A comparative analysis of classification algorithms for consumer credits

Claudia ANTAL-VAIDA

Bucharest University of Economic Studies, Bucharest, Romania

claudia_antal@ymail.com

Abstract. *Machine Learning is a constantly growing area which has the capacity to analyze massive amounts of data and find relevant patterns, a very important feature in the era of big data. It has a wide range of application areas, including the financial field, and proved to be efficient in solving various problems, including the prediction of the default probability of a customer to meet their obligations to the bank, using classification algorithms. Their output is further used when deciding whether to approve a loan or no, based on the previous behavior of the customers, hence reduces the loss of the bank. Even though Machine Learning algorithms proved to be efficient in solutioning this type of problems, none was identified for remarkable results. This paper studies 10 different methods applied on the same dataset (Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Kernel Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Bagging Classifier, Linear Discriminant Analysis, Neural Network - Multi Layer Perceptron) and performs a comparative analysis aiming to identify the one which outperforms the others. Their performance is evaluated based on some well-known statistical measures such as Accuracy, Misclassification Rate, Precision and Specificity. In addition, this paper also presents and evaluates the impact of feature selection on the overall performance of an algorithm.*

Keywords: Machine Learning, Banking, Consumer Credits, Feature Selection, Algorithms Assessment.

Introduction

The Digital Era changed the way industries are working, but also how the leadership makes decisions (Doukidis et al., 2004). If in the past most of them were relying only on human judgement, the approach quickly changes and most realizes the importance of data and the positive impact of data-driven decisions.

The Banking sector is undergoing similar changes and needs to keep pace with the fast-growing technologies, including the adoption of business analytic techniques, in order to remain competitive on the market (Raghynathan & Maiya, 2018). It should be mainly interested in understanding customer's behavior, how to improve interactions with customers and what motivates them to carry through their obligations (Dwight, 2013), which can be simplified and improved thru machine learning applications. Banks are storing huge amounts of data about their customers, but without dedicated solutions, it is difficult to analyze and interpret them, hence close to impossible to leverage in the decision-making process. This would be only one example where machine learning can simplify, but at the same time, increase accuracy and bring value.

As one of the major activities of a bank is handling cash and credits, predicting the worthiness of a customer has a high significance in its overall business strategy. Though, this is still a major challenge within the banking sector because scoring a customer based on limited features is not an easy task to perform without the aid of technology. For this purpose, classification algorithms, supervised learning processes, can be applied; they create groups based on labeled observation and classify a customer by comparing the information about him with the characteristics of the groups.

As the success of a bank depends on its capacity to collect back the money from its customer, the main focus in the following part of this paper will be to identify the algorithm which performs best for classification problems, based on the past behavior of similar clients. Over the last few years, various traditional statistical methods and machine learning techniques were proposed and discussed, but all were applied on different datasets, which can influence the outcome. To overcome this aspect, this paper proposes a performance evaluation of 10 different algorithms (Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Kernel Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Bagging Classifier, Linear Discriminant Analysis, Neural Network - Multi Layer Perceptron) applied on the same dataset to reduce the influence of the dataset's quality. In addition, the paper studies 3 different methods of selecting the relevant features and assesses the impact of this step on the outcome of the models. Reducing the number of features can decrease the algorithm run-time, but also the time for collecting data, in case less information is required. Current paper represents a continuation of the previously published article, "Business Analytics Applications for Consumer Credits" (Antal-Vaida, 2020), and includes additional conclusions and outlines the experimental results of the author's own research.

Literature review

There are various Machine Learning techniques which have been developed over the years, but none have been identified as being the best benchmark in any of the industries. Focusing on our area of research, scoring models for consumer credits, various papers published after 2008 were reviewed to better understand the classification methods, a widely used technique for scoring credits in the risk management field. In the following part of this paper, the main discoveries and conclusions of the literature review will be summarized.

