Probability of Loan Default – Applying Data Analytics to Financial Credit Risk Prediction

Aleksandra Łuczak[1], Maria Ganzha[1], Marcin Paprzycki[2]

Abstract In the banking industry, one of the important issues is how to establish credit worthiness of potential clients. With the possibility of collecting digital records of results of past credit applications (of all clients), it can be stipulated that machine learning techniques can be used in "credit decision support" systems. There exists a substantial body of literature devoted to this subject. Moreover, benchmark datasets have been proposed, to establish effectiveness of proposed credit risk assessment approaches. The aim of this work is to compare performance of *seven* different classifiers, applied to *two* different benchmark datasets. Moreover, capabilities of, recently introduced, methods for combining results from multiple classifiers, into a meta-classifier, will be evaluated.

1 Introduction

Banking industry of today collects "all possible" data concerning all of its customers. Moreover, for all practical purposes, such data is never discarded (even if clients close their accounts [27]). One of the areas, where collected data is expected to be of great value, is to provide support for the, so called, credit risk assessment. Specifically, it is assumed that application of data analytics methods, to the collected data, can help correctly assess probability of loan defaults. This belief can be observed, among others, in a large body of literature devoted to the subject (see, Section 3). However, existing work is, mostly, focused on specific classifiers, with the goal of improving their *individual* performance. The aim of our work is to *compare* performance of different credit risk assessment methods, when applied to two stan-

Aleksandra Łuczak & Maria Ganzha
Warsaw University of Technology, Warsaw, Poland, e-mail: `M.Ganzha@mini.pw.edu.pl`

Marcin Paprzycki
Systems Research Institute Polish Academy of Sciences, Warsaw, Poland, e-mail: `marcin.paprzycki@ibspan.waw.pl`

dard “benchmark datasets”. Moreover, we will investigate usefulness of a, recently introduced, meta-classifiers, which “merge” predictions of individual classifiers to provide a “combined score”.

To this effect, we proceed as follows. We start from a brief summary of necessary background knowledge related to the banking industry (Section 2). Next, in Section 3, we present an overview of related literature. We follow, in Sections 4 and 5, with an overview of the two datasets used in our work, and the experimental setup. This allows us, in Section 6, to summarize results of performed experiments. We conclude our contribution is Section 7.

2 How banks assess customers – short introduction

In all banks, there exist complicated procedures, used to comprehensively evaluate credit worthiness of their (potential) customers. Moreover, each bank has it own method to do so, and such methods are a closely guarded intellectual property. While one of the authors worked in a credit department of a large bank, for obvious reasons, we will be able to share only “common open knowledge/information” about the process of assessment of probability of loan default. Keeping this in mind, let us start from basics of credit risk assessment.

2.1 Need of Credit Risk Assessment

One of the most sensitive stages, of a bank processing a credit application, is the decision-making process. Following the digital transformation, banks introduced online lending and borrowing mechanisms. Here, the key element is a semi-automatic credit decision process, which is based, among others, on the demographic and the financial data. Here, the *semi-automatic process* means, usually that automatic positive decisions are accepted, while possible “problems” are remanded to human employees for final decisions.

The possibility of applying for loans online, is one of the reasons that banks are granting more and more credits. Here, note that, at the same time, the total number of physical bank branches is systematically decreasing (as predicted, for instance, in [28]). The fact that the increase in number of credits is a global phenomenon is confirmed, among others, by the Polish Financial Supervision Authority [1]. According to the available data, at the end of December 2019, household debt, to the banks, amounted to 671 billion PLN. In addition, as much as 15.44 million, or 40.6% of Poles were actively repaying a loan, or a credit.

With the increasing prevalence of loans, and the decreasing requirements of banks (which earn money on the credits), there is also a recognized risk of approving loans to persons who cannot cope with their repayment. Hence, the need to, automatically (as the total number of bank employees is decreasing) evaluate credit risk, of an increasing number of loan applicants.

