# Impact of Data Balancing During Training for Best Predictions

Suleiman Ali Alsaif and Adel Hidri
Computer Department, Deanship of Preparatory Year and Supporting Studies
Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
E-mail: saalsaif@iau.edu.sa, abhidri@iau.edu.sa

**Keywords:** machine learning, undersampling, oversampling, data balancing, ensemble learning

**Received:** March 26, 2021

*To protect the middle class from over-indebtedness, banking institutions need to implement a flexible analytic-based evaluation method to improve the banking process by detecting customers who are likely to have difficulty in managing their debt. In this paper, we test and evaluate a large variety of data balancing methods on selected machine learning algorithms (MLAs) to overcome the effects of imbalanced data and show their impact on the training step to predict credit risk. Our objective is to deal with data unbalance to achieve the best predictions. We investigated the performance of these methods by different learners when classification models are trained using MLAs.*

*Povzetek: Predstavljena je metoda strojnega učenja z neuravnoteženimi podatki za oceno tveganja prezadolžitve srednjega razreda.*

## 1 Introduction

To become compliant with changing and stricter regulatory demands in the world, banks are focused on undertaking preventive measures for effective credit risk management. One of its drivers is to strengthen existing learning models [1, 2] to increase their predictive power and help detect future defaults in advance.

Due to the inherent complex characteristics of imbalanced data sets, learning from such data requires new understandings, principles, algorithms, and tools to transform a vast amount of raw data efficiently into information and knowledge representation. The imbalanced learning problem is concerned with the performance of learning algorithms in the presence of under-represented data and severe class distribution skews [3, 4].

Most standard algorithms assume or expect balanced class distributions. Therefore, when presented with complex imbalanced datasets, these algorithms fail to properly represent the distributive characteristics of the data and resultantly provide unfavorable accuracies across the classes of the data. The imbalanced learning problem represents a recurring problem of high importance [5, 6, 7].

In this paper, we will present the impact of data balancing strategies at the training step. We test and evaluate a large variety of data balancing methods on selected MLAs to overcome the effects of imbalanced data and show their impact at the training step to predict credit risk. Our objective is to deal with data unbalance to achieve the best predictions. We investigated the performance of these methods by different learners when classification models are trained using MLAs.

The remainder of this paper is organized as follows: in section 2, we will highlight related work related to sampling methods for imbalanced learning. Section 3 provides an overview of the evaluation metrics in machine learning. In section 4, we will describe the enhanced training approach using data balancing strategies. In section 5, a performance analysis is brought forward to set the advantages of the proposed evaluation method. Section 6 discusses the work. Finally, we will conclude in section 7 our work and bring out some insights and potential future works.

## 2 Related work

Typically, the use of sampling methods in imbalanced learning applications consists of the modification of an imbalanced data set by some mechanism to provide a balanced distribution. Studies have shown that for several base classifiers, a balanced data set provides improved overall classification performance compared to an imbalanced dataset. These results justify the use of sampling methods for imbalanced learning [8].

Random undersampling (RUS) removes data from the original dataset. In particular, we randomly select a set of majority class examples and remove these samples from the dataset. Consequently, undersampling readily gives us a simple method for adjusting the balance of the original dataset [6, 8]. Figure 3 shows random undersampling and oversampling.

The mechanics of RUS follow naturally from its description by adding a set sampled from the minority class: for a set of randomly selected minority examples, augment the original set by replicating the selected examples and adding them to the dataset. In this way, the number of total examples in the minority class is increased and the class distribution balance is adjusted accordingly. This provides a mech-

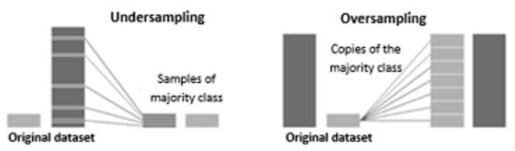


Figure 1: Random undersampling and oversampling.

anism for varying the degree of class distribution balance to any desired level.

