Modular Analysis of Dataset Balancing Techniques For Binary Classification

Albert Boateng
albert.boateng@g.bracu.ac.bd

***Abstract*—This paper presents a modular analysis of dataset balancing techniques for binary classification. The study's objective is to analyze the performance or check the necessity of different balancing techniques. Several balancing techniques, such as Random Oversampling, Random Undersampling, Synthetic Minority Over-sampling Technique (SMOTE), NearMiss and Tomek Links, and their combinations are evaluated using performance metrics such as accuracy, macro F1-score, and Micro F1-Score.**

The results reveal that Balancing techniques are not really necessary and hardly impact the performance of models especially in binary classification tasks. It is hoped that the findings will aid practitioners in making informed decisions in selecting balancing techniques for their problem domain.

***Index Terms*—Imbalanced, balancing, Oversampling, Undersampling**

I. Introduction

It is very unlikely to get a very balanced real world dataset and as such Imbalanced datasets are pervasive in various classification tasks. [1] Most people in the realm of datascience claim balancing imbalanced dataset improve the performance of the models used on the dataset. However, is there any such empirical evidence to this claim or is it just a conception in the minds of data scientist? As such, We will be analyzing the performance of balancing techniques on some real world data to figure out the real truth.

Nonetheless, there are a lot of balancing techniques available out there for data scientists. These balancing techniques can be broadly classified into Oversampling Techniques and Undersampling Techniques. Oversampling aims to balance the dataset by increasing the number of instances in the minority class [2]. This is done by duplicating instances from the minority class until the number of instances in that class is equal to the number of instances in the majority class [2]. Oversampling Techniques/algorithms include Random Oversampling, SMOTE, Adaptive Synthetic Sampling (ADASYN), Borderline Smote, KMeans Smote. SMOTE, for instance, creates synthetic samples of the minority class by interpolating new instances between existing ones thereby creating synthetic samples by randomly selecting one or more of the minority class samples and computing the difference between the feature vector of that sample and that of one of its k nearest neighbors. [3] The synthetic samples are then created by adding the difference to the feature vector of the selected sample. [3]

Undersampling, on the other hand, aims to balance the dataset by reducing the number of instances in the majority class. This is done by removing instances from the majority class until the number of instances in that class is equal to the number of instances in the minority class. [4] Unsampling Techniques/algorithms include NearMiss (version 1,2,3), Tomek links and several others. NearMiss, for instance, selects instances from the majority class that are "closest" to instances from the minority class and remove them from the dataset. [5] There are three versions of NearMiss: NearMiss-1 which selects the instances from the majority class that are closest to the minority class instances by computing the Euclidean distance between each majority class instance and the average feature vector of the minority class, NearMiss-2 which selects the instances from the majority class that are farthest from the minority class instances by computing the Euclidean distance between each majority class instance and the average feature vector of the minority class, and finally NearMiss-3 which selects the instances from the majority class that are farthest from the minority class instances by computing the average distance between each majority class instance and its k nearest neighbors in the minority class. [5] Tomek Links also identifies pairs of instances, one from the majority class and one from the minority class, that are very close to each other, and remove the majority class instance from the dataset. [6]

To measure and contrast the performance of the balancing techniques, some form of metric will be required. The common metrics used in Machine Learning or Data science include Accuracy, Precision, Recall, F1-Score (Macro, Micro, Weighted). Accuracy measures the proportion of instances that were correctly classified out of the total number of instances in the dataset. [7] Precision measures the proportion of correctly predicted positive instances out of the total number of instances that were predicted as positive. [8] Recall measures the proportion of correctly predicted positive instances out of the total number of positive instances in the dataset. [9] F1-Score combines both precision and recall into a single score that provides an overall measure of the model's accuracy. [10] Macro F1- calculates the F1 score for each class separately and then takes the average of all the F1 scores to give a final score. [11] Micro F1-score calculates by treating all instances as a single class and then calculating the precision and recall for that class. [12] Weighted F1-Score calculates by taking a weighted average of the F1 score for each class where the weight is proportional to the number of instances in each class. [13] Only Accuracy, Macro F1-Score, and Micro F1-Score will be used in our analysis.

