UGC-NET qualified in Computer Science and Applications for Assistant Professor in January 2018. My area of interest includes Data Mining, Artificial Intelligence, and Big Data Analytics. I have taught Cloud Computing and Software Testing and Audit at Institute of Engineering and Technology (I.E.T.), Lucknow in odd semester of 2018.