2020 11th International Conference on Information and Communication Systems (ICICS) 978-1-7281-6227-0/20/$31.00 ©2020 IEEE 10.1109/ICICS49469.2020.239556

Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results

Roweida Mohammed, Jumanah Rawashdeh and Malak Abdullah
Jordan University of Science and Technology
Irbid, Jordan
roweida.221@gmail.com, jsrawashdeh17@cit.just.edu.jo, mabdullah@just.edu.jo

***Abstract*—Data imbalance in Machine Learning refers to an unequal distribution of classes within a dataset. This issue is encountered mostly in classification tasks in which the distribution of classes or labels in a given dataset is not uniform. The straightforward method to solve this problem is the resampling method by adding records to the minority class or deleting ones from the majority class. In this paper, we have experimented with the two resampling widely adopted techniques: oversampling and undersampling. In order to explore both techniques, we have chosen a public imbalanced dataset from kaggle website *Santander Customer Transaction Prediction* and have applied a group of well-known machine learning algorithms with different hyperparamters that give best results for both resampling techniques. One of the key findings of this paper is noticing that oversampling performs better than undersampling for different classifiers and obtains higher scores in different evaluation metrics.**

***Index Terms*—Undersampling, Oversampling, Class Imbalance, Machine Learning, SVM, Random Forest, Naive Bayes, Recall, Precision, Accuracy**

I. Introduction

In machine learning and statistics, classification is defined as training a system with labeled dataset to identify a new unseen dataset to which class it belongs. Recently, there is enormous growth in data and, unfortunately, there is lack of quality labeled data. Various traditional machine learning methods assumed that the target classes have the same distribution. However, this assumption is not correct in several applications, for example weather forecast [1], diagnosis of illnesses [2], finding fraud [3], as nearly most of the instances are labeled with one class, while few instances are labeled as the other class. For this reason, the models lean more to the majority class and eliminate the minority class. This reflects on the models performance as these models will perform poorly when the datasets are imbalanced. This is called class imbalance problem. Thus, in such situation, although a good accuracy can be gained, however, we don't gain good enough scores in other evaluation metrics, such as precision, recall, F1-score [4] and ROC score.

Recently, there is a great interest in class imbalance issue. Several researchers consider it a challenging issue that needs more attention to resolve [5] [6]. One of the common approaches was to use resampling techniques to make the dataset balanced. Resampling techniques can be applied either by undersampling or oversampling the dataset. Undersampling is the process of decreasing the amount of majority target instances or samples. Some common undersampling methods contain tomeks' links [7], cluster centroids [8] and other methods. Oversampling can be performed by increasing the amount of minority class instances or samples with producing new instances or repeating some instances. An example of oversampling methods is Borderline-SMOTE [9]. Figure 1 shows the difference between the two techniques: oversampling and undersampling.

In this work, the imbalanced dataset of 'Santander Customer Transaction Prediction' from a Kaggle competitions (released in Feb, 2019) [1] has been used with different machine learning models to experiment oversampling and undersampling techniques and apply a full comparison with different evaluation metrics. Our code for this experiment can be found on github [10]. The results show that oversampling has better scores than the undersampling methods for different machine learning classifier models.

This paper is organized as follows: related work is shown in section II. Section III describes the dataset that has been used in this article. Our methodology and evaluation metrics are presented in section IV. Experiments and results are introduced in section V. Finally, the conclusion of the paper is provided in section VI.

II. Related work

The imbalance data challenge has attracted growing attention of researchers, recently. Authors in [11] proposed a famous method for undersampling. It works by eliminating the data points where target class does not equal the majority of its KNN. In [12], authors discussed several problems related to learning with class scatterings skewed. For example, the connection between class scatterings and price sensitive knowledge, and the boundaries of error frequency and accuracy to measure the act of models. In [13], the authors proposed a review for the most commonly used methods learning from imbalanced classes. They claimed that the bad performance of the models created by the typical machine

[1] https://www.kaggle.com/c/santander-customer-transaction-prediction

Fig. 1: Differences between undersampling and oversampling

learning methods on imbalanced classes is mostly due three main issues: error costs, class scattering, and accuracy.