II. Datasets

Twenty different varieties of imbalanced datasets, all obtained from Kaggle, were used in our analysis. The first dataset (Dataset 1) is the SEER Breast Cancer dataset which has 4,024 observations/rows and 16 variables/columns. Out of the 4,024 values in the target variable ('Status'), 3,408 of them were 'Alive' and 616 were 'Dead'. The second dataset (Dataset 2) is a vehicle insurance datasets which has 382,154 observations and 11 variables. Here, out of the 382,154 values in the target variable ('Response'), 319,553 of them are 0's and 62,601 of them are 1's. The third dataset (Dataset 3) is a credit card fraud detection dataset which contain 25,134 observations and 20 targets. Out of the 25,134 values in the target variable ('TARGET'), 24,712 are 0's and 422 are 1's. The fourth dataset (Dataset 4) is a Sample Telco Customer Churn dataset which contain 7,043 observations and 24 targets. Out of the 7,043 values in the target variable ('Churn'), 5,174 are 'No' and 1,869 are 'Yes'. The fifth dataset (Dataset 5) is a Vehicle Insurance Fraud Detection dataset which contain 15,420 observations and 33 targets. Out of the 15,420 values in the target variable ('FraudFound'), 14,497 are 'No' and 923 are 'Yes'.

The sixth dataset (Dataset 6) is a Permanent Neonatal Diabetes Mellitus (PNDM) Prediction dataset which contain 100,000 observations and 8 targets. Out of the 100,000 values in the target variable ('PNDM'), 95,178 are 'No' and 4,822 are 'Yes'. The seventh dataset (Dataset 7) is a Marketing Analysis dataset which contain 7,414 observations and 22 targets. Out of the 7,414 values in the target variable ('responded'), 6,574 are 'No' and 840 are 'Yes'. The eighth dataset (Dataset 8) is Rain in Australia dataset which contain 56,420 observations and 23 targets. Out of the 56,420 values in the target variable ('RainTomorrow'), 43,993 are 'No' and 12,427 are 'Yes'. The ninth dataset (Dataset 9) is an analysis of depression dataset which contain 1,429 observations and 23 targets. Out of the 1,429 values in the target variable ('depressed'), 1,191 are 0's and 238 are 1's. The tenth dataset (Dataset 10) is a Banking Marketing Targets dataset which contains 45,211 observations and 17 targets. Out of the 45,211 values in the target variable ('responded'), 39,922 are 'No' and 5,289 are 'Yes'.

The eleventh dataset (Dataset 11) is a Crack the model from credit card fraudster dataset which contains 1,000,000 observations and 8 targets. Out of the 1,000,000 values in the target variable ('fraud'), 912,597 are 0's and 87,403 are 1's. The twelve dataset (Dataset 12) is a Banking Marketing Targets dataset which contains 30,000 observations and 25 targets. Out of the 30,000 values in the target variable ('default'), 6,636 are 'Y' and 23,364 are 'N'. The thirteenth dataset (Dataset 13) is a Monkeypox detecting dataset which contains 25,000 observations and 11 targets. Out of the 25000 values in the target variable ('MonkeyPox'), 15,909 are 'Positive' and 9,091 are 'Negative'. The fourteenth dataset (Dataset 14) is a Heart Failure Prediction dataset which contains 368 observations and 60 targets. Out of the 368 values in the target variable ('Mortality'), 288 are 0's and 80 are 1's. The fifteenth dataset (Dataset 15) is a Hotel customers' booking details dataset which contains 36,275 observations and 19 targets. Out of the 36,275 values in the target variable ('booking_status'), 24,390 are 'Not_Canceled' and 11,885 are 'Canceled'.