In this context, the *Probability of Default* (PD), indicates the probability of a loss, associated with a given credit. It is, usually, considered within a certain time-horizon (usually one year). Simply said, high value of PD indicates that customer may stop paying back the loan (temporarily, or permanently). The probability of customer's load default is the primary parameter in calculating creditworthiness. This factor plays a major role in loss control, and income maximisation [15]. PD can be estimated with the use of a number of analytical tools, and majority of such tools are based on some form of, broadly understood, "data analytics". In practice, in majority of banks, linear discriminate analysis [7], or logistic regression [36], are applied, because these algorithms combine simplicity and efficiency, and are easy to explain (high interpretability) [19]. The latter makes then also slightly more desirable than neural network based approaches, which are not explainable and thus raise variety of ethical issues.

Separately, in scientific literature, a number of approaches have been studied. Moreover, to support this research, at least two open, tagged, datasets have been created, which can be used to compare performance of considered methods.

3 Related Work

This being the case, let us now briefly present state-of-the-art of analytic credit risk assessment, as represented in scientific publications. Since, as mentioned above, each bank has its own credit risk assessment procedures, which are a closely guarded intellectual property, only published literature can be discussed. We split the material into two parts. First, we summarize the proposed approaches, dataset(s) that they were applied to, and the reported quality of the loan default prediction. The latter will be used to compare results of our experiments to. Second, for the standard methods of data analytics, described in the reviewed literature, we provide short descriptions and basic references.

3.1 Data Analytics for Credit Risk Analysis

In the credit risk assessment, the most common method used to predict the PD, is a scoring card [37]. It consists of assessing (weighting) selected characteristics of the potential borrower. After assigning a score to each one of them, these ratings are combined (summed up). Here, if the total score is higher than the (financial institution dependent) cut-off point, client receives a credit [32]. This is a very simple method and is very easy to operationalize, but has many disadvantages. For example, it has a relatively weak predictive power, and cannot handle large datasets, and features with complicated relationships [24].

Nowadays, more focus is devoted to the predictive power of classification models, used to control credit risk, because even a small increase in performance results results in a large increase in profits [2, 4]. Therefore, with the development of machine learning methods, the improvement of the predictive power of scoring models is within reach of financial institutions, and they are taking up this approach [9, 30, 38]. In most cases, algorithms from the supervised ML family are used in the credit risk assessment, to find a correlation between the clients attributes, and their expected loan repayment. Here, note that in many articles, proposed approaches are confirmed by high accuracy [8, 31, 33].

Recently, many researchers have focused their attention on ensemble methods, and use of different base classifiers, to make the accuracy of prediction "as high as possible". Such heterogeneous classifiers can, for instance, ensemble three state-of-the art classifiers: logistic regression (LR), artificial neural network (ANN), and support vector machines (SVM) [3]. Hybrid ensemble methods were also based on Random Forest, Support Vector Machines, Decision Trees, Artificial Neural Network, Multidimensional Neural Network [5]. The method based on a hybrid associative memory with translation was reported in [16]. There exists a study, from 2015, that compares 41 different classification algorithms [26] and a study that compares Bagging DT, Random Subspace DT, Random Forest and Rotation Forest [35]. All this research shows good model performance, as measured in terms of "standard" metrics such as accuracy (ACC) and area under the ROC curve (AUC), which are above 0.7.

Nevertheless, in this contribution, we will compare obtained results to those reported in [6, 34]. This is related to the fact that they use similar datasets. Moreover, they follow the same general methodology (see, section 5). Nevertheless, we recognize the fact that a more broad comparative study (similar to that reported in [26], but aligned with our investigation) could be of value.

3.2 *Classifiers used in our work*

Let us now summarize the specific machine learning approaches that have been used in the context of credit risk assessment.

3.2.1 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a classifier, which is very often used in pattern recognition [18], and to build credit scoring models [25]. It is a simple algorithm, based on adding new elements to the class that they are "best fitting", on the basis of a process of comparing it to the nearest set of observations (k-nearest neighbours).

