ENSEMBLES FOR CLASS IMBALANCE PROBLEMS IN VARIOUS DOMAINS

Deepakindresh N, Gauthum J, Jeffrin Harris,
Harshavardhan J and Shivaditya Shivganesh

School of Computer Science Engineering,
Vellore Institute of Technology, Chennai, India

ABSTRACT

The paper is an analysis of class imbalance problems from various domains such as the medical field, sentiment analysis, software de-fects, water portability, and relationship status of students and summarizes the performance of data resampling techniques such as random undersampling and oversampling. Synthetic minority oversampling techniques combined with the power of ensemble methods such as bagging, boosting, and hybrid techniques are generally used to solve the class imbalance problem.

KEYWORDS

Machine Learning, Class Imbalance, Multi Domain Analysis.

1. INTRODUCTION

Data being the most important aspect of machine learning and data science, without which no matter how strong or complex models are used, it cannot solve a problem without having an abundant size of data. Although in many cases this is not true, for instance, real-time data does not always promise equally skewed data for all classes since one class might dominate over the other class and hence lead to class imbalance wherein the smaller class is misconstrued as noise in the dataset, thus misleading machine learning algorithms. This is a major problem as the models would generally underfit the data and if tweaked with a lot of hyperparameter tuning, it again quickly escalates to overfitting, which is why data resampling techniques are used where the data corresponding to the lower class are either randomly resampled or oversampled using nearest neighbors and the other method is under-sampling the higher classes. Both have their advantages and they are analyzed in this paper. Apart from data resampling techniques, another great way to counteract the class imbalance problem is using ensemble models which combine the advantages of many models to give priority and weights to the models that perform best in one particular subset of the data than others.

Ensembles for class-imbalance problems (ECIP)[7] are a subset of ensembles that use data resampling to pre-process imbal-anced data before learning. The performance ofvarious ECIPs is experimentally compared in this work utilizing the performance metrics F1 scoreand g-Mean over five datasets that predict the rate of heart failure, customer satisfaction, software defect, student relationship and water potability. Each of these datasets from different domains has a class imbalance problem due to which nor-mal stand-alone models suffer from true positive identification which is identifying the lower class.

David C. Wyld et al. (Eds): SIP, CRBL, WiMNeT, BIOM, EDUPT, NLDM, SOFTFM, COMIT - 2022
pp. 173-180, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121718

2. DATA ANALYSIS

Data was collected from kaggle and it can be clearly seen from the graphs how big of a class imbalance problem is present in the dataset. For preprocessing columns with no correlation to the dataset were removed with a threshold set to less than 0.2 and ows with NA values were removed. For software defect, dataset with features such as lines of code etc was used to predict if the data has software defects or not, from the [Fig 1] class imbalance problem is visible where the system with no defects is the higher class and with defects is the subordinate class. The data in both the classes are approximately in an 11:89 ratio which could cause tremendous bias in the classification.

For students, dataset [2] the response variable is the relationship status of the studentwhich is predicted using features such as age, gen-der, marks, parents' occupation etc and as seen from the [Fig 5] the ratio of single students to com-mitted is 6:19.

For the Heart failure dataset, various features were used to determine if the patient suffers from heart disease or not and the class imbalance ratio is close to 14:86. The data split can be seen in [Fig 2].

For water, potability [4] dataset the features used where chemical quanti-ties such as ph values etc, and class imbalance are visible from [Fig 4] where the imbalance ratio is 4:21%.

For the airline, [3] dataset the features were ratings on each aspect such as reviews on maintenance, safety, communication, food quality, etc were used to predict the total re-view whether the travel experience was good or bad.The class imbalance ratio is 19:81 and is visualized in [Fig 3]

3. CLASSIC ENSEMBLES

There are two sections namely the Classic ensembles and Ensembles for the class imbalance problem. The Classic ensembles employed in this research are briefly described in the first section, and the ECIP is described in the next section.