The sixteenth dataset (Dataset 16) is an HR-dataset of ScaleneWorks which contains 8,995 observations and 18 targets. Out of the 8,995 values in the target variable ('Status'), 7,313 are 'Joined' and 1,682 are 'Not Joined'. The seventeenth dataset (Dataset 17) is an Oil Spill Classification dataset which contains 937 observations and 50 targets. Out of the 937 values in the target variable ('target'), 896 are 0's and 41 are 1's. The eighteenth dataset (Dataset 18) is a Term Deposit Prediction dataset which contains 31,647 observations and 18 targets. Out of the 31,647 values in the target variable ('subscribed'), 27,932 are 'no' and 3,715 are 'yes'. The nineteenth dataset (Dataset 19) is a Health Insurance Lead Prediction dataset which contains 23,548 observations and 14 targets. Out of the 23,548 values in the target variable ('Response'), 17,848 are 0's and 5,700 are 1's. The twentieth dataset (Dataset 20) is a Diabetes Binary Classification dataset which contains 768 observations and 9 targets. Out of the 768 values in the target variable ('Class variable (0 or 1)'), 500 are 0's and 268 are 1's.

III. Methodology

All the 20 datasets were individually preprocessed accordingly and converted to the appropriate numerical form for training. Next, the preprocessed datasets were then splitted into training set and test set in the ratio 0.7 to 0.3. The various sampling techniques, Random Over-sample, Random Under-sample, SMOTE, TomekLinks, NearMiss, Random Over-sample + TomekLinks, Random Over-sample + NearMiss, SMOTE + TomekLinks, SMOTE + NearMiss, were then respectively applied to the training set to obtain 10 different sets of training data (including the Actual unsampled training set).

The ten different training sets were then passed through five different Machine Learning models/Algorithms: Decision Tree, Naive Bayes, K-Nearest Neighbor (KNN), AdaBoost Classifier, and Logistic Regression. Models predictions where then made and comparisons where done with the testing set to compare and contrast the performance of each balancing techniques. Specifically, Accuracy, Macro F1-Score, and Micro F1-Score of the balancing techniques for each model were obtained. The mean of the performances of the models for each balancing technique where then collected and plotted on a Bar graph for visualization.

IV. Results and Analysis

The metric results obtained from all the datasets are shown below (Fig. 1. to Fig. 20). The Unsampled data (Actual data) and Tomek Links had the highest performance (accuracy and F1-Score across almost all the datasets. NearMiss algorthm had the worst performance across all the 20 datasets. Random Over- sample + TomekLinks and Random Over-sample + NearMiss had equivalent performance. Also, SMOTE +

TomekLinks and SMOTE + NearMiss had equivalent performance.

Analyzing the results further, one would realise that the balancing algorithm(s) which performed better than the unsampled dataset preformed with a very small performance difference. Also, the unsampled dataset performed far far better than all the balancing algorithms, except of course Tomek Link. This implies that the balancing techniques actually had no significant implications on the performance. In some cases, it even worsened the performance of models.

Because Tomek Link's had comparable performance to the Unsampled dataset, we decided to look into its outputs but realised that it doesn't actually balance the dataset as we thought. It rather just add some few minority data points or remove some few majority data points base on your parameter selection. As such, it produces almost the same training dataset as the Unsampled dataset. It performed little better in some of cases due to those few data points it added or removed. Nevertheless, if Tomek Links doesnt equally balance datasets, why then is it classified a balancing technique? This is another question for another day. However, the fact presented here clearly oppose the assumption that balancing dataset improve performance. We can cautiously even state that it rather worsen the performance as highlighted in the results.