The typical distance function is an Euclidean distance. In our work, we used the KNN modules available in *sklearn.neighbors.KNeighborsClassifier* [1].

3.2.2 Support Vector Machines

Support Vector Machines (SVM) is an ML method proposed by Vapnik [17]. It is based on the search for a hyperplanes separating classes. Established hyperplanes should be such that the closest observations from each class are as far away from each other as possible. In our work we have experimented with the SVM in two versions: with (a) linear, and (b) radial kernel functions. Here, we used *sklearn.svm.LinearSVC* [2] and *sklearn.svm.SVC* [3] modules.

3.2.3 Random Forest (RF)

The Random Forest (RF) algorithm was proposed by Breiman [13, 12]. It is based on the decision tree, and the classification committee. This means that the final response of the algorithm is selected as the average response of each decision tree (seen as an "independent classifier"). For the random forest we used *sklearn.ensemble.RandomForestClassifier* [4] modules.

3.2.4 Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (GBDT) is a generalization of boosting [29] to an arbitrary differential loss functions [22]. GBDT is an accurate and effective off-the-shelf procedure that can be used for both the regression and the classification problems. It can be applied in a variety of areas. To experiment with the GDBT we used *sklearn.ensemble.GradientBoostingClassifier* [5] modules.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html\#sklearn.neighbors.KNeighborsClassifier`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html?highlight=svm\#sklearn.svm.LinearSVC`

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html\#sklearn.svm.SVC`

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random\%20forest\#sklearn.ensemble.RandomForestClassifier`

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html\#sklearn.ensemble.GradientBoostingClassifier`

3.2.5 AdaBoost

An AdaBoost is an adaptive boosting algorithm, proposed by Freund and Schapire [20]. This is a meta-estimator that begins by fitting a classifier into the original dataset and then fits additional copies of the classifier on the same dataset However, in this case, the weights of incorrectly classified instances are adjusted, such that the subsequent classifiers focus more on difficult cases. In case of AdaBoost, we used *sklearn.ensemble.AdaBoostClassifier* [6] modules.

3.2.6 Extrene Gradient Boosting

The Extrene Gradient Boosting (XGBoost) algorithm is one of the most popular, and most effectively implemented algorithms, in the family of algorithms based on the gradient boosting method [21]. For the XGBoost we used *xgboost.XGBClassifier* [7] modules.

3.2.7 Meta-classifiers

While the above listed approaches can be (and have been) applied individually to credit risk assessment (see, Section 3.1), recently it has been realized that each one of them captures “separate aspects” of the data. The question has thus been posed: is it possible to combine results from multiple classifiers to develop a meta-classifier, which will improve the overall quality of prediction? Such meta-classifiers have been tried, with moderate success, in non-financial domains (see, for instance, [23, 14]). However, they involve relatively complex methods that require a lot of attention to be correctly implemented.

Searching in the same direction, recently, a different class of simple meta-classifiers have been proposed. Specifically, Weighted Average Recall Error (WARE), and Weighted Average Type-1 Error (WA-T1E) algorithms, are based on an easy to implement consensus models. These classifiers are similar to the Weighted Average Prediction Error (WA-PE) method used in [11]. Here, the data sample is divided into two parts, one for training the classifiers, and one to calculate the Recall $Recall_m$, and the Type I error $T1err_m$ (where, m is the index of the classifier). After normalizing the weights, for the $Recall_m$ and the $T1err_m$, posteriori probabilistic classes of observations are computed as follows:

$$WARE_1(x) = \sum_{m=1}^{M} Recall_m p_1^{h_m}(x) \tag{1}$$

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html?highlight=adaboost\#sklearn.ensemble.AdaBoostClassifier

[7] https://xgboost.readthedocs.io/en/latest/python/python_api.html\#module-xgboost.sklearn

$$WA - T1E_1(x) = \sum_{m=1}^{M} T1err_m p_1^{h_m}(x) \tag{2}$$

where p_1 is the posteriori probability of belonging to class 1 and h_m is a classifier. Here, observation x is assigned to class 1 if $WARE_1(x) > WARE_0(x)$, or to class 0 otherwise. The same applies to the WA-T1E method, where the observation x is assigned to class 1 if $WA - T1E_1(x) > WA - T1E_0(x)$, or to class 0 otherwise.