Support Vector Machines (SVMs) is a technique which implies 3 elements: a linear combination of features, an objective function which evaluates both the training and the testing set, and an algorithm used to identify the most optimal parameters. This method focuses on the outliers, which are closer to the opposite limits, and proved to perform especially on classification problems and feature selection for determining the default probability of a client (Huang et al., 2007; Bellotti & Crook, 2009; Keramati & Yousefi, 2011; Zhou & Wang, 2012; Harris, 2013; Lessman et al., 2015; Ha & Nguyen, 2016). A 2009 study presented a hybrid model of SVMs and genetic algorithms, which outperformed on identifying relevant feature selection, but also on optimizing the model parameters. It also confirmed the previous findings, and more than that, it outlined the fact that the credit type can majorly contribute to the identification of the relevant features (Huang et al., 2007). In 2013, a paper evaluated this technique from two different perspectives: a restrictive one, considering the credits with a delay under 90 days, and an extensive one, analyzing the credits with a delay over 90 days. The later approach provided better predictions, especially because of the higher number of analyzed cases (Harris, 2013). Even though most of the articles outlined good performances, some downsides were identified as well: extensive training time, high complexity and extensive computational power, the „black-box” behavior, making it almost impossible to trace back how a certain result was reached (Huang et al., 2007), but also the need for a high number of SVMs for good results (Bellotti & Crook, 2009).

Random Forest is a classification and regression method which uses a group of decision trees for training, resulting in a class which is the mode of classes when applied for classification problems, and the median of the trees' prediction, when applied for regressions. A 2012 paper outlined an analysis using weighted decision trees, calculated based on their previous performance

and on the training errors (Zhou & Wang, 2012). This approach outperformed not only the standard random forest algorithm, but also the results obtained thru Support Vector Machines (SVMs) and K-NN (K-Nearest Neighbor). Various studies outlined a great advantage of this method, which is the reduced training time, due to the parallel run (Zhou & Wang, 2012; Lessman et al., 2015; Ha & Nguyen, 2016; Wang et al., 2015; Hamori et al., 2018).

K-Nearest Neighbor is a nonparametric classifier which learns from the similarities between classes. It uses a distance function for the relevant features and once a new record is added to the model, it analyzes its pattern and compares it with the nearest neighbors, adding it to the most similar class. (Bellotti & Crook, 2009; Keramati & Yousefi, 2011; Zhou & Wang, 2012; Lessman et al., 2015; Ha & Nguyen, 2016).

Artificial neural networks are nonlinear statistical models, which are working very similar to the human brain. These are well performing on data analysis where relationships are unknown. It was documented that it has the capacity to identify complex patterns and make predictions for new independent data (Keramati & Yousefi, 2011; Lessman et al., 2015; Hamori et al., 2018). A paper published in 2008 emphasized the high interdependency between its performance, the choice of the activation function and the number of hidden layers used.

Logistic regression is a type of linear regression which does not require linearity between the independent variables, while the dependent ones do not require a normal distribution. This method can be applied regardless the types of variables: discrete, continuous or binary. This algorithm proved to perform for classification problems but having similar results to others which are easier to apply (Bellotti & Crook, 2009; Keramati & Yousefi, 2011; Wang et al., 2015).

Discriminant Analysis is an alternative to logistic regression and starts from the premises that the relevant features have a multivariate distribution, mainly focusing on reducing the distance between the members of the class and increasing the distance between classes (Bellotti & Crook, 2009; Keramati & Yousefi, 2011; Lessman et al., 2015).

Other algorithms which were reviewed were Lasso Logistic regression (Wang et al., 2015), Decision Trees (Keramati & Yousefi, 2011; Lessman et al., 2015; Ha & Nguyen, 2016; Wang et al., 2015), Bagging (Lessman et al., 2015; Ha & Nguyen, 2016; Hamori et al., 2018), Boosting (Hamori et al., 2018), (which outperformed SVMs, Neural Networks and Random Forest), Multi Layer Perceptron (Lessman et al., 2015; Ha & Nguyen, 2016) and Deep Learning (Ha & Nguyen, 2016; Hamori et al., 2018; Yu et al., 2016), which proved to be very powerful, offering stability in training, scalability for large sets of data and ability to generalize. The later method exceeded the benchmark methods previously used for classification problems.