Authors in [14] suggested the resampling methods due to the difficulty of identifying the minority target. They applied a new resampling method by which equally oversampling of infrequent positives and undersampling of the non-infected majority depending on synthetic circumstances created by class-specific sub-clustering. They stated that their new re-sampling technique achieved better results than traditional random resampling. In [15], authors applied three dissimilar methods to an advertising dataset. Logistic regression, Chi-squared automatic interaction detection, and neural network. The performance of three methods was created by the means of accurac, AUC, and precision. They compared several different imbalance datasets produced from the real dataset. They stated that precision is a good measure for imbalanced dataset.

An addition method had been introduced in [8] in which it uses k-means grouping to balance the imbalanced instances by decreasing the amount of majority instances. Also, Authors in [16] applied an undersampling method to remove information points from majority instances constructed on their spaces between each other.

For our work it will be different than the previous research because we will be using the dataset of 'Santander Customer Transaction Prediction' from a Kaggle competitions (released in Feb,2019) in order to compare between oversampling and undersampling methods.

III. Dataset

The competition from Kaggle website, "Santander Customer Transaction Prediction", is a binary classification challenge in which a dataset with numeric data fields had been provided to solve a problem. The challenge is to predict and identify which customers will make a specific transaction in the future regardless of the amount of money transacted. Knowing that the dataset is imbalanced, we used this data to tackle and review the imbalance data problem. The competition posted two datasets, training and testing datasets. In general, the whole dataset contains 202 features and 200,000 entries. The datasets has no missing values. Figure 2 shows the distribution of target column (0: refers to the number of customers that will not make the transaction, and 1: refers to the customers that will make the transaction successfully). From this figure, we can easily notice that the dataset is imbalanced dataset.

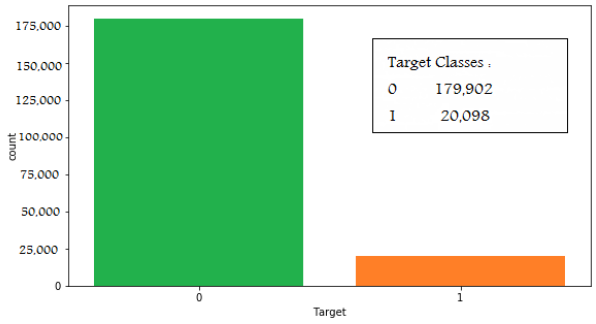

Fig. 2: Distribution of target classes

IV. Methodology

In order to give a comprehensive view of the unbalanced data problem, we have started with exploring the dataset. After downloading the training dataset from Kaggle, we have split the data into train and target dataset. Then, we scaled the dataset. Moreover, we have ranked the features and selected the important ones for our experiment. Therefore, the underlying frequency distribution of the features has been studied, and the correlation matrix between the features has been calculated. As a result, we have found that there is a small correlation between features, that means the features are mostly independent from each other. For this reason, a feature selection technique has been applied in order to select the most important features and drop the rest. Figure 3 illustrates the distribution of some features from the dataset.

After exploring the given dataset and preparing it to become compatible with machine learning algorithms, we used two resampling techniques, which depends on changing the class distribution. Also, we studied different classification models