Therefore, taking into account their simplicity, we have decided to experiment also with these two meta-classifiers, to see if their application can improve the overall performance of the credit risk estimators.

4 Datasets used in experiments

Let us now move our attention to the data used in our experiments. As mentioned above, there exist multiple tagged datasets that can be used to evaluate performance of approaches to credit risk prediction. In our work we have used two of them: (a) German Credit [8], and (b) Give Me Some Credit [9]. Both datasets contain information about the demographic characteristics and the financial situation, of potential borrowers. Let us now describe each one of them in more detail.

4.1 German Credit dataset

The German redit data contains 1 000 000 observations, representing two classes: good creditor (699 774 observations), and bad creditor (300 226 observations). The data set is described by 20 features, including 7 continuous features.

- *duration* – Duration in month
- *credit_amount* – Credit amount
- *installment_commitment* – Installment rate in percentage of disposable income
- *residence_since* – Present residence since
- *age* - Age in years
- *existing_credits* – Number of existing credits at this bank
- *num_dependents* – Number of people being liable to provide maintenance for

and 13 categorical features

- *checking_status* – Status of existing checking account
- *credit_history* – Credit history
- *purpose* – Purpose of loan

[8] `https://www.openml.org/d/260`

[9] `https://www.kaggle.com/c/GiveMeSomeCredit/data?select=cs-training.csv`

- *savings_status* – Status of savings account/bonds
- *employment* – Present employment since
- *personal_status* – Personal status and sex
- *other_parties* – Other debtors / guarantors
- *property_magnitude* – Magnitude of personal property
- *other_payment_plans* – Other installment plans
- *housing* – Housing (own or rent)
- *job* – Status of present job
- *own_telephone* – Telephone
- *forgein_worker* – Foreign worker.

The data has been curated, resulting in “no data missing”. However, some categorical variables contain information that they do not represent “real data”. For example, in some cases, the variable *saving_status* contains the value *no known savings*, indicating that the actual data is missing. All such cases are summarized in Table 1.

Feature name	Number of missing values	Representations of missing values	Ratio of missing values
checking_status	394 051	’no checking’	39.40%
credit_history	40 856	’no credits/all paid’	4.08%
savings_status	184 016	’no known savings’	18.40%
other_parties	905 898	none	90.58%
property_magnitude	151 898	’no known property’	15.18%
other_payment_plans	808 505	none	80.85%
own_telephone	594 649	none	59.46%

Table 1 Information about missing values in categorical features of the German Credit dataset

To apply machine learning, all categorical variables have been manually converted, from string variables, to numerical variables. In addition, strings listed in Table 1 were transformed into label 0. Nevertheless, no data was removed form the dataset. This decision may be questioned and we plan to investigate this aspect further in the future.

4.2 Give Me Some Credit dataset

This dataset is smaller, and uses 10 features, to describe 150 000 borrowers. On the basis of the data provided, it should be anticipated whether the person described there will experience financial difficulties, in the next two years. This feature is tagged as *SeriousDlqin2yrs*. This information allows one to determine the credibility of the borrower and can be used to make a decision about approving loan application. Below, we present features available in the dataset, with their description.