The objective of the Easy Ensemble method [9, 10, 11, 26] is to overcome the deficiency of information loss introduced in the traditional RUS method. Easy Ensemble can be considered as an unsupervised learning algorithm that explores the majority class data by using independent random sampling with replacement [8].

The Balance Cascade algorithm takes a supervised learning approach that develops an ensemble of classifiers to systematically select which majority class examples to undersample. In Balance Cascade, the sequential dependency between classifiers is mainly exploited for reducing the redundant information in the majority class. This sampling strategy leads to a restricted sample space for the following undersampling process to explore as much useful information as possible [5].

Another example of informed undersampling uses the K-Nearest Neighbor (KNN) classifier [12] to achieve undersampling. Based on the characteristics of the given data distribution, four KNN undersampling methods were proposed, namely, NearMiss-1 and NearMiss-3 methods [5]. The NearMiss-1 method selects those majority examples whose average distance to the three closest minority class examples is the smallest [8]. The NearMiss-3 selects a given number of the closest majority examples for each minority example to guarantee that every minority example is surrounded by some majority examples [8].

SMOTE (Synthetic Minority Oversampling Technique) is an oversampling method. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbs an instance one attribute at a time by a random number within the difference to the neighboring instances [5].

Borderline-SMOTE1 and Borderline-SMOTE2 only oversample or strengthen the borderline minority examples [5]. The detailed procedure of Borderline-SMOTE1 is as follows: for every instance p in the minority class, we calculate its m nearest neighbors from the whole training set. If all nearest neighbors of $p$ are majority examples, $p$ is considered noise and is not operated in the following steps. If the number of $p'$ majority nearest neighbors is larger than the number of minority ones, $p$ is considered easily misclassified and put into a set DANGER. The examples in DANGER are the borderline data of the minority class. Then, new synthetic data are generated along the line between the minority borderline examples (data in *DANGER*) and their nearest neighbors of the same class, thus strengthening borderline examples [5].

Data cleaning techniques, such as Tomek links, have been effectively applied to remove the overlapping that is introduced by sampling methods [5]. One can use Tomek links to cleanup unwanted overlapping between classes after synthetic sampling, where all Tomek links are removed until all minimally distanced nearest-neighbor pairs are of the same class. By removing overlapping examples, one can establish well-defined class clusters in the training set, which can, in turn, lead to well- defined classification rules for improved classification performance. The Edited Nearest Neighbor (ENN) rule removes examples that differ from two of its three nearest neighbors.

## 3 Evaluation metrics in machine learning

In machine learning, there are a variety of metrics to evaluate the performance of classifiers. In this subsection, we will mention the most common evaluation metrics used in our work [13, 14, 15].

A confusion matrix is a specific table that is usually used to represent the performance of a classification model on test data for which the true values are known [16, 17]. As shown in Table 1, it consists of two columns and two rows that contain the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Table 1 is an example of confusion matrix with two-class classifier.

The elements in the diagonal represent the number of instances for which the predicted class is equal to the true class, while off-diagonal elements are those that are misclassified by the classifier. The higher the diagonal values of the confusion matrix, the better, indicating correct predictions. In the case of over-indebtedness detection, over-indebted customers are the positive samples while no over-indebted customers are the negative samples.

Considering the confusion matrix in the table 1, we define the most basic terms as follows:

- TN: If the actual class is no over-indebted and the predicted value is also no over-indebted, so we have a TN. In practice, we want this value to be as high as possible.

- FP: If the actual class is not over-indebted and the predicted value is over-indebted, we have then a FP. In practice, we want this value to be as low as possible because it impacts customers, their banking experiences, and their lives.

- FN: If the actual class is over-indebted and the predicted value is not over-indebted, then we have a FN. Like the FP, we want this value to be as low as possible because the bank loses money when the customer who is deeply in debt, is considered as normal.

