Improving Machine Learning Performance in Credit Scoring by Data Analysis and Data Pre-Processing

Bogdan Ichim[1,2] and Bilal Issa[1]
[1]*Faculty of Mathematics and Computer Science, University of Bucharest, Str. Academiei 14, Bucharest, Romania*
[2]*Simion Stoilow Institute of Mathematics of the Romanian Academy, Str. Calea Grivitei 21, Bucharest, Romania*

Keywords: Data Science, Data Analysis, Data Pre-Processing, Balanced Random Forest, Extreme Gradient Boosting, Light Gradient Boosting Machine, Neural Networks.

Abstract: In this paper we showcase several data analysis and data pre-processing techniques which, when applied to the dataset Give Me Some Credit, lead to improvements in the performance of several machine learning algorithms in classifying defaulters and non-defaulters in comparison with other existing solutions from the literature. Our study underscores the importance of these techniques in data science in general, and in enhancing the machine learning outcomes in particular.

1 INTRODUCTION

Credit scoring is the process of evaluating individuals' credit worthiness based on their financial history and behaviour (including income, age, debt, etc.) and predicting whether they will experience financial distress in the future. It plays a crucial role in preventing financial losses by avoiding loans to high-risk individuals and ensuring that creditworthy individuals are not overlooked. However, predicting financial distress can be difficult, requiring a lot of data to capture underlying patterns of defaulters and non-defaulters. In this paper we expand on the work of (Lessmann et al., 2015) and (Gunnarsson et al., 2021). We attempt to address the difficulties associated with a certain dataset from an analytical point of view and study methods that can overcome the obstacles like missing data, data errors and severe class imbalance.

In many papers like (Lessmann et al., 2015) and (Gunnarsson et al., 2021) the authors assessed various machine learning algorithms across different datasets, without directly manipulating the data, as their focus is on the algorithms themselves. Consequently, we selected one dataset of moderate complexity – which, while it doesn't achieve the highest performance scores, it presents enough challenges for further exploration and improvement.

The contributions made by this paper are the following:

- We have analysed the Give Me Some Credit (GMC) dataset. This is useful for further data pre-processing and choosing the right techniques to apply.
- Pre-processed the data to a form suitable as input for several machine learning algorithms (including neural networks).
- We have studied the effect of imputation algorithms on GMC dataset and their impact on the algorithm's performance.
- Engineered seven new features that are ranked in the top 10 most significant predictors for all algorithms tested.
- Tested a wide range of data-level class balancing methods and compared their performance to algorithm-level approaches, finding the latter to be the best balancing practice for the particular case of the GMC dataset.
- Implemented the balanced sampling strategy found in the Balanced Random Forest algorithm and compared its results to other algorithms well-known for their performance like Extreme Gradient Boosting.
- Improved the state-of-the-art results by 1% in the AUC score through our comprehensive analysis and carefully engineered features.

The code and the pre-processed dataset used for these experiments are available on demand.

Ichim, B. and Issa, B.
Improving Machine Learning Performance in Credit Scoring by Data Analysis and Data Pre-Processing.
DOI: 10.5220/0013118500003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 249-255
ISBN: 978-989-758-737-5; ISSN: 2184-433X

2 RELATED WORK

In recent years machine learning has captured the attention of academia, business and industry due to its remarkable capacity to process vast volumes of data in order to identify complex patterns.

In particular it allows banks and other similar financial institutions to evaluate the credit worthiness of companies or individuals with greater accuracy, objectively and without bias, in order to approve or deny loan requests.

The Give Me Some Credit dataset was first introduced in a Kaggle competition and further was utilized as a standard dataset for credit scoring in a number of academic publications. Some of these studies achieved their best results using ensemble methods, as demonstrated by (Lessmann et al., 2013) and (Lessmann et al., 2015), while others by employing gradient boosting algorithms, such as in (Gunnarsson et al., 2021) and (Gidlow, 2022).

However, most of these studies overlooked the essential data pre-processing steps, including the detection of outliers, data imputation, feature engineering and class balancing, which may lead to a suboptimal performance of the algorithms tested.