- *Age* – Age in years

- *NumberOfDependents* – Number of the borrower's dependants
- *MonthlyIncome* – Amount of monthly income
- *DebtRatio* – The ratio of the sum of monthly debt. alimony and maintenance costs to monthly income expressed as a percentage
- *RevolvingUtilizationOfUnsecuredLines* – The ratio of the total balance on cards and personal lines of credit to the sum of credit limits expressed as a percentage
- *NumberOfOpenCreditLinesAndLoans* – Number of loans and open credit lines
- *NumberRealEstateLoansOrLines* – Number of housing loans and credits and household credit lines
- *NumberOfTime30-59DaysPastDueNotWorse* – Number of cases where the borrower was 30-59 days late in making payments in the last 2 years
- *NumberOfTime60-89DaysPastDueNotWorse* – Number of cases where the borrower was 60-89 days late in making payments in the last 2 years
- *NumberOfTimes90DaysLate* – Number of cases where the borrower was above 90 days late in making payments in the last 2 years

Data available in this dataset is incomplete. Specifically, there are two features that have considerable data gaps. For the *MonthlyIncome* attribute, data is missing for 29731 observations, while for the *NumberOfDependents* attribute, data is missing for 3924 observations.

However, most of the tested algorithms (mentioned in section 3.1) requires that the dataset should *not* contain any missing data. To avoid "simple imputing", which means: to fill in the missing values with 0 or -1, we used the KNNImputer algorithm from the sklearn package. This algorithm calculates the k-th nearest neighbours of a given observation, based on the Euclidean distance. Next, it selects the value of the feature to be completed from the k-neighbors. This data augmentation step was applied after the division of the dataset into the teaching and the testing parts.

5 Experimental setup

Let us now summarize the key technical aspects of the performed experiments.

- We have separately experimented (including applied preprocessing), with the two datasets described in Section 4.
- In each case, we have applied all classifiers, described in Section 3.
- Only open source classifiers were used (as specified in Section 3.1).
- In each case, available data was randomly divided into the standard 7:3 ratio, between the training and testing subsets.
- During the training, the models were fitted, and their hyper-parameters were optimized using special software, described in what follows.
- During the testing, all metrics described in Section 5.1, were calculated.
- Meta-classifiers were used to combine "suggestions" of the individual classifiers.

It should be noted that all algorithms were automatically optimized. For the scientific context of tuning of hyperparameters of classifiers, see [10]. Overall, the main goal of optimization was to minimize the function:

$$loss(f(X)) = 1 - AUC(f(X), y). \tag{3}$$

where X is the feature, y is the target, and f is the decision function. To achieve the needed optimization, the *hyperopt* [10] package has been used. Additionally, all algorithms had a parameter set to unbalanced classes (matching the characteristics of the datasets), to further improve the performance.

5.1 Performance evaluation – methodological considerations

In the works summarized in Section 3, to assess paerformance of the proposed approaches, the following standard performance metrics have been used: accuracy (ACC), Matthews correlation coefficient (MCC), the area under the ROC curve (AUC), Recall, Precision, Type I error, and Type II error. The formulae, representing these metrics have been summarized in the following equations:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$MCC = \tag{5}$$

$$= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$Type\ I\ error = \frac{FP}{TN+FP} \tag{9}$$

$$Type\ II\ error = \frac{FN}{TP+FN} \tag{10}$$

All variables, found in equations above, are calculated on the basis of the error matrix, represented in Table 2.

		Predicted	values
		Positive [11]	Negative [12]
True values	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 2 Confusion matrix used in this study.

[10] `https://github.com/hyperopt/hyperopt`

It is important to make the following meta-level observation. Regardless of the academic discussions concerning value of each performance metrics (e.g. the famous "tug-of-war" between the Precision and the Recall), our approach is rooted in the real-world practices, followed by the banking industry. As noted, one of authors has worked in a large bank and, therefore, we can reasonably claim that from the bank perspective it are the Recall and the Type I error that are the most important. In other words, bank's focus is on making sure that bad loans will not happen. Hence, these two metrics are crucial from the point of view of credit risk, as they indicate how much the financial institution could potentially lose, by giving funding to a person who will stop paying the debt. Type I error indicates the ratio of how many people the model predicted as good [13] and they turned out to be bad [14] to the total number of all "good people". Recall, on the other hand, indicates the fraction of how many people the model correct predicted as bad, to all bad creditors. In an ideal world, the financial institution would like the Recall to be close to 1. This fact provided us with the guideline, which results should be considered "the best". Therefore, in our work, *we have focused on obtaining the best Recall score*.