Table 1: Confusion matrix.

| | | Predicted class | |
|---|---|---|---|
| | | No over-indebted | Over-indebted |
| Actual class | No over-indebted | $TN$ | $FP$ |
| | Over-indebted | $FN$ | $TP$ |

- TP: If the actual class is over-indebted and the predicted value is also over-indebted, we have a TP. Like the TN, we want to predict all customers that will be in over-indebtedness to allow the bank to establish a support program for its consumers.

The sensitivity, also called the True Positive Rate (TPR), measures the proportion of positives that are correctly identified. The sensitivity is defined by the formula in Eq. (1) [18].

$$Sensitivity = TP/(TP + FN) \tag{1}$$

The specificity, also called the True Negatives Rate (TNR), measures the proportion of negatives that are correctly identified. The specificity is defined by the formula in Eq. (2) [18].

$$Specificity = TN/(TN + FP) \tag{2}$$

The Gini coefficient is defined as twice the area between the ROC (Receiver Operating Characteristic) curve and the chance diagonal. It takes values between 0 (no difference between the score distributions of the two classes) and 1 (complete separation between the two distributions). The ROC curve (see figure 1) is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied.

The curve is created by plotting the TPR on the y-axis against the False Positive Rate (FPR) on the x-axis at various threshold settings. The TPR rate is also depicted as sensitivity in machine learning. The FPR rate can be calculated as follows:

$$FPR = 1 - Specificity \tag{3}$$

A classification of a new object is obtained by comparing the score $s$ of the object with a classification threshold $t$. If $s > t$ the object is classified as coming from class 1 and if $s > t$ as coming from class 0.

The AUC (Area Under the Curve) is the area under the ROC curve. The performance of a classifier model is calculated by calculating the AUC [19, 20].

Related to the figure 1, we have:

$$AUC = Area\ A + Area\ B \tag{4}$$

The Gini coefficient is given by the following equation:

$$Gini = 2 * AUC - 1 \tag{5}$$

- At point $(0, 0)$, the classifier considers all instances as negatives. There are no false positives, but also no TP.

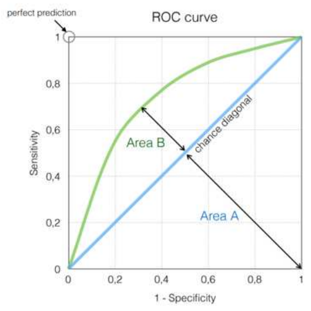


Figure 2: ROC curve.

- At point $(1, 1)$, the classifier considers all instances as positives. There are no false positives, but also no TN.

- At point $(1, 0)$, the classifier has no TP and no TN. In this case, the performance of the classifier is worse than random. Its performance can be totally improved by selectively reversing the classifiers answer.

- At point $(0, 1)$, the classifier has $100\%$ sensitivity and $100\%$ specificity. This point is called a perfect classification.

## 4 Enhancing training using data balancing strategies

The class imbalance problem corresponds to domains for which one class is represented by a large number of examples while the other is represented by only a few.

The used data consists of a collection of consumers to whom the bank gives credit. Analyzing the consumers in a binary sense, the natural classes that arise are negative or positive for a consumer who payed back or did not pay back their credit, respectively. From experience, one would expect the number of no over-indebted clients to exceed greatly the number of over-indebted clients. Indeed, this dataset contains 281,616 negative (majority class) samples and 678 positive (minority class) samples. We will have a classifier tend to provide a severely imbalanced degree of accuracy, with the majority class having close to 100% accuracy and the minority class having accuracies of 0-10% (see Figure 2).

Data are extremely imbalanced. We apply several sampling methods on the train set using the imbalanced learn package. It is a Python package offering a number of resampling techniques commonly used in datasets showing strong between-class imbalance. It is compatible with Scikit-learn and is part of the scikit-learn-contrib project [21].