A machine learning model which proved to be very useful in our experiments is the Balanced Random Forest (BRF) algorithm, which is a variation of the Random Forest (RF) algorithm. The former algorithm makes sure that all classes in the dataset are represented equally at the training step, by randomly selecting the same number of samples from all classes. This variation of the RF algorithm makes it an ideally suited algorithm for tackling severely unbalanced datasets, in particular for the Give Me Some Credit dataset which will be further analysed. For more information we point the reader to (Chen et al., 2004).

In parallel a new architecture of neural networks known as Attentive Interpretable Tabular Learning (TabNet) was proposed in (Arik and Pfister, 2021). It was developed specifically to handle tabular data, making it a suitable neural network to be used for handling credit data. It utilizes an attention mechanism in order to select the most relevant features at each decision-making stage. This allows the model to concentrate on different subsets of features, enhancing interpretability and functionality on structured data.

In this paper we investigate the performance of the BRF, TabNet and other algorithms in combination of several data pre-processing techniques, improving the existing solutions proposed by other authors for the GMC dataset.

3 METHODOLOGY

3.1 Objectives and Workflow

The goal of this paper is to study the capacity of several machine learning algorithms to assess the credit worthiness of individuals by predicting whether or not a default will happen in the following two years. By conducting a deep analysis of the problem, we also aim to improve some standard metrics for performance which were previously achieved by other authors. A particular aspect of the GMC dataset is that it contains imbalanced samples of defaulters and non-defaulters. The problem of identifying defaulters prevails over the problem of identifying non-defaulters in the world of credit scoring because it helps reducing losses for the financial institutions. As a result, the model's ability to predict each class independently should be considered, rather than just its accuracy.

To accomplish these objectives, we use in our article a structured workflow. First, the structure and the content of the GMC dataset are analysed and understood. Next, basic data cleaning is performed, including the removal of duplicates and the removal of some attributes that are not relevant. After that, a comprehensive stage of exploratory data analysis is made in order to find potential outliers and correlations in the data. Subsequently, a process of feature engineering is done with the scope of adding new features to the dataset in order to accentuate the significance of certain features. (This turned out later to be a crucial step for getting the best possible outcomes.) Several class balancing methods were also employed for addressing the severe imbalance of the dataset, before finally applying the machine learning algorithms.

3.2 Dataset

The Give Me Some Credit dataset was released for a Kaggle competition which took place from September to December 2011. It is frequently used by various researchers in order to predict borrower distress within two years. The dataset comprises 250,000 anonymized entries with twelve key features, including Age, Monthly Income and Debt Ratio. The primary target variable in these studies (“SeriousDlqin2yrs”) indicates whether borrowers experienced financial distress, defined as being 90 days or more overdue on any credit account within two years (1 for distress, 0 otherwise). The GMC dataset is considered difficult for the fact that it contains a few features, predicts default two years in

the future and has a severe class imbalance (6.7% defaulters) compared to other available datasets that predict default only six months in the future and contain more balanced samples, for example 20% and 50% default rate in the HMEQ and the Australian Credit respectively. The difference in the classification performance as measured by AUC in previous studies is illustrated in Figure 1.

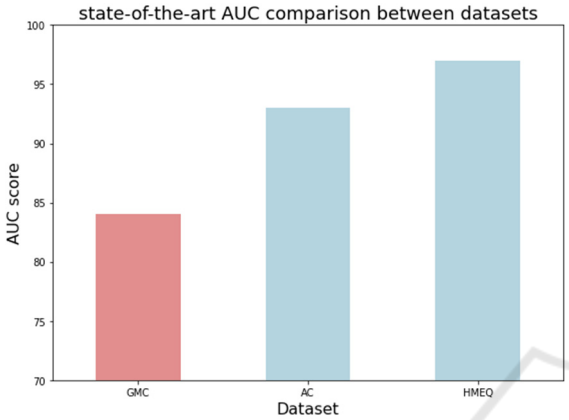

Figure 1: Comparison between the best AUC scores obtained on the GMC dataset in red, and the Australian Credit (AC) and HMEQ.