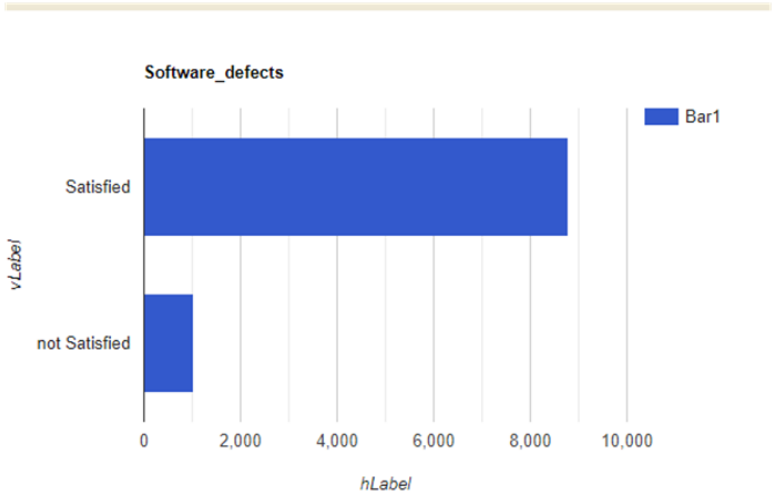

Figure 1. Distribution of defect and no defect class labels

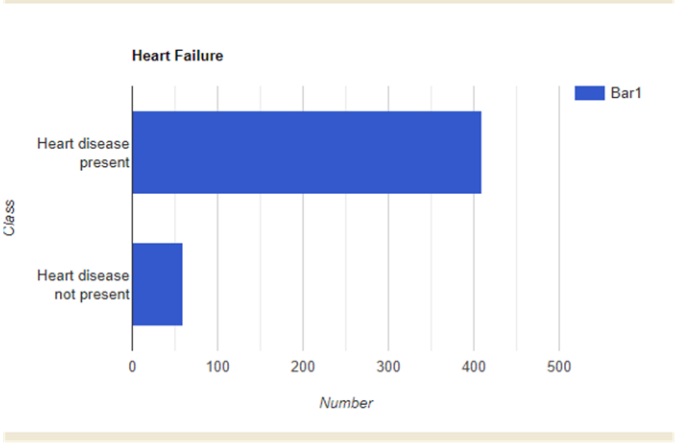

Figure 2. Distribution of disease and no disease class labels

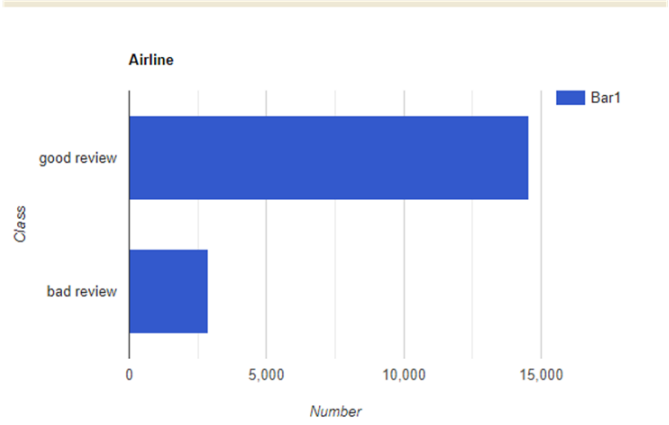

Figure 3. Distribution of good and bad review class labels

3.1. Bagging

Bootstrap Aggregation (aka Bagging) is a basic yet effective ensemble approach. The Bootstrap approach is applied to a high-variance machine learning system, such as decision trees [8], in the process of bagging. Consider the case when there are N observations and M features. A sample from observation is selected randomly with replacement(Bootstrapping). A subset of features is selected to create a model with a sample of observations and a subset of features. The feature from the subset that yields the best split on the training data is chosen. This process is repeated to produce a large number of models, each of which is trained in parallel. A forecast is made by aggregating all of the models' predictions. Bagging is the ideal approach if the single model's issue is overfitting. Boosting, on the other hand, does not prevent over-fitting; in fact, this strategy is plagued by this issue. As a result, Bagging is more successful than Boosting in this situation.

3.2. Boosting

Boosting is a set of algorithms that apply weighted averages to turn poor learners become good learners. In contrast, bagging had each model run in dependently before aggregating the outputs

in the end without giving any model preference. Boosting is all about ”collaboration.” Each model that runs determines which features will be prioritized in the following model. If the issue is that a single model performs poorly, Bagging is unlikely to produce a better bias. Boosting, on the other hand, may result in a combined model with reduced errors by maximizing the advantages and minimizing the drawbacks of a single model.

4. ECIP MODEL

The hybridization of data resampling techniques and classic ensembles is the ensemble for the class-imbalance problem. For the learning process, data resampling algorithms give balanced data distribution to ensembles. The baseline model that was used for these problems is the Decision tree classifier which runs on non-resampled data to differentiate the performance between normal models and ECIP models. The ECIP is categorized into three parts: boosting-based ensembles, bagging-based ensembles, and hybrid ensembles.

4.1. Boosting based ECIP Models

4.1.1. Adaboost

AdaBoost is a boosting technique. The technique’s main focus is on difficult-to-understand cases. The complete dataset is presented to each classifier sequentially at the start of the learning phase, with identical weights assigned to all of the instances. Following iteration, the algorithm’s main attention is on the cases that are difficult to classify, which are correctly categorized in succeeding iterations.

4.1.2. SMOTE Boost

SMOTEBoost is a blending of SMOTE and AdaBoost into a single method. With the aid of SMOTE, synthetic instances are produced in SMOTEBoost to oversample the minority class with each round of boosting. Each poor learner receives balanced material in this manner, allowing them to learn more effectively.