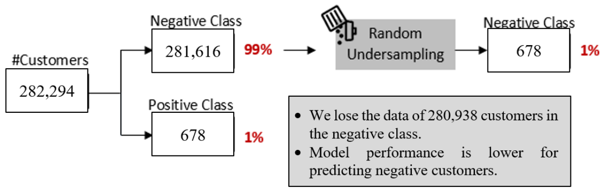

#Customers
282,294
Negative Class
281,616
99%
Positive Class
678
1%
Random Undersampling
Negative Class
678
1%
We lose the data of 280,938 customers in the negative class.
Model performance is lower for predicting negative customers.

Figure 3: Evaluation process.

Suppose a classifier achieves 10% accuracy on the minority class of the bank data set. Analytically, this would suggest that 610 minority samples are misclassified as majority samples. The consequence of this is equivalent to a 610 over-indebted clients classified as no over-indebted. In the bank, this consequence can be overwhelmingly costly.

Therefore, it is evident that for this domain, we require a classifier that will provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class.

Furthermore, this also suggests that the conventional evaluation practice of using singular assessment criteria, such as the overall accuracy or error rate, does not provide adequate information in the case of imbalanced learning. The immense hindering effects that these problems have on standard learning algorithms are the focus of most of the existing solutions. When standard learning algorithms are applied to imbalanced data, the induction rules that describe the minority concepts are often fewer and weaker than those of majority concepts, since the minority class is often both outnumbered and under-represented [8].

## 5 Computational results and analysis

Baseline algorithm performs the RUS for balancing data. The bank selects randomly a set from the negative (no over-indebted) class equals to the set of the positive (over-indebted) class and it removes all the other negative samples from the data. When applying random RUS in the train set, we might be losing representative samples of the negative class.

Our objective is to deal with data unbalance to achieve the best predictions for both classes and avoid favoring one of the classes over the other. We aim to get as much information as possible from the large number of majority available class examples.

The experiments were conducted using a 10-fold stratified cross-validation [22]. The experiments have been conducted to find the best sampling techniques. We investigated the performance of RUS, SMOTE, Balance Cascade, and Easy Ensemble by different learners when classification models are trained using XGB (eXtreme Gradient Boosting) [23], LR (Logistic Regression) [24], and ET (Extremely Randomized Trees) [25]. We used AUC (Area Under the Curve), Gini coefficient, and confusion matrix to evaluate the compared algorithms. We will compare the performance of the models before and after resampling to compare the increase of classifier performance due to resampling.

### 5.1 Undersampling methods

The bar charts of the figure 4 show the variation of Gini coefficient, sensitivity, and specificity after applying the NearMiss-1 and NearMiss-3 sampling methods to train data. The Gini coefficient achieves very high values for all algorithms except the XGB. The values for sensitivity are a little improved, but for the specificity the values are lower than the random under sampling for all algorithms.

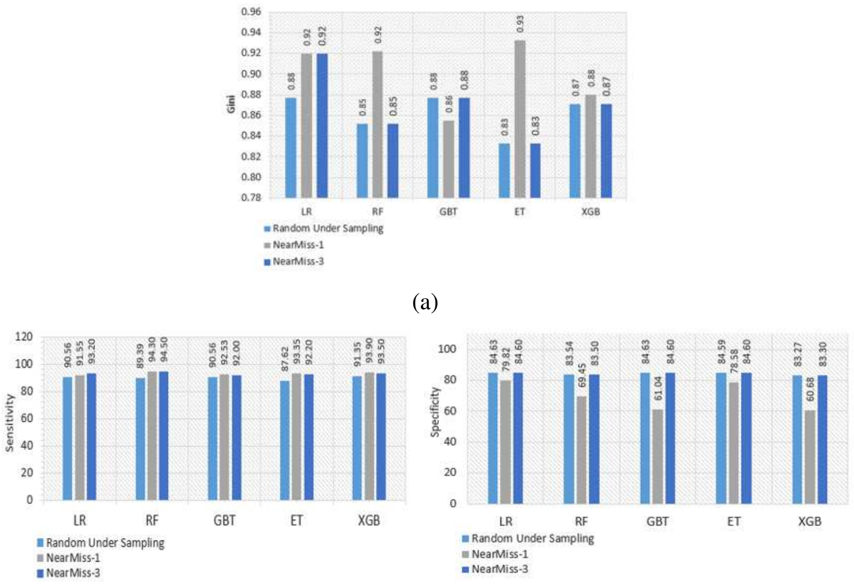
Undersampling tends to outperform the model in terms of sensitivity (see figure 4(b)), but at a very high cost of specificity (see figure 4(c)). However, while the value of sensitivity increased, the value of specificity decreased. Applying class-imbalance learning NearMiss-1 method on this data set is not necessarily beneficial.