3.3 Technical Implementation

To implement the code for our experiments, we have used the programming language Python together with libraries like TensorFlow, PyTorch and Scikit-Learn. TensorFlow is a library developed by Google (Google Brain, 2016) for employing neural networks and machine learning algorithms. Another important library is PyTorch (Paszke et al., 2019), developed by Meta and having a similar purpose as TensorFlow. Lastly, the most used library in our paper is Scikit-learn (Pedregosa et al., 2011), which contains implementations of the most popular machine learning algorithms and can be used to define and train various models. It also contains other algorithms related to data pre-processing such as imputation, data-level balancing and data splitting algorithms.

4 DATA PRE-PROCESSING

4.1 Outlier Elimination

The first step in data pre-processing was to detect and eliminate outliers. We tested various methods and discovered that the Z-score and the regression analysis proved to be the most effective techniques among them.

The Z-score procedure automatizes the process of detecting outliers. All data points with distance from the mean bigger than a set number of standard deviations are considered to be outliers. For our project we have chosen to set the threshold at 3 after analysing different outcomes and establishing this threshold as being the best. For example, this threshold correctly classified individuals whom age is less than 18 as outliers, which is the case since the minimum age accepted to apply for a loan is 18. In the end 5047 entries were identifies as outliers by this method, out of which 517 were in the defaulters class.

The other technique which we employed in order to eliminate outliers is the regression analysis. This method is used to examine the relationship between a particular predictor and a fixed target by training a linear model and then plotting the fitted line. The points situated beyond a certain distance from the fitted line (measured in standard deviations) are further identified as outliers.

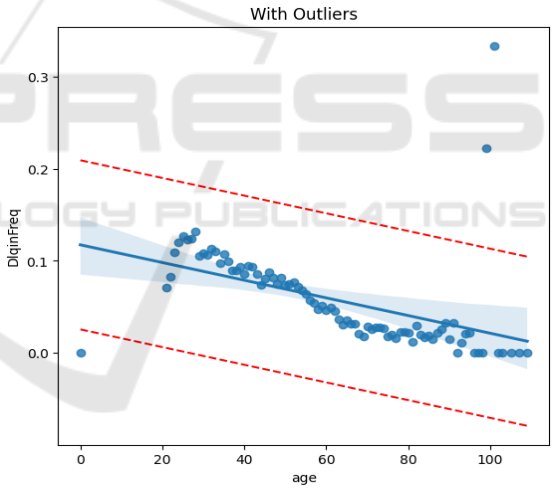

Figure 2: The fitted line of the age groups and their delinquency rate. The red lines represent the chosen distances from the fitted line.

4.2 Imputation

In order to deal with missing values in “Monthly Income” and “Number of dependents” columns three methods for imputation were tested. The first two methods were the KNN Imputer and the Simple Imputer, chosen for their efficiency. However, the resulting distributions and performance scores indicate that these techniques are not suitable for the GMC dataset. The last method was the Iterative

Imputer, which has a built-in machine learning algorithm that sees a feature with missing values as a target and the remaining variables as predictors in order to iteratively input the missing values.

All three imputation outcomes were tested on several different machine learning algorithms (including Random Forest and Extreme Gradient Boosting). In the end the Iterative Imputer delivered the most satisfactory results.

4.3 Feature Engineering

One feature engineering technique that is useful for this dataset is the log transformation. This method is particularly efficient for transforming features with skewed distributions. This is because many machine learning algorithms assume that the data is normally distributed, therefore it may be more difficult for some algorithms to fit such data. A good example of a feature with skewed distribution in the GMC dataset is the "monthly income", as it originally contains values that range from zero to three million.

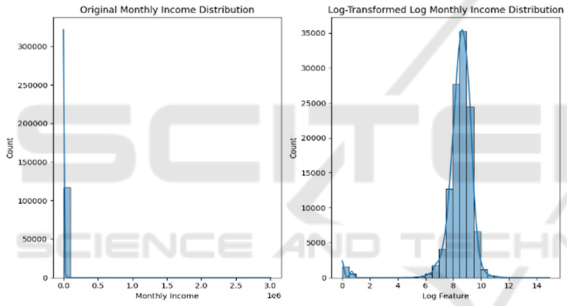

Figure 3: Showcasing the skewed distribution of "monthly income" before and after applying the logarithm function.