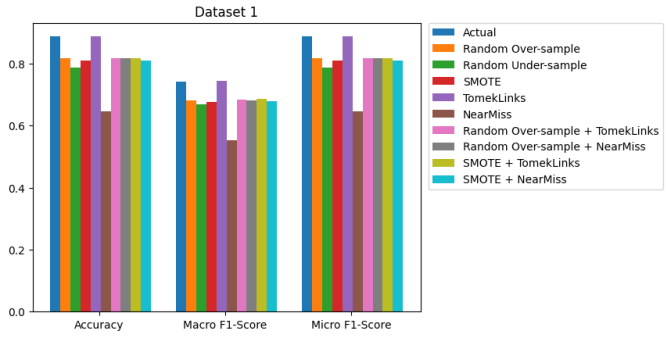

Fig. 1. Metric Results from Dataset1

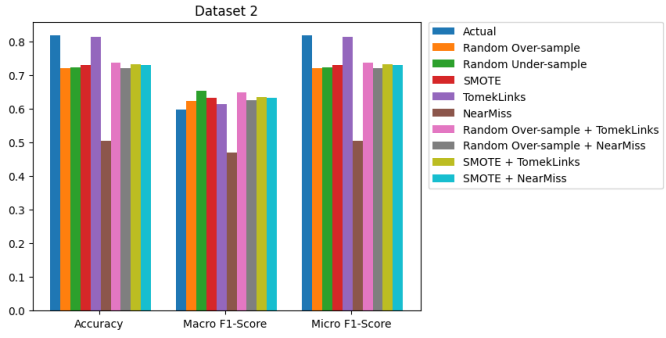

Fig. 2. Metric Results from Dataset2

V. Conclusion

We know that the number of datasets used in this analysis is not large enough to make such a broad generalization and also our analysis was based on only binary classifications. How-