4.1.3. RUS Boost

RUSBoost is a combination of random undersampling and AdaBoost. It works similarly to SMOTE-Boost and MSMOTEBoost, but it ensures that each round of boosting has a balanced data distribution by removing the majority of class instances at random. RUSBoost is a simpler and quicker ensemble to deal with class-imbalance problems than SMOTEBoost and MSMOTEBoost since a lower amount of data points are provided to the classifiers during each round of boosting

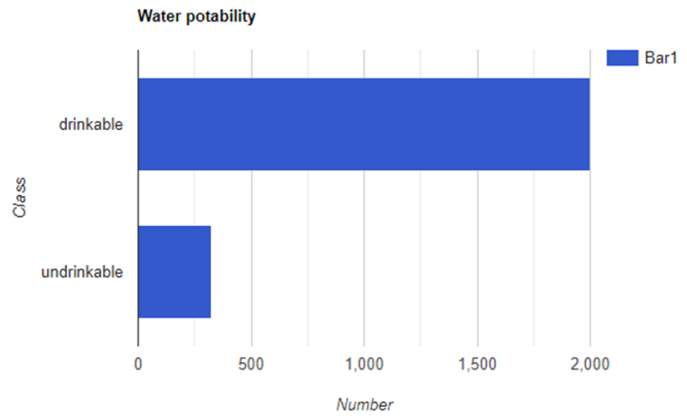

Figure 4. Distribution of drinkable and not drinkable class labels

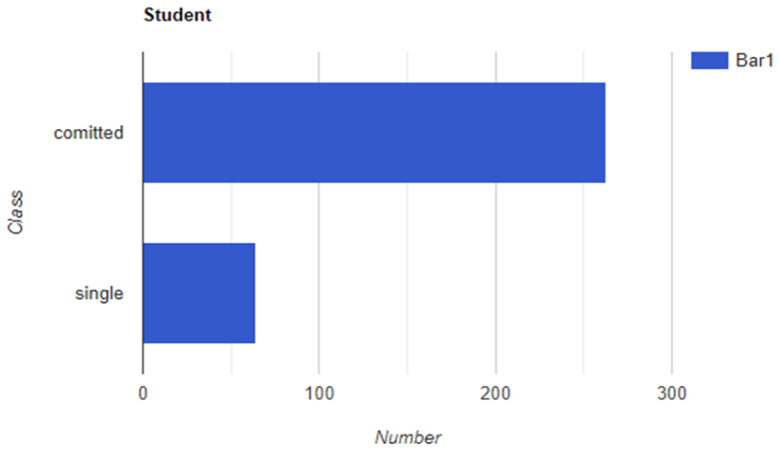

Figure 5. Distribution of in-relationship and not in relationship review class labels

4.2. Bagging based ECIP Models

4.2.1. Over Bagging

This technique is an amalgam of oversampling sampling with Bagging. In this approach, each subset of training data is generated by oversampling the instances of the minority class. The subsets of the balanced training dataset are then used to train the individual classifiers in forming the ensemble.

4.2.2. Under Bagging

This technique is a combination of undersampling and bagging. Each classifier in the ensemble is built iteratively from a subset of a balanced training dataset obtained via undersampling, which contains the majority of class instances. When an unseen instance is provided to the ensemble after the individual classifiers have been constructed, the majority vote on the individual classifiers' predictions is used to output the class label of the unseen instance.

5. RESULTS

The results of the experiments with all the models mentioned above for the software defects, airline, heart disease, water potability, and student relation-ship are provided in the tables below.

From Table 1 It can be inferred that RusBoost works with the best efficiency for the Software Defects dataset followed by SmoteBoost and EasyEnsemble.

Table 1. Software Defects

S.No	Model	g-mean	F1-score
1	Decision Tree	0.3963	0.158590
2	Bagging	0.192285	0.069665
3	AdaBoost	0.096198	0.018182
4	Easy Ensemble	0.615819	0.202381
5	RusBoost	0.618666	0.227378
6	SmoteBoost	0.615819	0.202381
7	underBagging	0.556696	0.184332
8	Over Bagging	0.213254	0.069444

From the Table 2 it is prominent that under bagging works with best efficiency for the Airline dataset followed by Over Bagging and Bagging.

Table 2. Airline

S.No	Model	g-mean	F1-score
1	Decision Tree	0.915663	0.851518
2	Bagging	0.918186	0.890930
3	AdaBoost	0.865817	0.820467
4	Easy Ensemble	0.891907	0.740586
5	RusBoost	0.906828	0.828364
6	SmoteBoost	0.891907	0.740586
7	underBagging	0.928717	0.870704
8	Over Bagging	0.921145	0.885077

From the Table 3 It can be clearly deduced that SmoteBoost is the most efficient for the Heart Disease dataset followed by Easy Ensemble and Over Bagging.