Another aspect of feature engineering is the creation of new variables based on the observed ones, with the role of emphasizing relationships that otherwise would not appear in the original dataset, resulting in an improvement in the predictive power. The newly created features are the following:

- Average Months Late;
- Max Late Payment;
- Monthly Debt;
- Monthly Balance;
- Credit Utilization;
- Income Per Dependent;
- Total Number of Past Due.

4.4 Class Balancing

To address the severe imbalance in the data, we tested fifteen data-level and three algorithm-level balancing techniques. Among the best performing data-level methods, we mention Tomek-links (Ivan, 1976), Synthetic Minority Over-sampling Technique (SMOTE), and Edited Nearest Neighbours (Tang et al., 2015). Comparing the two balancing methods on the GMC dataset, we found that algorithm-level sampling techniques consistently outperformed others when combined with pre-processing steps. Most methods, like Under-sampling and Cluster Centroids, reduced the AUC or showed minimal improvement, with exceptions such as Tomek Links and ENN, which performed better. However, these improvements were still not sufficient to match the performance achieved through algorithm-level balancing technique.

4.5 Feature Selection

The last part of the pre-processing step was to implement and analyse various feature selection methods such as the Recursive Feature Elimination (RFE) proposed by (Guyon et al., 2002), Forward & Backward Selection, as well as the integrated selection methods found in some machine learning algorithms.

While the first two methods were computationally expensive due to their iterative nature of training and testing a model after each modification, they produced highly comparable results to the integrated methods. All methods ranked the seven engineered features among the top ten most important, even when the original features used for their creation were ranked as the least important, highlighting the importance of feature engineering in producing more significant splits at the tree level in algorithms such as the BRF.

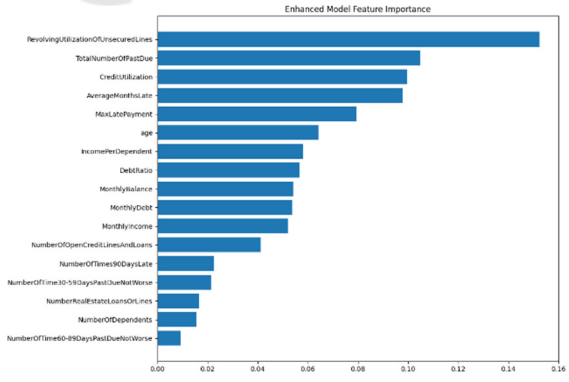

Figure 4: Showcasing the feature importance for the model Balanced Random Forest, highlighting the role of feature engineering in perfecting the machine learning results.

5 EXPERIMENTS

5.1 Performance Metrics

The selection of an adequate measure for performance is an important part of a data science project. It offers a mathematical foundation for comparison, assisting in the process of choosing the most effective model and guaranteeing its dependability in practical settings.

The most common performance measure for classification, i.e. accuracy, can be highly misleading in the context of imbalanced data. In the particular case of the GMC dataset defaulters are a small proportion of around 6% of the total population, the trivial model predicting that all of the samples belong to the non-defaulters class would obtain an accuracy of around 94%. Therefore, we decided to use the Area Under the Curve (AUC) as a performance measure, as it is recommended also by (Lessmann et al., 2015). Also, we have selected this metric in order to ensure consistency with the existing literature in the field, facilitating direct comparisons other similar studies.

The Receiver Operating Characteristic Curve (ROC) is drawn by calculating the True Positive Rate (TPR) and False Positive Rate (FPR) at N selected thresholds indexed by i. (The threshold selection in fact controls the model's ability to predict each class.) Let $x_i = \mathrm{FRP}_i$ and $y_i = \mathrm{TRP}_i$ be the values of the i-th selected threshold. Then the AUC represents the area under ROC and can be computed by the trapezoidal rule:

$$\mathrm{AUC} = \frac{1}{2}\sum_{i=1}^{N-1}(x_{i+1} - x_i)(y_{i+1} + y_i).$$

The model with greater AUC is generally considered to be the better one.