This fact has two important consequences. (1) Since our goal was to achieve the best Recall score, it could have happened that the fact that our results are better than that reported in the literature / obtained using other classifiers, can be seen as slightly unfair (if they were focused on improving other performance metrics). (2) At the same time, our results are *are real-world grounded* as the performance metrics of our choice are the ones that are actually pursued by the banks (the Recall, in particular). Obviously, in this way our approach differs from the academic research found in the literature.

6 Experimental Results

Let us now summarize the obtained results (keeping in mind the the methodological considerations described in Section 5). For performance comparison the results from [6, 34] were used as a "baseline". Let us now describe individual results for the two datasets.

6.1 Results for the German Credit Data

In Table 3, we summarise the results obtained for the German Credit dataset. Moreover, as a comparison, first, we show results from [6, 34], where its author used the same classifiers, but did not optimize the hyperparameters. Second, we report results from [34], where authors, firstly, used unsupervised algorithms to cluster observations, and then they build supervised individual models. We believe that both comparison fairly illustrate the comparative quality of our results.

The first conclusion is that there is no single model that has the highest score across all results. However, the model that "leads the way" is the XGBoost (highest AUC, ACC, MCC and Precision). However, as mentioned above, we focus on the Recall value. Hence, the "winner" seems to be the LinearSVM algorithm that has the highest score (0.989). Unfortunately, it also has the highest Type I error (0.957). This means that, in the banking practice, this approach should be avoided (as discussed in Section 5). Therefore, we cannot say that this is the best algorithm.

This being the case, the Adaboost and the GBDT are the "best overall algorithms", to apply to our problem, from the bank's point of view. The Recall score is high (0.65) and they have relatively low Type I error (0.09; although, again, not the lowest), so these are the algorithms that banks may want to be most interested in.

[13] good – debtors not delayed in repayment

[14] bad – debtors delayed in repayment

	AUC	ACC	MCC	Precision	Type I error	Type II error	Recall	Recall [6]	Recall [34]
XGBoost	**0.890**	**0.832**	**0.587**	**0.761**	0.143	0.239	0.645	0.393	–
Adaboost	0.779	0.831	0.584	0.754	0.091	0.351	0.649	0.44	–
GBDT	0.779	0.830	0.583	0.752	0.092	0.350	0.650	0.393	0.418
RandomForest	0.648	0.750	0.350	0.639	0.095	0.610	0.390	0.343	0.754
SVM_rbf	0.530	0.672	0.082	0.398	0.112	0.828	0.172	0.097	–
LinearSVM	0.516	0.328	0.081	0.308	0.957	**0.011**	**0.989**	0.477	0.459
KNN	0.548	0.695	0.139	0.481	**0.083**	0.821	0.179	0.373	0.443

Table 3 Results for the German Credit data.

It is also worth mentioning that, compared to [6], algorithms XGBoost, AdaBoost, GBDT, LinearSVM, SVM_rbf and RF have a higher Recall score. Finally, compared to [34], GBDT and LinearSVM also have a higher Recall score. However, the Random Forest and the KNN algorithm have worse scores. This may be due to differences in data preparation for modelling.

6.2 Results for the Give Me Some Credit dataset

In order to verify the effectiveness and stability of our strategy we have also applied the same methodology to the Give Me Some Credit dataset. Comparison between our classifiers and these from [6] is shown in Table 4. Recall, that there, authors build the same classifiers, but did not optimize hyperparameters.