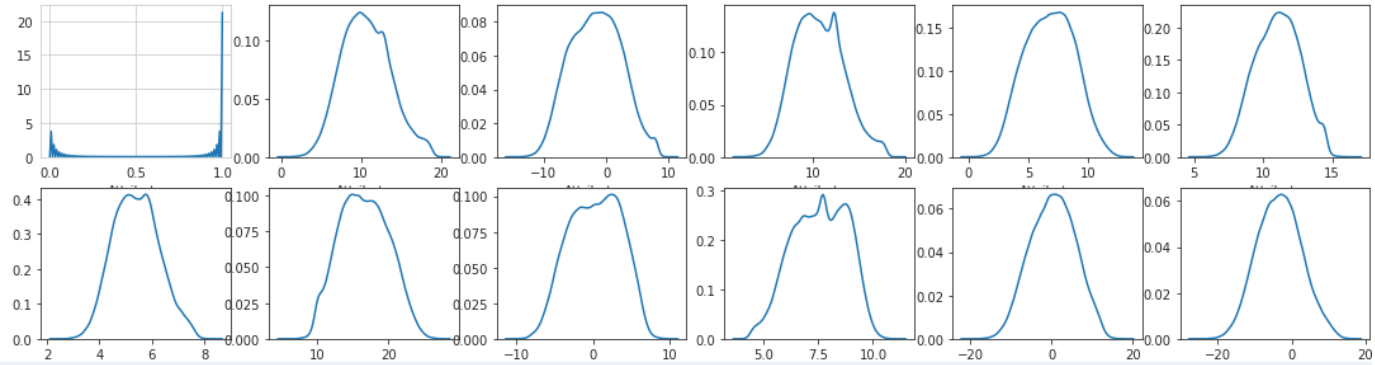

Fig. 3: A small sample to show features distribution

for this experoment. Table I shows the different classifiers that are used for both resampling techniques. To compare between classifiers, we have used different evaluation metrics (Accuracy, Precision, Recall, F1-Score [17], and ROC [18]) (note that higher is better).

Abbreviation	Machine Learning classifier models
SVM(Linear)	Support Vector Machine with Linear kernel
SVM(Poly)	Support Vector Machine with Poly kernel
SVM(RBF)	Support Vector Machine with RBF kernel
NB	Gaussian Naive Bayes
LR	Logistic regression
DT1	Decision tree
DT2	Decision tree
DT3	Decision tree
RF	Random Forest
GB	Gradient Boosting Classifier
BC(NB)	Bagging Classifier with NB
BC(DT)	Bagging Classifier with DT
AB	AdaBoosting [19]
VE	Voting-Ensembling NB, LR, DT depth=18, and RF

TABLE I: Different classifier models

We should mention that the most common evaluation metric for classic models is the accuracy metric [20] [21] [22]. However, accuracy is not a favorable measurement when dealing with imbalanced datasets [21] [22] [23]. As many experts have detected that for much skewed target distributions, the recall of the minority target is sometimes 0, which means no classification instructions had been generated for minority target. From information retrieval field, we can use the terms in which the minority target has worse precision and recall compared to the majority target. Knowing that accuracy seats extra weight on the majority target compared to minority target, this makes it hard for a classifier to accomplish well on the minority target. For this purpose, extra metrics are upcoming into general use.

In latest years, many new metrics for imbalanced datasets were proposed from other fields. Some of these metrics are recall and positive predictive, ROC and AUC [18], F-measure and other metrics. For imbalance problematic, F-measure is a popular evaluation metrics [17]. It is a mixture of positive predictive and recall. It has a high value when both positive predictive and recall are high. Possibly, the best common metric to measure general classification performance is ROC [18].

In our work, we used scikit-learn [24], numpy [25], and pandas [8] packages to implement these models and for data adaptation.

V. Experiments and Results

A. Oversampling Minority Class

For our first experiment, we used the oversampling technique. A non heuristic algorithm is known as random oversampling. Its main objective is to balance class spreading through the random repetition of minority target instances. Figure 4 shows how the class target is distributed after using this method on our dataset and it equals to 120,000.

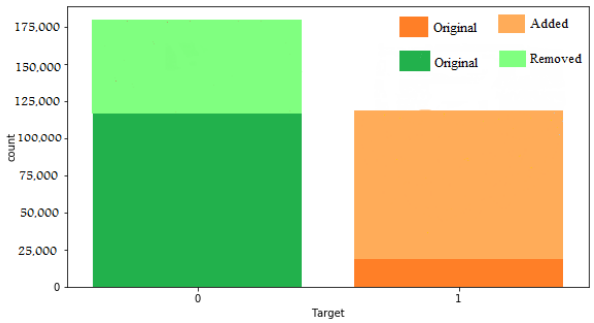

Fig. 4: Class target distribution after oversampling method

However, this technique has two limitations. First, it will rise the probability of over-fitting, as it creates the same reproductions of the minority class instances [26]. Second, it makes learning procedure more time overwhelming especially if the original dataset is now equally huge, but imbalanced; the same as our dataset. It is good to use this method when you don't have a lot of data to work with.