All these articles addressed the same topic – machine learning algorithms used for consumer credits – and all of them performed well, especially when compared with traditional statistical methods. However, it is difficult to recommend only one, because the analysis was performed on different datasets and could have been influenced by various variables. It should also be considered that the results are highly dependent on the quality of the input, its diversity and variety, covering as many scenarios as possible.

Methodology

Classification algorithms make predictions about the associations of an object to a class, considering various features. For learning how to perform a correct classification, they need a training set to determine the characteristics of a class, and a test set to check its accuracy. To create such a model, there are couple of steps to be followed, represented in Figure 1, which will be demonstrated in the next part of this paper.

Figure 1. Building a predictive model

Source: Authors' own research.

Step 1: Collecting relevant data which will later feed the model

Step 2: Data analysis:

- Checking if null values exist in the dataset. If yes, those can be replaced with the mean or the median of the variable.
- Data Scaling is mainly required to avoid the situations when the high numbers of a certain attribute can overlay the others which could have a higher significance. It mainly consists in converting all the attributes to the same scale and can be done thru:
 - Normalization – scales the variables to fit between 0 and 1;
 - Standardization – transforms the variables to a mean of 0 and a standard deviation of 1.

Step 3: Selecting the relevant features – this step is optional, especially when the dataset does not have a high number of variables;

Step 4: Splitting the dataset into a training set and a testing set;

Step 5: Selecting the model(s);

Step 6: Training the algorithm;

Step 7: Testing the algorithm;

Step 8: Evaluating the algorithm;

Step 9: Improving the algorithm.

In the following part of this paper all the steps will be applied one by one to better understand how to perform such an analysis and the results will be reviewed.

Experimental evaluation

The dataset used for this research was posted by I-Cheng Yeh in UCI Machine Learning Data repository (Yeh, 2016), and contains 30,000 observations and 25 columns (one identifier, 23 dependent variable and a dependent variable), described in Table 1. For performing the analysis, Python was used for both feature selection but also for running the algorithms.

Table 1. Dataset description

Attribute	Description	Type
ID	Unique identifier of the client	Categorical (Nominal)
LIMIT_BAL	Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.	Numeric
SEX	Gender (1=male; 2=female)	Binary
EDUCATION	Education (1= graduate school; 2 = university; 3=highschool; 4=others; 5=unknown)	Categorical
MARRIAGE	Marital status (1=married; 2=single; 3=other)	Categorical
AGE	Age	Numeric

Attribute	Description	Type
PAY_1	The repayment status in September 2005 (-1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.)	Categorical
PAY_2	The repayment status in August 2005	Categorical
PAY_3	The repayment status in July 2005	Categorical
PAY_4	The repayment status in June 2005	Categorical
PAY_5	The repayment status in May 2005	Categorical
PAY_6	The repayment status in April 2005	Categorical
BILL_AMT1	Amount of bill statement in September, 2005	Continuous
BILL_AMT2	Amount of bill statement in August, 2005	Continuous
BILL_AMT3	Amount of bill statement in July, 2005	Continuous
BILL_AMT4	Amount of bill statement in June, 2005	Continuous
BILL_AMT5	Amount of bill statement in May, 2005	Continuous
BILL_AMT6	Amount of bill statement in April, 2005	Continuous
PAY_AMT1	Amount paid in September, 2005	Continuous
PAY_AMT2	Amount paid in August, 2005	Continuous
PAY_AMT3	Amount paid in July, 2005	Continuous
PAY_AMT4	Amount paid in June, 2005	Continuous
PAY_AMT5	Amount paid in May, 2005	Continuous
PAY_AMT6	Amount paid in April, 2005	Continuous
default.payment.next.month	Will pay next month? (1= yes; 0= no)	Binary (Categorical)

Source: <https://archive.ics.uci.edu>.