Table 3. Heart Disease

S.No	Model	g-mean	F1-score
1	Decision Tree	0.600481	0.480000
2	Bagging	0.693375	0.592593
3	AdaBoost	0.725563	0.500000
4	Easy Ensemble	0.764811	0.645161
5	RusBoost	0.762713	0.564103
6	SmoteBoost	0.764811	0.645161
7	underBagging	0.740322	0.666667
8	Over Bagging	0.745177	0.692308

From the Table 4 It can be understood that under Bagging works with best efficiency for the Water potability dataset followed by Over Bagging and RusBoost.

Table 4. Water Potability

S.No	Model	g-mean	F1-score
1	Decision Tree	0.915663	0.851518
2	Bagging	0.918186	0.890830
3	AdaBoost	0.865817	0.820467
4	Easy Ensemble	0.891907	0.740586
5	RusBoost	0.906828	0.828364
6	SmoteBoost	0.891907	0.740586
7	underBagging	0.928717	0.870704
8	Over Bagging	0.921145	0.885077

From the Table 5 It can be seen that RusBoost works best for the Student Relationship dataset followed by Under Bagging and Decision tree.

Table 5. Student Relationship

S.No	Model	g-mean	F1-score
1	Decision Tree	0.647382	0.357143
2	Bagging	0.551447	0.352941
3	AdaBoost	0.445904	0.234294
4	Easy Ensemble	0.572263	0.285714
5	RusBoost	0.709171	0.413793
6	SmoteBoost	0.572263	0.285714
7	underBagging	0.683986	0.434783
8	Over Bagging	0.450254	0.250000

5.1. Metrics and Evaluation

5.1.1. F1 Score

F1 Score [6] is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. The equation1 is provided below.

$$F1_{Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

5.1.2. G-Mean

G-Mean [5] represents the geometric mean of true positive rate and true negative rate. A good prediction model should have high g-Mean. The equation 2 is provided below

$$gMean = \sqrt{Recall * Specificity}$$

6. CONCLUSION

Class imbalance problems are a widespread concern in the current era of Big Data Collection. Implementation of efficient methods to counteract class imbalance problems is vital for progression in the machine learning domain. From our experiments it is evident that the best models for every domain consistently are RusBoost, UnderBagging followed by a little lesser performance from Overbagging, Easy ensemble, and Smote Boost. As this experiment was

conducted on datasets from multiple domains this could be a proof of concept that ECIP models will produce a better result for datasets having class imbalance problems as our best models generalize well for all types of data with various sizes.

ACKNOWLEDGEMENTS

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. Shivani Gupta, Associate Professor, School of Computer Science and Engineering (SCOPE), for her consistent encouragement and valuable guidance pleasantly offered to us throughout the project work.We are extremely grateful to Dr. Neela Narayanan, Dean of the School of Computer Science and Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support. We also take this opportunity to thank all the faculty of the School for their support and the wisdom imparted to us throughout the course. We thank our parents, family, and friends for bearing with us throughout our project and for the opportunity they provided us in undergoing this course at such a prestigious institution.

REFERENCES

[1] S. Mohan, C. Thirumalai, και G. Srivastava, 'Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques', IEEE Access, τ. 7, σσ. 81542–81554, 2019.

[2] H. P. Libbey, 'Measuring student relationships to school: Attachment, bonding, connectedness, and engagement', The Journal of school health, τ. 74, τχ. 7, σ. 274, 2004.

[3] M. An και Y. Noh, 'Airline customer satisfaction and loyalty: impact of in-flight service quality', Service Business, τ. 3, τχ. 3, σσ. 293–307, 2009.

[4] A. Y. Itah και C. E. Akpan, 'Potability of drinking water in an oil impacted community in southern Nigeria', 2005.

[5] R. P. Espíndola και N. F. F. Ebecken, 'On extending f-measure and g-mean metrics to multi-class problems', WIT Transactions on Information and Communication Technologies, τ. 35, 2005.

[6] H. Huang, H. Xu, X. Wang, και W. Silamu, 'Maximum F1-score discriminative training criterion for automatic mispronunciation detection', IEEE/ACM Transactions on Audio, Speech, and Language Processing, τ. 23, τχ. 4, σσ. 787–797, 2015.

[7] R. Malhotra και K. Lata, 'Using Ensembles for Class-Imbalance Problem to Predict Maintainability of Open Source Software', International Journal of Reliability, Quality and Safety Engineering, τ. 27, σ. 2040011, 2020.

[8] F. Pedregosa κ.ά., 'Scikit-learn: Machine Learning in Python', Journal of Machine Learning Research, τ. 12, σσ. 2825–2830, 2011.