We used different classification models to make the prediction with random oversampling technique. We have experimented different hyperparamters for each model. Table II shows the best hyperparamters for the classifier models with using oversampling technique.

Best Parameters for Oversampling Technique	
Classifiers	**Hyper-parameters**
SVM (Linear Kernel)	C = 1.0 Max iteration = 10 Learning rate = 0.0001
SVM (Poly Kernel)	C = 0.1 Degree = 1 Gamma = 0.1
SVM (RBF)	C = 0.1 Gamma = 0.2
Naïve Bayes	Default hyperparameters
Logistic Regression	Default hyperparameters
Decision Tree 1	Default hyperparameters
Decision Tree 2	Criterion = entropy Max depth = 20
Decision Tree 3	Criterion = gini Max depth = 18
Random Forest	Criterion = entropy Max depth = 25 Max features = log2 N_estimators = 150
Gradient Boosting	Learning rate = 0.1 Max depth = 10 N_estimators = 60 Subsample = 1.0
Bagging with Naïve Bayes	N_estimators = 10 oob score = False
Bagging with decision trees	Criterion = gini
Ada Boosting	Algorithm = SAMME.R Learning rate = 0.1 N_estimators = 50

TABLE II: Classifier models hyperparamters with oversampling

Table III shows the evaluation metrics for all the classifiers with the mentioned hyperparametrs. We can see that Random Forest has the highest score between all of evaluation metrics and performed better than the other classifiers. Furthermore, Figure 5 presents the charts of the results of different evaluation metrics.

Models	Accuracy	Precision	Recall	F1	ROC-Curve
SVM(Linear)	0.73	0.74	0.73	0.73	0.73
SVM(Poly)	0.73	0.74	0.72	0.73	0.73
SVM(RBF)	0.80	1.00	0.60	0.75	0.80
NB	0.76	0.77	0.75	0.76	0.76
LR	0.73	0.74	0.73	0.73	0.73
DT1	0.94	0.90	0.99	0.94	0.94
DT2	0.87	0.80	0.91	0.87	0.87
DT3	0.84	0.83	0.86	0.84	0.84
RF	**0.998**	**0.999**	**0.997**	**0.998**	**0.998**
GB	0.93	0.92	0.94	0.93	0.93
BC(NB)	0.76	0.77	0.75	0.76	0.76
BC(DT)	0.98	0.97	0.998	0.988	0.987
AB	0.68	0.70	0.64	0.67	0.68
VE	0.93	0.93	0.93	0.93	0.93

TABLE III: Evaluation metrics for the classifiers with Oversampling

B. Undersampling Majority Class

In our second experiment, we used the undersampling technique. The best simple undersampling algorithm is random undersampling [17]. It is a non-heuristic algorithm which try to balance target distributions over eliminating randomly from majority class instances. By this operation, it may remove possibly valuable data that can be essential for classifier models, but it is useful when you have a lot of data. Figure 6 shows the target class after using this method on our dataset and it equals to 14,000.

Various heuristic undersampling algorithms have been presented or announced from cleaning the data in latest years [16]