First and foremost, we need to check the occurrence of null values in the dataset as it may cause errors in training. When identified, two steps can be followed:

- Deleting the observation, which is not recommended, especially when the dataset is not very large as valuable information could be removed.
- Replacing the null variables with the mean or the median of that specific attribute, hence removing the impact on the output.

The dataset we use does not have null variables, as highlighted in Figure 2, hence no action is required.

```
In [3]: total = dataset.isnull().sum().sort_values(ascending = False)
...: percent = (dataset.isnull().sum()/dataset.isnull().count()*100).sort_values(ascending = False)
...: pd.concat([total, percent], axis=1, keys=['Total', 'Percent']).transpose()
Out[3]:
```

	default.payment.next.month	PAY_6	LIMIT_BAL	...	PAY_AMT4	PAY_AMT5	ID
Total	0.0	0.0	0.0	...	0.0	0.0	0.0
Percent	0.0	0.0	0.0	...	0.0	0.0	0.0

Figure 2. Checking the existence of null values

Source: Authors' own research/

Relevant feature selection is one of the main concepts applied in Machine Learning and can highly influence the quality and accuracy of the output; the non-relevant features can negatively

impact the performance and efficiency of the model. This step aims to identify the most optimal set of variables which has the highest impact on the dependent variable. The advantages of performing it are:

- Avoids overloading the model – considering only the most significant features will positively impact the performance of the model;
- Improves accuracy – reducing the number of variables will force the model to focus only on the ones which have a real impact on the output;
- Reduces the training time.

There are various methods for feature selection, couple of them being documented below:

1. **Univariate selection** – statistical tests can be used for selecting the features which are highly correlated with the dependent variable. For performing such tests, the scikit-learn library (Pedregosa, et al., 2011) from Python can be used; it offers various statistical tests to select a specific number of variables. For the current analysis, `f_classif`, a function which uses ANOVA tests, was used to pick the first 10 variables with the highest impact on the model and the results are presented in Figure 3.

Figure 3. Top 10 most impactful variables based on Univariate selection

Source: Authors' own research.

Based on this test, the most impactful variables are the statuses of the payments in the past 6 months, followed by the total credit amount and the amount of payments on September, August and June. This method takes into consideration only the numerical variables, hence the categorical ones were not included.

2. **Predefined methods for evaluating the importance of the variables** – these are iterative methods and associate a score to each variable, the maximum value being given to the attributes with the highest impact on the dependent variable.

2A. Extra Tree Classifier – this method introduces an estimator which runs several decision trees on various data subsets, improving based on their mean. The result obtained with this method is presented in Figure 4.

Figure 4. Top 10 most impactful variables based on the Extra Tree Classifier method

Source: Authors' own research.

Based on the Extra Tree Classifier method, the most influential variables are the repayment status in September, the Age, the total amount of credit, the amount of bill statement in September, the repayment status in August, and the amount of bill statement on August, July, April and May and the amount paid in April.

2B. Lasso regularizations – this method applies penalties on the insignificant variables, converting their coefficient to 0 and excludes them from the model. The results are presented in Figure 5.

Figure 5. The most impactful variables based on Lasso CV method

Source: Authors' own research.

According to this method, the dependent variable is majorly influenced by the amounts paid during the analyzed timeframe, the total amount of credit and the amount of bill statements over the analyzed period.

3. The Correlation matrix shows the correlation between the independent variables, but also their correlation with the dependent variables. The relationship can be positive, meaning that when the independent variable increases, the dependent one will increase too, or a negative

relationship, meaning that whenever the independent variable increases, the dependent one will decline.

Figure 6. Correlation Matrix

Source: Authors' own research.

The output of the matrix is represented in figure 6, but having 24 variables in place, increases the difficulty of reading it. Though, the most impactful variables are the total amount of credit (negative relationship), and the payment status over the analyzed period (positive correlation).

Before selecting the algorithms, the dataset needs to be split into a training set and a test set. In order to achieve it, the `train_test_split` function from the Python sklearn library (Pedregosa, et al., 2011) was applied and will result in two sets: a training one, representing 67% of the observations (20 100), and a testing one, representing 33% (9 900).