As shown in the figure 4, the values of the three metrics are greater than the baseline. But the prediction of the positive examples is better than the negatives ones. For example, XGB algorithm gives 93.5% (see figure 4 (b)) in sensitivity whereas 88.4% in specificity (see figure 4 (c)).

### 5.2 Oversampling coupled with Undersampling

We apply SMOTE methods to the data and then the random undersampling. Figure 5 shows the impact of the SMOTE and random undersampling on the performance of the models. This technique tends to produce high specificity without reducing the sensitivity when compared to other resampling techniques. As shown in Figure 5, the performance of this technique is similar to the combination of SMOTE+RUS. It tends to produce a good prediction for TN as well as the TP. The combination of SMOTEB1 with NearMiss-1 performs very poorly. The three metrics have lower values. This sampling technique is not helpful for our data and our models.

As shown in the figure 5, it can be seen that the sensitivity fell slightly. However, the Gini (see figure 5(a)) and the specificity (see figure 5 (c)) are improved. It is clear that SMOTE Tomek Links coupled with NearMiss-3 performs fairly. It tends to have a good value for the three metrics. Its results look like the results of the technique of SMOTEB2 coupled to NearMiss-3. SMOTE ENN coupled with random under sampling performs badly. It produces high specificity, but with very poor sensitivity (see figure 5(b)).



(a) (b) (c)

Figure 4: (a) Gini, (b) Sensitivity, and (c) Specificity of undersampling methods.

For example, we can see in the figure 5(c) that the XGB algorithm has 92.20% in specificity against 78.60% in sensitivity, which is a relatively bad value. It sacrifices high sensitivity for high specificity.

## 5.3 Ensemble

As shown in the figure 6, it is clear that the Easy Balance gives better results with XGB and LR algorithms in contrast to the other ones. While XGB and LR achieve very good values for both sensitivity (see figure 6(b)) and specificity (see figure 6(c)). They perform fairly for both; the other algorithms produce high specificity and very poor sensitivity.

From the figure 6, we can observe that the Balance Cascade method gives similar results to the Easy Ensemble but with a little decrease in terms of sensitivity.

## 6 Discussion

We compare the results of our experiments and find that the best resampling technique to use is often dataset and model dependent, certain resampling techniques tend to perform better when coupled with certain classifiers. It is obvious that there is not a best sampling technique for all models. The results show that the Easy Ensemble performs better than the others when XGB is used as a classifier.

It attains a better balance between sensitivity and specificity than almost all other methods. Instead of trading off one metric against another, it succeeds to give similar good results for all metrics. The graph in the figure 7 summarizes the effect of feature selection and data balancing with XGB algorithm on our data.

In Figure 8, we clearly see that:

– After feature selection: 4,564 clients were misclassified before as over indebted are correctly identified as normal and 3 clients were misclassified as normal are correctly identified as over-indebted.

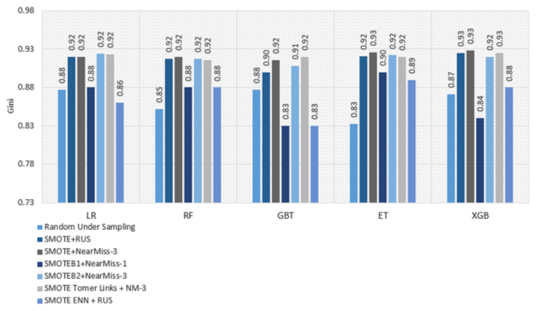
– After data balancing: we misclassified only 1 clients from the positive class to classify more 856 no over-indebted clients correctly.