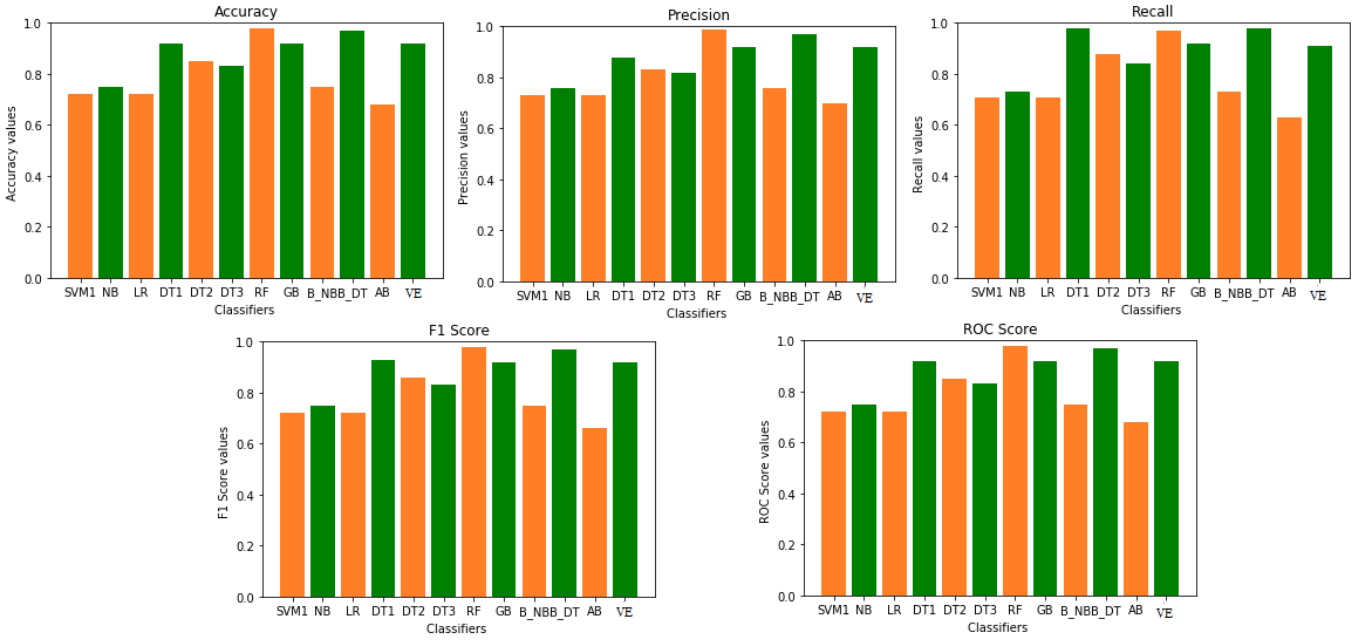

Fig. 5: Charts of different evaluation metrics for oversampling

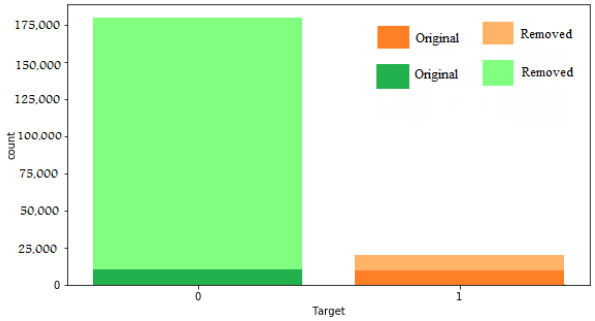

Fig. 6: Class target distribution after undersampling method

[27] [28] [29] [30]. They are created on two dissimilar noise model theories. Some researchers think that the instances, which are close to classification margin of two classes, are known as noise. On the other hand, some researcher deliberates instances through more neighbors of various labels are known as noise. We used the same classification models of oversampling experiment to make the prediction with undersampling technique. We have experimented different hyperparamters for each model. Table IV shows the best hyperparamters for the classifier models with using undersampling technique.

When we used the random undersampling technique, the score for the classifier models was poor when compared to the random oversampling technique. Table V offers the evaluation metrics for all the classifiers for the random undersampling technique. We can notice that some classifiers have the same scores as in the oversampling method or got worse in some other classifiers. Naive Bayes (NB) in undersampling method got the higher score compared to the other classifiers. Figure 7 presents the charts for the evaluation metrics for the classifier models.