After the split, ten different algorithms were evaluated in order to see which one performs better. All 23 variables were taken into consideration, and the results are summarized in Table 2.

Table 2. Evaluation of the algorithms with 23 variables

Algorithm	Accuracy	TP	FP	FN	TN
Logistic Regression	81.66%	7591	174	1642	493
K-NN (15 neighbors)	81.23%	7349	416	1442	693
SVM	81.60%	7572	193	1629	506
Kernel SVM	82.44%	7438	327	1411	724
Naïve Bayes	59.38%	4267	3498	523	1612
Decision Tree	73.78%	6410	1355	1241	894
Random Forest (50 estimators)	81.88%	7351	414	1380	755
Bagging Classifier (Kernel SVMs)	82.35%	7443	322	1425	710
Linear Discriminant Analysis	81.74%	7566	199	1609	526
Neural Network - Multi-Layer Perceptron	82.46%	7339	426	1310	825

Source: Authors' own research.

where:

- TP (True Positive) represents the number of cases where the prediction of the model was correct, the client paying his debts on time;
- FP (False Positive) – also known as Type I Errors, represents the number of cases when the prediction indicated that the client would meet his obligations, but it did not actually happen;
- FN (False Negative) – known as Type II Errors, represent the number of cases the model predicted that a client will not meet his obligations, but he actually did;
- TN (True Negative) – outlined the number of cases when the prediction aligned with the facts, and the client did not meet his obligations.

As outlined in Table 2, most of the algorithms achieved an accuracy over 80%, the only ones underperforming being Naïve Bayes and Decision Trees. However, even if the overall accuracy of Naïve Bayes was under 60%, it is noticeable the increased prediction of True Negative cases, indicating a high accuracy of identifying the clients which would not meet their obligations.

For the second part of the analysis, the number of independent variables was reduced to 10 using the Extra Tree Classifier method, previously tested in this paper. Therefore, the variables taken into account in the next part of this paper were the repayment status in September (PAY_1) and August (PAY_2), the age (AGE), the amount of the given credit (LIMIT_BAL), the amount of bill statement in September (BILL_AMT1), August (BILL_AMT2), July (BILL_AMT3), May (BILL_AMT5) and April (BILL_AMT6), the amount paid during the first month of the analysis, April (PAY_AMT6), and the default payment of the next month (default.payment.next.month). The results obtained are highlighted in Table 3.

Table 3. Evaluation of the algorithms with 10 variables

Algorithm	Accuracy	TP	FP	FN	TN
Logistic Regression	81.65%	7586	179	1638	497
K-NN (15 neighbors)	82.09%	7350	415	1358	777
SVM	81.60%	7572	193	1629	506
Kernel SVM	82.61%	7492	273	1449	686
Naïve Bayes	81.49%	7110	655	1177	958
Decision Tree	73.14%	6350	1415	1244	891
Random Forest (50 estimators)	82.11%	7328	437	1334	801
Bagging Classifier (Kernel SVMs)	82.60%	7491	274	1449	686
Linear Discriminant Analysis	81.64%	7556	209	1609	526
Neural Network - Multi Layer Perceptron	82.56%	7365	400	1327	808

Source: Authors' own research.

The results of the analysis with 10 variables improved, 9 out of 10 algorithms obtaining an accuracy over 80%.

Results and discussions

In the previous part of this paper, 10 different classification algorithms were tested on the same dataset, first run with all independent variables included and a second run considering only the top 10 most impactful variables, mainly to outline the impact of the feature selection process on the performance of the algorithm. The results are consolidated in Table 4.