5.2 Model Selection and Tuning

Several algorithms were tested for the task of classifying defaulters and non-defaulters. These are:

- Random Forest (RF);
- Support Vector Classifier (SVC);
- Logistic Regression (LR);
- Extreme Gradient Boosting (XGBoost);
- Balanced Random Forest (BRF);
- Light Gradient Boosting Machine (LGBM);
- Dense Neural Networks (DNN);
- Deep Belief Networks (DBN);
- TabNet.

For tuning of the models' hyperparameters an extensive grid search was performed for each of the algorithms. As an example, tuning the Balanced Random Forest model involved the testing of 19,200 combinations of parameters utilizing a 5-fold cross-validation approach in order to identify the optimal settings and ensure model stability.

5.3 Experimental Results

In the following we describe the experiments done in order to improve the performance of credit scoring models on the GMC dataset. First we have established a baseline model which was chosen to be the RF model tuned with grid search and 5-fold cross-validation, without using any data pre-processing steps such as imputation, feature engineering or class balancing. This was set as a reference point for further enhancements.

Then we have incorporated some data pre-processing steps such as outlier removal and feature engineering. This has resulted in a small improvement of the AUC by 0.32%.

In the following step we have focused on evaluating the effect of the different data imputation methods (Simple Imputer, KNN Imputer and Iterative Imputer) across five algorithms: RF (Breiman, 2001), SVC (Hearst et al., 1998), Logistic Regression, XGBoost (Chen et al., 2016), and LGBM (Ke et al., 2017). The Iterative Imputer with LGBM achieved the highest improvement in AUC of 0.863, showing a quite significant effectiveness, while the other imputation methods generally did not improve performance.

Next, class balancing techniques were tested. Cluster-based down-sampling reduced the AUC significantly (but achieved high defaulter classification rates), while SMOTE slightly improved the AUC by 0.7%. In order to further identify effective data-level balancing strategies we have used a comprehensive grid search with several resampling methods, for example Tomek-Links. Algorithm-level balancing methods were compared against data-level methods and combined approaches. The Balanced Random Forest algorithm outperformed all other methods, achieving the highest AUC score and demonstrating strong handling of class imbalance. It is based on the original Random Forest algorithm, but during training it ensures that all classes are equally represented by randomly choosing a fixed number of samples from all classes, potentially selecting the same sample more than once. This algorithm also has an option for employing cost-sensitive learning,

which we have tested but this did not improve in anyway the results.

Finally, some classic neural network architectures like the Dense Neural Networks (DNN), Deep Belief Networks (DBN) and the new TabNet (Arik and Pfister, 2021) were added to the experiments. While the DNN and TabNet performed comparably to the tree based methods with an AUC around 0.860, DBN significantly underperformed, highlighting its limitations for this particular dataset.

The best results obtained for each algorithm are presented below in Table 1.

Table 1: Experimental Results.

ML algorithm	Balancing Method	AUC score
Base line RF	-	0.850
RF	U+TL*	0.860
SVC	-	0.770
LR	-	0.804
BRF	-	**0.875**
XGBoost	-	0.874
LGBM	ENN**	0.870
DNN	SMOTE	0.856
DBN	-	0.773
TabNet	SMOTE	0.860

Abbreviations above have the following meaning:
* U = Under-sampling, TL = Tomek-Links
** ENN = Edited Nearest Neighbours.

The ROC of the best performing model can be visualised and compared to TabNet in Figure 5 below.

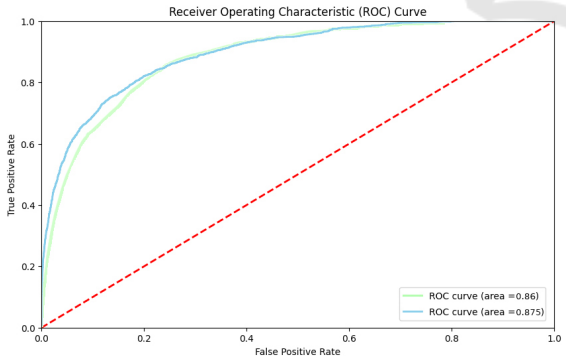

Figure 5: ROC of the BRF model (Blue) compared to TabNet (Green).