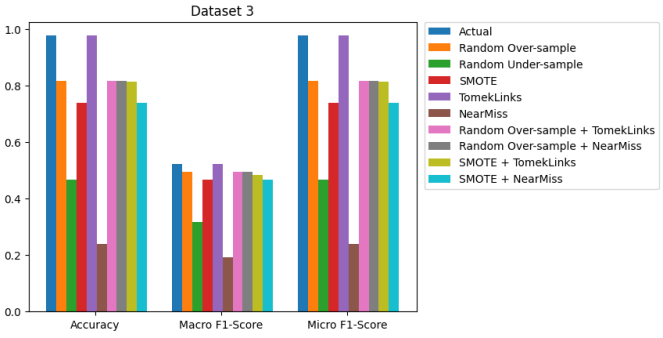

Fig. 3. Metric Results from Dataset3

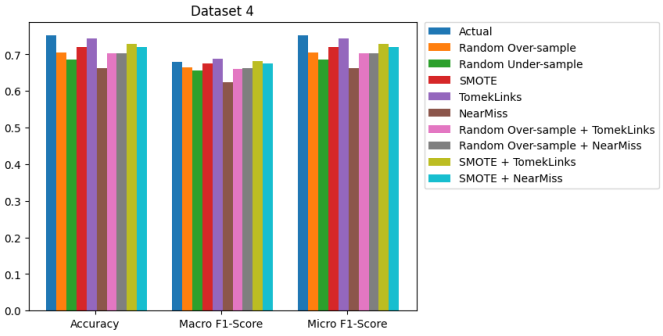

Fig. 4. Metric Results from Dataset4

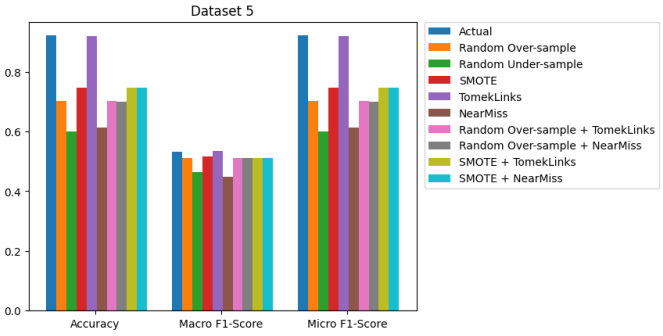

Fig. 5. Metric Results from Dataset5

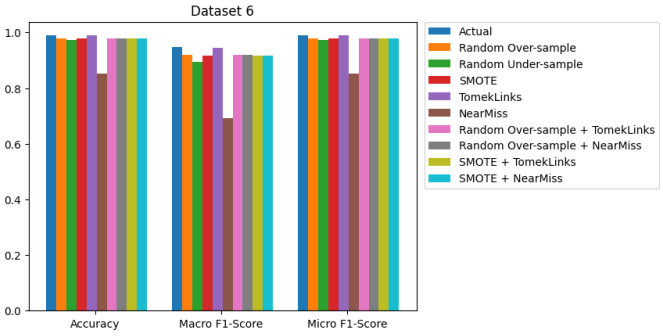

Fig. 6. Metric Results from Dataset6

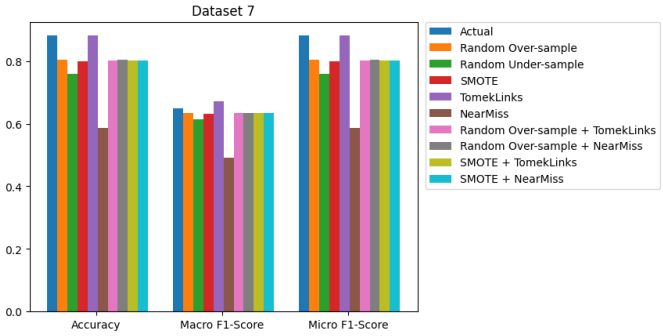

Fig. 7. Metric Results from Dataset7

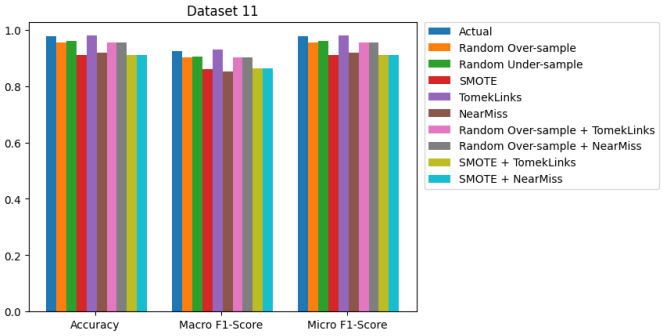

Fig. 11. Metric Results from Dataset11

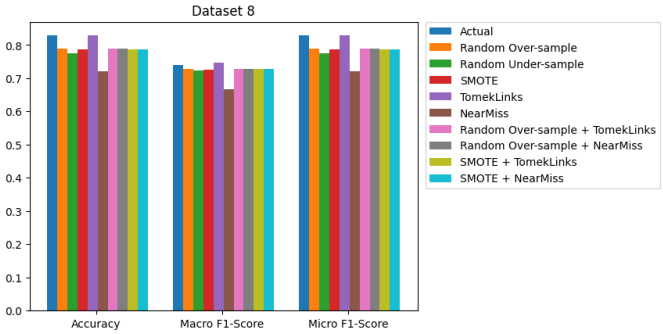

Fig. 8. Metric Results from Dataset8