VI. Conclusion

In this paper, we presented two techniques to handle the problem of class imbalance and applied them to different machine learning classification models. We have used the dataset provided by the competition from Kaggle website, ”Santander Customer Transaction Prediction”. This data was released in 2019 and it is a binary classification challenge to predict and identify which customers will make a specific transaction in the future regardless of the amount of money transacted. Knowing that the dataset is imbalanced, we used this data to tackle and review the imbalance data problem. We tried the oversampling technique for the dataset and measured the classifiers with different evaluation metrics; as well for the other technique, undersampling. We noticed how oversampling perform better than undersampling for different classifiers and get higher scores in different evaluation metrics. For future work, we are planning to apply different deep learning techniques with both resampling techniques to compare between both.

Best parameters for Undersampling Technique	
Classifiers	**Hyper-parameters**
SVM (Linear Kernel)	Default hyperparameters
SVM (Poly Kernel)	C = 0.1 Degree = 3 Gamma = 0.1
SVM (RBF)	C = 0.001 Gamma = 0.5
Naïve Bayes	Default hyperparameters
Logistic Regression	Default hyperparameters
Decision Tree 1	Default hyperparameters
Decision Tree 2	Criterion = entropy Max depth = 20
Decision Tree 3	Max depth = 18
Random Forest	Criterion = gini Max depth = 25 Max features = log2 N_estimators = 150
Gradient Boosting	Learning rate = 0.1 Max depth = 3 N_estimators = 60 Subsample = 0.5
Bagging with Naïve Bayes	N_estimators = 10 oob score = False
Bagging with decision trees	Criterion = entropy Max depth = 20
Ada Boosting	Algorithm = SAMME.R Learning rate = 0.1 N_estimators = 50

TABLE IV: classifier models hyperparamters with undersampling

Models	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC-Curve**
SVM(Linear)	0.74	0.74	0.73	0.74	0.74
SVM(Poly)	0.74	0.78	0.67	0.72	0.74
SVM(RBF)	0.50	0.0	0.0	0.0	0.50
NB	**0.77**	**0.77**	**0.75**	**0.76**	**0.77**
LR	0.74	0.74	0.73	0.74	0.74
DT1	0.59	0.59	0.59	0.59	0.59
DT2	0.62	0.64	0.57	0.60	0.62
DT3	0.60	0.62	0.55	0.58	0.60
RF	0.75	0.75	0.76	0.75	0.75
GB	0.73	0.75	0.70	0.73	0.73
BC(NB)	0.77	0.78	0.75	0.76	0.76
BC(DT)	0.67	0.68	0.63	0.66	0.67
AB	0.68	0.70	0.63	0.66	0.68
VE	0.75	0.76	0.74	0.75	0.75

TABLE V: Evaluation metrics for the classifiers with undersampling

References

[1] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, “Prediction of weather-induced airline delays based on machine learning algorithms,” in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, pp. 1–6.

[2] B. Krawczyk, M. Galar, Ł. Jeleń, and F. Herrera, “Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy,” *Applied Soft Computing*, vol. 38, pp. 714–726, 2016.

[3] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, “Effective detection of sophisticated online banking fraud on extremely imbalanced data,” *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2013.

[4] C. J. Van Rijsbergen, *The geometry of information retrieval*. Cambridge University Press, 2004.