## 7 Conclusion

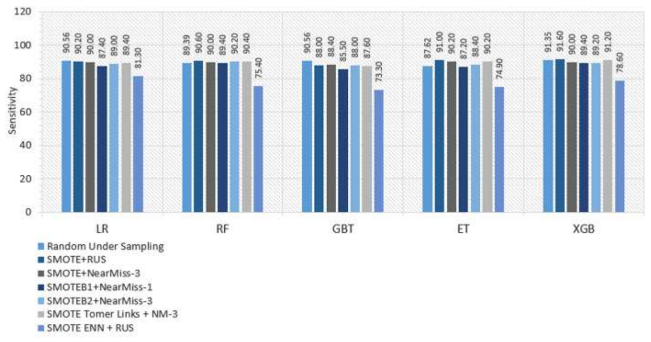
In this work, we tested and evaluated a large variety of data balancing methods on selected MLAs to overcome the effects of imbalanced data and show their impact on the training step to predict credit risk management. Our objective is to deal with data unbalance to achieve the best predictions.

After balancing the data with the easy ensemble method, we reduced the overfitting of the positive class and we reached a good balance between sensitivity and specificity. These contributions have shown that there isn't any ideal sampling method for balancing the data, and we have demonstrated through conducted experiments that it is possible to improve the overall performance of learning algorithms in the field of over-indebtedness prediction by using data balancing methods.

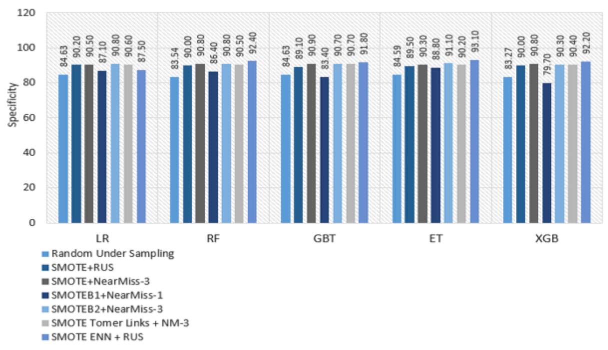
The set of conducted experiments aims at validating the importance of pre-processing the data before applying any machine learning algorithm. Balancing the data with advanced methods and changing the machine learning algorithm helped us to build a new effective and accurate pre-



(a)

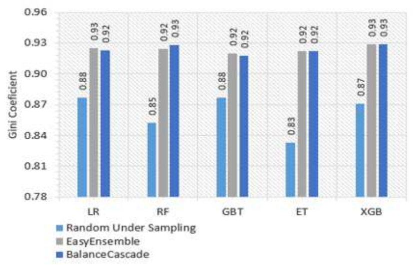


(b)

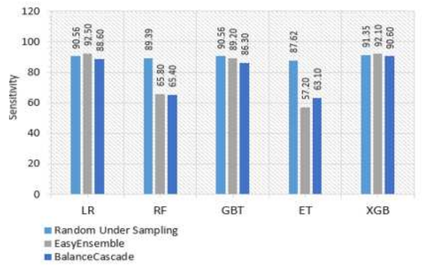


(c)

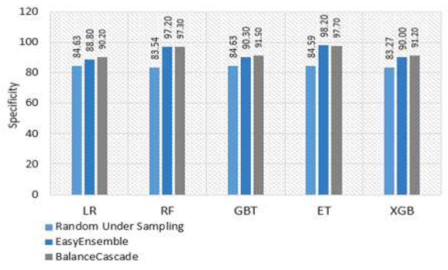
Figure 5: (a) Gini, (b) Sensitivity, and (c) Specificity of oversampling coupled with undersampling.