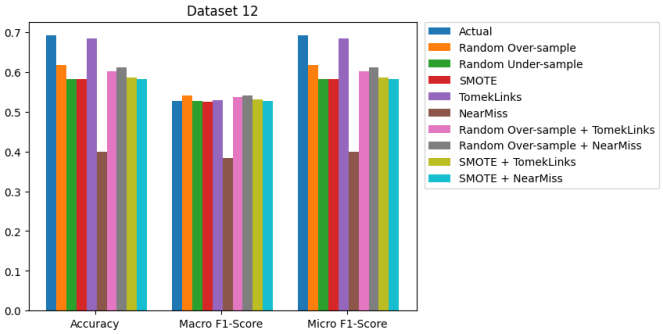

Fig. 12. Metric Results from Dataset12

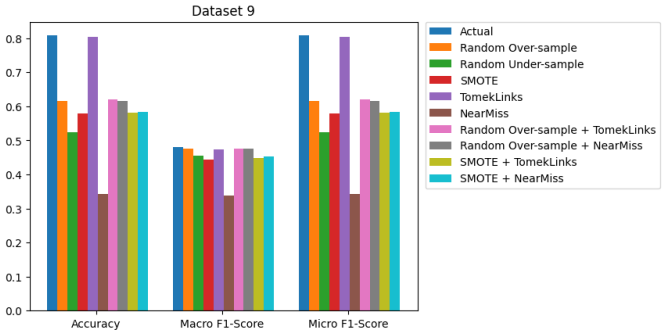

Fig. 9. Metric Results from Dataset9

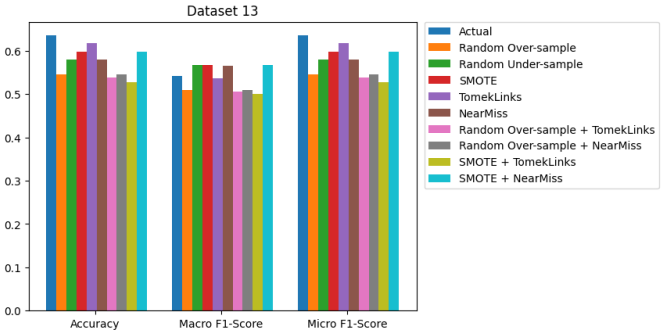

Fig. 13. Metric Results from Dataset13

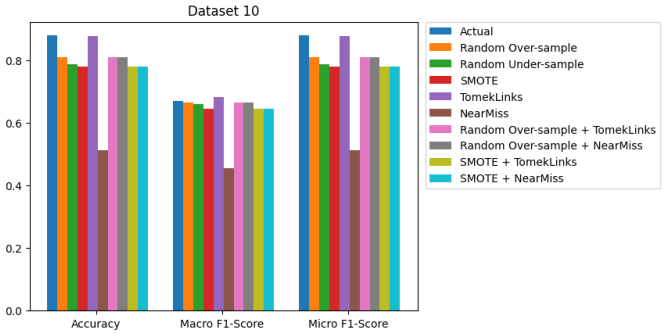

Fig. 10. Metric Results from Dataset10

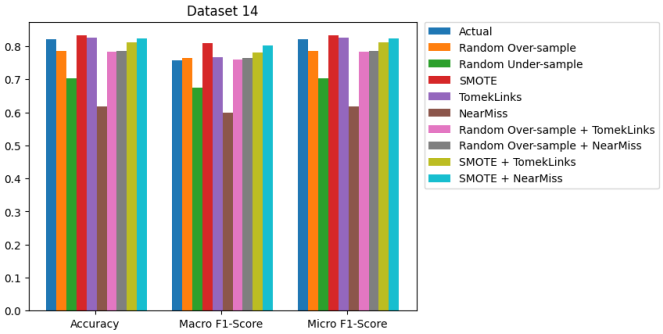

Fig. 14. Metric Results from Dataset14

ever, this is just to ignite further research into these balancing techniques. We are currently restricted by resources and would