The final results of the experiments presented above underscore the importance in data science of careful data analysis, proper data pre-processing, balancing, fine-tuning, and the selection of methods tailored to each specific algorithm and dataset.

5.4 Comparison with Other Solutions

In Table 2 below we compare our solution to the GMC problem with other existing solutions from the literature.

Table 2: Comparison of our result with existing solutions.

Solution	Model	AUC score
(Lessmann et al., 2013)	Ensemble	0.865
(Chen, 2021)	RF	0.797
(Gunnarsson et al., 2021)	XGBoost	0.848*
(Dumitrescu et al., 2022)	PTLR	0.857
(Gidlow, 2022)	XGBoost	0.867
current	BRF	**0.875**
* Extracted from Figure 5 in (Gunnarsson et al., 2021)		

The reader should note a relative improvement compared to the previous best existing solution by approximatively 1% in the AUC score. Together, these articles contributed significantly to our project development, directed us through their previous knowledge and discoveries, and enabled us to achieve this new result reported here.

6 CONCLUSIONS AND FUTURE WORK

In this paper we study how data analysis and data pre-processing techniques can be effectively used for credit scoring, highlighting the potential impact of machine learning for analysing complex datasets and enhancing decision-making in the case of financial institutions.

The GMC dataset presents significant challenges such as outliers, missing data and class imbalance, necessitating robust data pre-processing in order to improve machine learning models classification performance. The Iterative Imputer proved to be the most effective imputation method from those tested. Experiments revealed that several algorithms struggle with class imbalance, while (for this particular dataset) the Balanced Random Forest algorithm outperformed all of them. Through our study we pursued a structured approach including various stages of testing and refinement. This emphasizes the importance of careful analysis in developing tailored models that effectively address data-specific challenges.

Our future work plan is to implement an ensemble system that combines the strengths of the BRF, XGBoost and LGBM in order to leverage the complementary strengths of each model, enhance

predictive performance and increase robustness. Such system may deliver superior performance as was previously remarked in (Lessmann et al., 2013) and (Lessmann et al., 2015).

REFERENCES

Arik, S., Pfister, T. (2021). TabNet: Attentive interpretable tabular learning. In *AAAI 2021, Proceedings of the 35th AAAI conference on artificial intelligence*, pages 6679 – 6687.

Breiman, L. (2001). Random forests. *Machine learning* 45, pages 5 – 32.

Chen, L. (2021). Statistical Learning for Analysis of Credit Risk Data. *IOSR Journal of Mathematics* 17, pages 45 – 51.

Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *KDD 2016*, *Proceedings of the 22nd ACM International Conference on Knowledge Discovery & Data Mining*, pages 785 – 794.

Chen, C., Liaw, A., Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California Berkeley, report number* 666, pages 1 – 12.

Dumitrescu, E., Hué, S., Hurlin, C., Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research* 297, pages 1178 – 1192.

Gidlow, L. (2022). The Effect of Dataset Size on the Performance of Classification Algorithms for Credit Scoring. *University of Cape Town, available at* http://hdl.handle.net/11427/37193.

Google Brain (2016). TensorFlow: A system for large-scale machine learning. In *OSDI 2016, Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, pages 265 – 283.

Gunnarsson, B. R., Vanden Broucke, S., Baesens, B., Óskarsdóttir, M., Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't?. *European Journal of Operational Research* 295, pages 292 – 305.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*, 389 – 422.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications* 13, pages 18 – 28.

Ivan, T. (1976) An Experiment with the Edited Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics* 6, pages 448 – 452.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30, pages 1 – 9.

Lessmann, S., Baesens, B., Seow, H. V., Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* 247, pages 124 – 136.

Lessmann, S., Seow, H., Baesens, B., Thomas, L. C. (2013). Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. *Credit Research Centre, Conference Archive.*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32, pages 8024 – 8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825 – 2830.

Tang, B., He, H. (2015). ENN: Extended nearest neighbor method for pattern recognition. *IEEE Computational Intelligence Magazine* 10, pages 52 – 60.