(a)

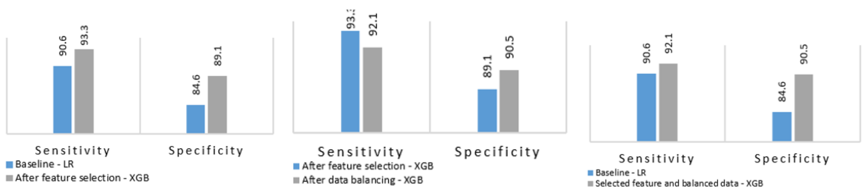


(b)



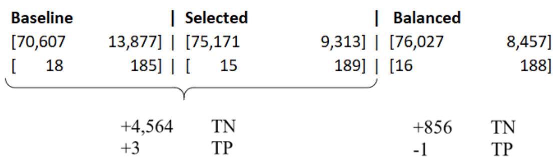
(c)

Figure 6: (a) Gini, (b) Sensitivity, and (c) Specificity of ensemble methods.



(a) (b) (c)

Figure 7: Improvement of sensitivity and specificity.



```
Baseline                | Selected                | Balanced
[70,607     13,877] | [75,171          9,313] | [76,027        8,457]
[    18        185] | [    15            189] | [16              188]

        +4,564   TN                    +856    TN
        +3       TP                    -1      TP
```

Figure 8: Confusion matrix.

dictor. credit risk models need to consolidate new COVID-19 pandemic related data points to guarantee their output prevails valid and robust. Better and deeper insights can be accomplished by boring into a broader range of data sources as well as upgrading data platform technologies. This relates to our short-term work.

## References

[1] A.A. Abaker and F.A. Saeed (2021) A comparative analysis of machine learning algorithms to build a predictive model for detecting diabetes complications, *Informatica (Slovenia)*, vol. 45, no. 1, pp. 117–125, `https://doi.org/10.31449/inf.v45i1.3111`.

[2] J. Sun (2021) Prediction and estimation of book borrowing in the library: Machine learning, *Informatica (Slovenia)*, vol. 45, no. 1, pp. 163–168, `https://doi.org/10.31449/inf.v45i1.3431`.

[3] L.J. Mena and J.A. Gonzalez (2009) Symbolic one-class learning from imbalanced datasets: Application in medical diagnosis, *Int. J. Artif. Intell. Tools*, vol. 18, no. 2, pp. 273–309, `https://doi.org/10.1142/S0218213009000135`.

[4] W.J. Liu, X.Y. and Z. Zhou (2009) Exploratory undersampling for class-imbalance learning, *Trans. Sys. Man Cyber. Part B*, vol. 39, `https://doi.org/10.1109/TSMCB.2008.2007853`.

[5] A. Mahani and A. Baba-Ali (2019) *Classification Problem in Imbalanced Datasets*, `https://doi.org/10.5772/intechopen.89603`.

[6] L.E.B. Ferreira, J.P. Barddal, F. Enembreck, and H.M. Gomes (2018) An experimental perspective on sampling methods for imbalanced learning from financial databases, *Proceedings of the 2018 International Joint Conference on Neural Networks*, pp. 1–6, `https://doi.org/10.1109/IJCNN.2018.8489290`.

[7] G. Goel, L. Maguire, Y. Li and S. McLoone (2013) Evaluation of sampling methods for learning from imbalanced data, *in Huang DS., Bevilacqua V., Figueroa J.C., Premaratne P. (eds) Intelligent Computing Theories*, pp. 392–401, `https://doi.org/10.1007/978-3-642-39479-9\_47`.

[8] H. He and E. Garcia (2009) Learning from imbalanced data,*IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, `https://doi.org/10.1109/TKDE.2008.239`.

[9] T.G. Dietterich (2000) Ensemble methods in machine learning, *Proceedings of the First International Workshop on Multiple Classifier Systems*, pp. 1–15, `https://doi.org/10.1007/3-540-45014-9_1`.

[10] M. Ghosh and P.G. Sanyal (2018) Performance assessment of multiple classifiers based on ensemble feature selection scheme for sentiment analysis, *Applied Computational Intelligence and Soft Computing*, pp. 1–12, `https://doi.org/10.1155/2018/8909357`.