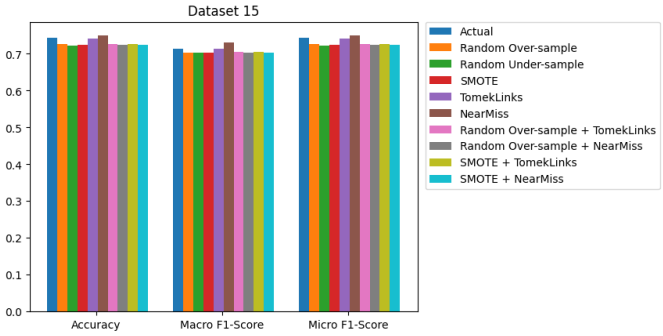

Fig. 15. Metric Results from Dataset15

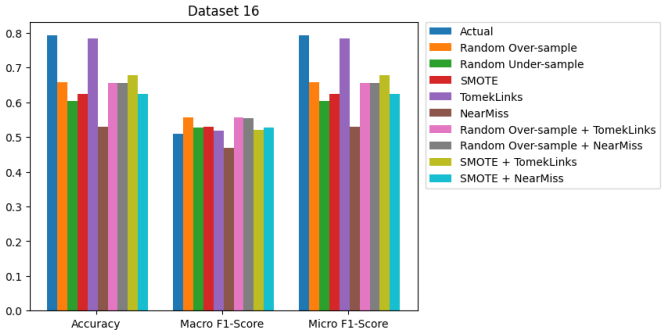

Fig. 16. Metric Results from Dataset16

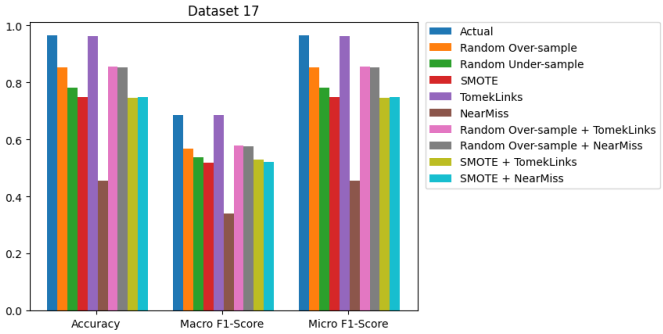

Fig. 17. Metric Results from Dataset17

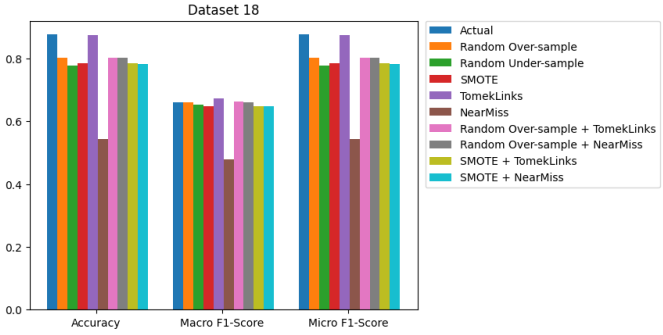

Fig. 18. Metric Results from Dataset18

however expand this analysis to huge amount of datasets once

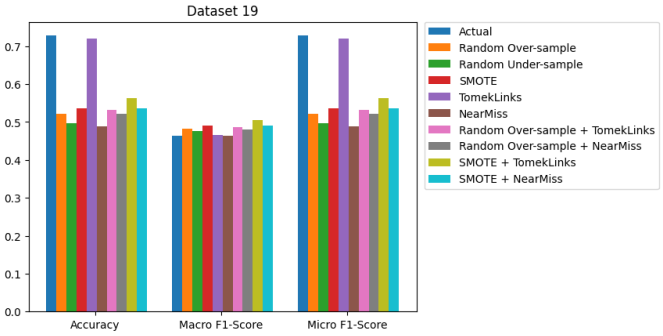

Fig. 19. Metric Results from Dataset19

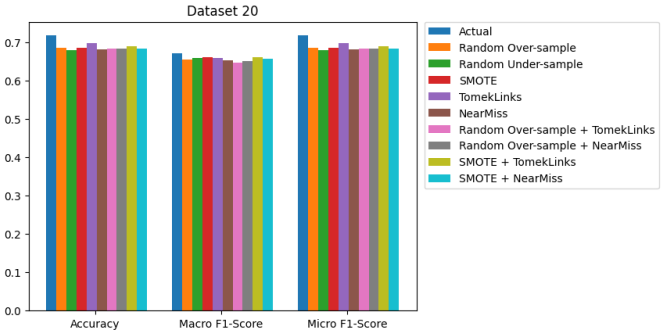

Fig. 20. Metric Results from Dataset20

we get the computational capabilities. Nonetheless, we entreat all practitioners to make carefully evaluation and analysis of these balancing techniques before deploying them into their work.

ACKNOWLEDGMENT

We acknowledge the original sources of all the datasets we used in our analysis. Due to the quantity of datasets used, we didn't get the space to cite or highlight the actual sources of all the datasets. But we do cherish the time and energy used to obtain these data.

REFERENCES

[1] Kubat, M. (2000). "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection". In Proceedings of the Fourteenth International Conference on Machine Learning

[2] J. Zhai, X. Wu, and Y. Li, "A novel random oversampling method for class imbalance learning," Neurocomputing, vol. 237, pp. 220-229, 2017.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.

[4] T. Wang, J. Zhang, L. Zhu, and X. Chen, "Random undersampling based on clustering for imbalanced data classification," Knowledge-Based Systems, vol. 192, pp. 105413, 2020.

[5] I. 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, vol. 126, pp. 1-12, 2003.

[6] I. Tomek, "Two modifications of CNN," IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, no. 11, pp. 769-772, 1976.

[7] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," Radiology, vol. 143, no. 1, pp. 29-36, 1982.

[8] L. Davis, "A survey of techniques for precise probabilistic inference," Computational intelligence, vol. 14, no. 4, pp. 531-595, 1998.

[9] C. E. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.

[10] A. R. Azzalini and A. W. Bowman, "A look at some data on the old faithful geyser," Applied Statistics, vol. 47, no. 5, pp. 591-598, 1998.

[11] Y. Lei, L. Wei, H. Lu, and J. Li, "Large-scale multi-label learning with missing labels," In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 311-318, 2019.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of machine learning research, vol. 12, pp. 2825-2830, 2011.

[13] J. Chen, C. Zhu, and D. Ye, "Learning to rank using an ensemble of lambda-gradient models," In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 275-283, 2008.