Table 4. Comparative analysis of the algorithms

Algoritm	Accuracy – 23 variables	Accuracy – 10 variables	Delta
Logistic Regression	81.66%	81.65%	-0.01%
K-NN (15 neighbors)	81.23%	82.09%	0.86%
SVM	81.60%	81.60%	0.00%
Kernel SVM	82.44%	82.61%	0.16%
Naïve Bayes	59.38%	81.49%	22.11%
Decision Tree	73.78%	73.14%	-0.64%
Random Forest (50 estimators)	81.88%	82.11%	0.23%
Bagging Classifier (Kernel SVMs)	82.35%	82.60%	0.24%
Linear Discriminant Analysis	81.74%	81.64%	-0.10%
Neural Network - Multi Layer Perceptron (MLP)	82.46%	82.56%	0.09%

Source: Authors' own research.

The limited number of variables insignificantly improved the accuracy of the algorithms in most of the cases, while in 3 cases the impact was negative, the accuracy suffering a decrease. The only case where significant improvements were noticed was for Naïve Bayes, where the accuracy increased by 22pp.

Considering that the second run, which included a subset of the variables, significantly increased the accuracy for one of the algorithms, we will further use it to establish which one performed best. In order to measure it, the key performance indicators outlined in Table 5 will be calculated, while the results are summarized in Table 6.

Table 5. Key performance indicators for model evaluation

KPI	Formula	Description
Accuracy (calculated in Python)	$\frac{TP + TN}{Total\ number\ of\ cases}$	Represents the ratio of the cases when the prediction was correct.
Misclassification Rate	$\frac{FP + FN}{Total\ number\ of\ cases}$	Represents the ration of the cases when the prediction was wrong.
Precision	$\frac{TP}{TP + FP}$	Shows the percentage of cases when the prediction of positive cases was correctly identified.
Specificity	$\frac{TN}{FP + TN}$	Shows the percentaje of negative cases which were correctly identified.

Source: <https://www.ritchieng.com>.

Table 6. Algorithms evaluation

Algoritm	Accuracy	Misclassification Rate	Precision	Specificity
Logistic Regression	81.65%	18.35%	97.69%	73.52%
K-NN (15 neighbors)	82.09%	17.91%	94.66%	65.18%
SVM	81.60%	18.40%	97.51%	72.39%
Kernel SVM	82.61%	17.39%	96.48%	71.53%
Naïve Bayes	81.49%	18.51%	91.56%	59.39%
Decision Tree	73.14%	26.86%	81.78%	38.64%
Random Forest (50 estimators)	82.11%	17.89%	94.37%	64.70%
Bagging Classifier (Kernel SVMs)	82.60%	17.40%	96.47%	71.46%
Linear Discriminant Analysis	81.64%	18.36%	97.31%	71.56%
Neural Network - Multi Layer Perceptron (MLP)	82.56%	17.44%	94.85%	66.89%

Source: Authors' own research.

Kernel SVM had the highest accuracy, followed by Bagging Classifier (Kernel SVMs) and Neural Network - Multi Layer Perceptron (MLP). Although the logistic regression did not have the highest accuracy, it achieved the highest Precision and Specificity, outlining its capacity to properly identify true positive and true negative cases; second in the row would be the SVMs.

It is difficult to point out one algorithm which performed best, because 9 out of 10 achieved an accuracy over 80%, hence, whenever such an analysis is required, multiple algorithms should be evaluated in order to assess their performance on a given dataset.

Conclusion

This paper presents a comparative analysis of 10 different machine learning algorithms used for one of the most common issues the banking system is facing: identifying the default probability of a customer to meet his obligations in the upcoming month. Even though this topic was addressed in other papers too, the main purpose of this research was to analyze the performance of machine learning algorithms when executed on the same dataset in order to reduce the impact of the dataset quality on the outcome, and to assess the possible impact of feature selection on the behavior of the algorithms. The first part of this paper considered all the variables in the dataset and 8 out of the 10 algorithms achieved an accuracy over 80%, while the second run considered only a subset, representing the top 10 variables which have the highest influence on the dependent variable. The later approach significantly improved the accuracy only for one of the algorithms, Naive Bayes, which increased by 22 pp. In both approaches, the least performant algorithm was the decision tree method, for which accuracy was below 80%. First withdrawn conclusion is in line with the results of other studies and outlines that there is no algorithm which necessarily outperforms the others, hence multiple methods should be tested for evaluating which provides better results on a specific dataset.