[11] R.R. Saifan, K. Sharif, M. Abu-Ghazaleh, and M. Abdel-Majeed (2020) Investigating algorithmic stock market trading using ensemble machine learning methods, *Informatica (Slovenia)*, vol. 44, no. 3, pp. 311–325, `https://doi.org/10.31449/inf.v44i3.2904`.

[12] K. Q. Weinberger and L.K. Saul (2009) Distance metric learning for large margin nearest neighbor classification, *JMLR*, pp. 207–244, `https://doi.org/10.5555/1577069.1577078`.

[13] M. Hossin and M. Sulaiman (2019) A Review on Evaluation Metrics for Data Classification Evaluations, *International Journal of Data Mining &*

*Knowledge Management Process (IJDKP)*, vol. 5, no. 2, pp. 1–11, `https://doi.org/10.5121/ijdkp.2015.5201`.

[14] M. Fatourechi, R. Ward, S. Mason, J. Huggins, A. Schlogl, and G. Birch (2009) Comparison of evaluation metrics in classification applications with imbalanced datasets, *Proceedings of the Seventh International Conference on Machine Learning and Applications*, pp. 777–782, `https://doi.org/10.1109/ICMLA.2008.34`.

[15] K.M. Ghori, R.A. Abbasi, M. Awais, M. Imran, A. Ullah, and L. Szathmary (2020) Performance analysis of different types of machine learning classifiers for non-technical loss detection, *IEEE Access*, vol. 8, pp. 16 033–16 048, `https://doi.org/10.1109/ACCESS.2019.2962510`.

[16] K. Ting (2017) *Encyclopedia of Machine Learning and Data Mining*, ser. Springer Reference, C. Sammut and G. I. Webb, Eds., `https://doi.org/10.1007/978-1-4899-7687-1`.

[17] O. Caelen (2017) A bayesian interpretation of the confusion matrix, *Annals of Mathematics and Artificial Intelligence*, vol. 81, `https://doi.org/10.1007/s10472-017-9564-8`.

[18] S. Kumar (2020) Sensitivity, specificity, generalizability, and reusability aspirations for machine learning (ML) models in MHealth, *Proceedings of Deep Learning for Wellbeing Applications Leveraging Mobile Devices and Edge Computing*, pp. 1, `https://doi.org/10.1145/3396868.3402495`.

[19] P. Flach, J. Hernández-Orallo, and C. Ferri (2011) A coherent interpretation of auc as a measure of aggregated classification performance, *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 657–664, `https://doi.org/10.5555/3104482.3104565`.

[20] L.E. Raileanu and K. Stoffel (2004) Theoretical comparison between the gini index and information gain criteria, *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, `https://doi.org/10.1023/B:AMAI.0000018580.96245.c6`.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.* (2011) "Scikit-learn: Machine learning in python, *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, `https://doi.org/10.5555/1953048.2078195`.

[22] S. Raschka, J. Patterson, and C. Nolet (2020) Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence, *Information*, vol. 11, no. 4, pp. 193, `https://doi.org/10.3390/info11040193`.

[23] H. Drucker and C. Cortes (1995) *Boosting decision trees*. Cambridge, MA, USA: MIT Press, pp. 479–485, `https://doi.org/10.5555/2998828.2998896`.

[24] D.W. Hosmer and S. Lemeshow (2013) *Applied logistic regression*. John Wiley and Sons, `https://doi.org/10.1002/9781118548387`.

[25] C. Latha and S. Jeeva (2019) Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques, *Informatics in Medicine Unlocked*, vol. 16, pp. 100203, `https://doi.org/10.1016/j.imu.2019.100203`.

[26] E.K. Ampomah, Z. Qin and G. Nyame and F.E. Botchey (2020) Stock market decision support modeling with tree-based AdaBoost ensemble machine learning models, *Informatica (Slovenia)*, vol. 44, no. 4, pp. 477–489, `https://doi.org/10.31449/inf.v44i4.3159`.