[5] Y. Sun, A. K. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

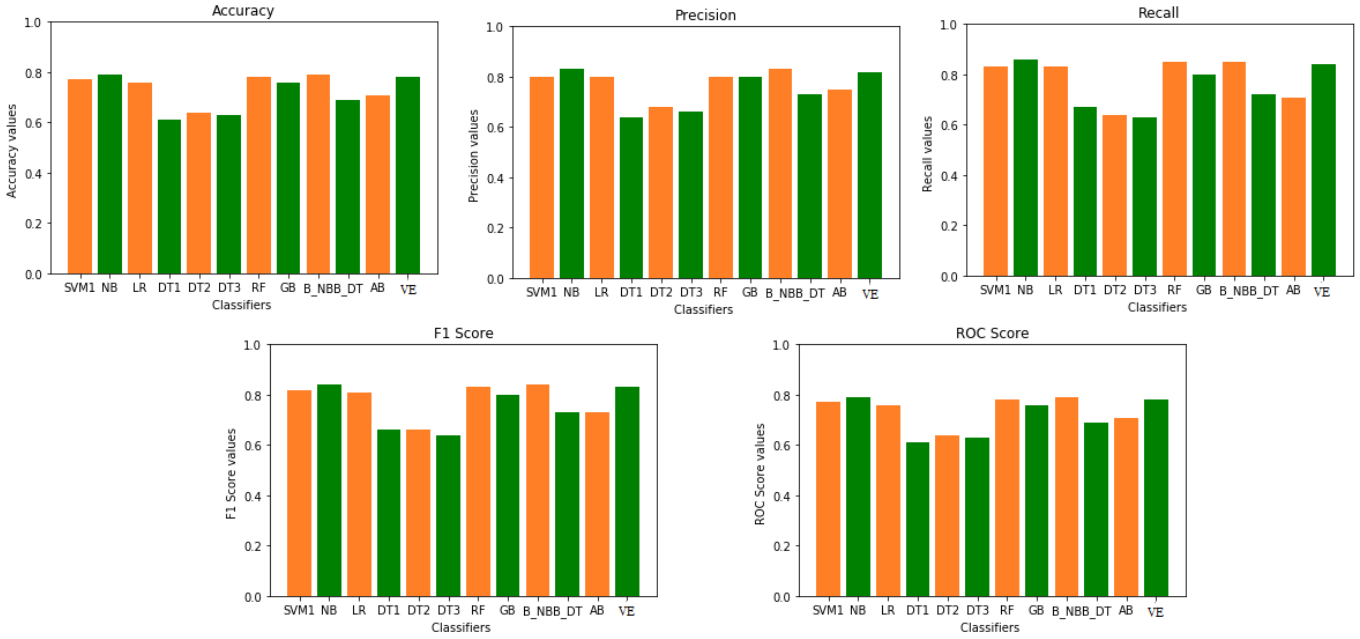

Fig. 7: Charts of different evaluation metrics for undersampling

[6] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[7] I. Tomek, "A generalization of the k-nn rule," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 121–126, 1976.

[8] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[9] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[10] https://github.com/Roweida-Mohammed/Code_For_Santander_Customer_Transaction_Prediction.

[11] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 408–421, 1972.

[12] M. C. Monard and G. E. Batista, "Learmng with skewed class distrihutions," *Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC*, vol. 85, no. 2002, p. 173, 2002.

[13] S. Visa and A. Ralescu, "Issues in mining imbalanced data sets-a review paper," in *Proceedings of the sixteen midwest artificial intelligence and cognitive science conference*, vol. 2005. sn, 2005, pp. 67–73.

[14] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from imbalanced data in surveillance of nosocomial infection," *Artificial intelligence in medicine*, vol. 37, no. 1, pp. 7–18, 2006.

[15] E. Duman, Y. Ekinci, and A. Tanrıverdi, "Comparing alternative classifiers for database marketing: The case of imbalanced datasets," *Expert Systems with Applications*, vol. 39, no. 1, pp. 48–53, 2012.

[16] 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, 2003.

[17] A. Estabrooks and N. Japkowicz, "A mixture-of-experts framework for learning from imbalanced data sets," in *International Symposium on Intelligent Data Analysis*. Springer, 2001, pp. 34–43.

[18] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[20] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," in *International symposium on intelligence computation and applications*. Springer, 2009, pp. 461–471.

[21] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[22] M. Hossin, M. Sulaiman, A. Mustapha, N. Mustapha, and R. Rahmat, "A hybrid evaluation metric for optimizing classifier," in *2011 3rd Conference on Data Mining and Optimization (DMO)*. IEEE, 2011, pp. 165–170.

[23] R. Ranawana and V. Palade, "Optimized precision-a new measure for classifier performance evaluation," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 2254–2261.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[25] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, p. 90, 2007.

[26] 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.

[27] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 14, no. 3, pp. 515–516, 1968.

[28] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Nashville, USA, 1997, pp. 179–186.

[29] I. Tomek, "Two modifications of cnn," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 6, p. 769–772, 1976.

[30] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 2001, pp. 63–66.