The same analysis targeted to assess the impact of selecting the most relevant features on the algorithm performance. Even though multiple methods were tested (univariate selection, predefined methods and correlation matrix), the selection was made based on Extra Tree Classifiers and analyzed the impact on the model accuracy, which led to a significant increase only for one of them, Naive Bayes. For the others, the impact was insignificant, in 3 out of the 10 cases outlining a small negative impact. The dataset had only 23 independent variables, but in real life cases, feature selection could significantly improve the overall performance: not only the accuracy, but could also reduce the training time, avoid overfitting the model and could simplify collecting the data, focusing only on the ones which are highly impacting the dependent variable. To conclude, although Ha & Nguyen (2016) highlighted improvements in accuracy when applying the feature selection, the current experimental results did not reflect the same, significant enhancements being observed only for one of the analyzed algorithms.

In conclusion, none of the algorithms can easily become a benchmark for classification problems, but the ones with a high potential would be the Kernel Support Vector Machines and the Bagging Classifier (with Kernel Support Vector Machines), even though Hamori et al. (2018) did not outline outstanding results for the later. In addition, even if the logistic regression did not stand out in terms of accuracy, as observed in the current study, but also as previously noted by Lessman et al. (2015), it achieved the highest Precision and Specificity, outlining its capacity to correctly identify true positive and true negative cases. For the upcoming research, the aim is to assess different enhancement opportunities to increase the accuracy and the performance of the algorithms for classification problems applied in the banking sector.

References

- Antal-Vaida, C. (2020). Business Analytics Applications for Consumer Credits, *Database System Journal*, 14-23.
- Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *ScienceDirect - Expert Systems with Applications*, 36, 3302-3308.
- Doukidis, G., Mylonopoulos, N., & Pouloudi, N. (2004), *Social and Economic Transformation in the Digital Era*, IGI Global.
- Dwight, M. (2013). *A framework for Applying Analytics in Healthcare – What can be Learned from the Best Practices in Retail, Banking, Politics and Sports*, Pearson Education Inc.
- Ha, V.-S., & Nguyen, H.-N. (2016). Credit scoring with a feature selection approach based deep learning. *MATEC Web of Conference*. 54. Cape Town, South Africa: EDP Sciences.
- Hamori, S., Kawai, M., Kume, T., Murakami, Y., & Watanabe, C. (2018). Ensemble Learning or Deep Learning? Application to Default Risk Analysis, *Journal of Risk and Financial Management*.
- Harris, T. (2013). Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions. *Elsevier - Expert Systems with Applications*, 40, 4404-4413.
- Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines, *ScienceDirect - Expert Systems with Applications*, 33, 847-856.
- Keramati, A., & Yousefi, N. (2011). A Proposed Classification of Data Mining Techniques in Credit Scoring, *International Conference on Industrial Engineering and Operations Management*, 416-424, Kuala Lumpur, Malaysia.
- Lessman, 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*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 2825-2830, Retrieved from Scikit learn: <https://scikit-learn.org/stable/>.
- Raghynathan, B., & Maiya, R. V. (2018). *SMACing the Bank - How to Use Social Media, Mobility, Analytics, and Cloud Technologies to Transform the Business Processes of Banks and the banking Experience*, CRC Press.
- Wang, H., Xu, Q., & Zhou, L. (2015). Large Unbalanced Credit Scoring Using Lasso-Logistic Regression Ensemble, *PloS Ones*.
- Yeh, I.-C. (2016). *UCI Machine Learning Repository*, Retrieved from <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>.
- Yu, L., Yang, Z., & Tang, L. (2016). A Novel Multistage Deep Belief Network Based Extreme Learning Machine Ensemble Learning Paradigms for Credit Risk Assessment, *Flexible Services and Manufacturing Journal*, 28.
- Zhou, L., & Wang, H. (2012). Loan Default Prediction on Large Imbalanced Data Using Random Forests. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 10, 1519-1525.