	AUC	ACC	MCC	Precision	Type I error	Type II error	Recall	Recall [6]
XGBoost	0.869	0.81	**0.348**	0.228	0.02	0.772	0.769	0.17
Adaboost	**0.873**	0.801	0.345	0.222	0.019	0.778	0.784	0.178
GBDT	0.872	0.792	0.341	0.215	**0.018**	0.785	**0.799**	0.17
RandomForest	0.867	0.778	0.326	0.204	**0.018**	0.796	0.796	0.152
SVM_rbf	0.656	0.643	0.12	0.106	0.044	0.894	0.584	0.0
LinearSVM	0.513	**0.934**	0.12	**0.621**	0.065	**0.379**	0.027	0.006
KNN	0.54	0.931	0.055	0.265	0.066	0.735	0.019	0.013

Table 4 Results for the Give Me Some Credit data.

Note that, here, the XGBoost algorithm did not have the highest values of the 4 performance metrics, as in above Section. In this case, only the MCC metric (0.348) had the highest value, but this value is much lower than the satisfactory one (above 0.5). The best algorithm in terms of the number of "best" values of metrics was the SVM with the linear kernel, because it had the highest ACC (0.934), Precision (0.621) and the lowest Type II error (0.379).

However, note that the most important metric for us is Recall. Here, the algorithm that had the highest Recall score (0.799) was the GBDT. It also had the second highest AUC (0.872), and the lowest Type I error (0.018, including Random Forest), which (as noted above) is the second equally important metric in credit risk control. Note also that Adaboost had the highest AUC (0.873). The worst results were achieved by the SVM_rbf and the KNN models.

Compared to the [6], all algorithms calculated with our methodology achieve higher Recall scores, especially in boosting and bagging models. For instance, for XGBoost, AdaBoost, GBDT and Random Forest, the difference is more than 50% percentage points. This is an improvement in the quality of the classifiers of over 300%. This may be due to the fact that the data was very unbalanced and our algorithms took this fact into account.

6.3 Results for the meta-classifiers

Let us now discuss the results for the two consensus models: the **Weighted Average Recall Error (WARE)** and the **Weighted Average Type-1 Error (WA-T1E)**, described in section 3.1 Both were applied to the two datasets. They were tested because they had promising results reported in [11]. In Table 5 results for the German Credit dataset are presented. Next, in the Table 5, results for the Give Me Some Credit dataset are summarized.

	WARE	WA-T1E
AUC	0.884	0.869
ACC	0.827	0.806
MCC	0.568	0.507
Recall	0.585	0.458
Precision	0.786	0.817
Type I error	0.161	0.196
Type II error	0.214	0.183

Table 5 Results for the German Credit data for consensus models.

In Table 5 we can notice that the meta-classifiers have not reached a single higher metric, over the best metric from the individual models, shown in Table 3. However, the MCC metric is high and is comparable to the results of the XGBoost model. The Recall score was lower than expected, and the Type I error was one of the highest.

	WARE	WA-T1E
AUC	0.873	0.867
ACC	0.876	0.924
MCC	0.395	0.391
Recall	0.667	0.432
Precision	0.305	0.432
Type I error	0.026	0.041
Type II error	0.695	0.568

Table 6 Results for the Give Me Some Credit data for consensus models.

Table 6 shows more promising results. The MCC metric achieved a better result (0.395) than the best result from individual models (0.348, for the XGBoost model). The AUC achieved the same result (0.873), as the best result from the individual models (for the AdaBoost model). Moreover, ACC, Precision and Type II error, have the second best result, compared to the individual models shows in Table 4. However, for our most important metrics, i.e. Recall and Type I error, the results are one of the worst.

7 Concluding remarks

With the expanding and rapidly growing credit market, and the popularity of machine learning methods, ML-based credit scoring approaches are becoming an increasingly important aspect of differentiating between good and bad borrowers. Due to the area in which banks operate, even a small change in the predictive power of the models results in large income growth. In this work, we propose to change the popular (in scientific literature) approach to credit scoring models so that the main goal is the best possible Recall score, and a methodology to achieve this.

In this context we have discussed: optimization of hyperparameters of the models, in terms of the best Recall score, and taking into account class imbalance. We have conducted experiments on two publicly available benchmarks (Section 4). In both cases, the best Recall was achieved by the SVM model with the linear kernel, but it had the highest Type I error, which is the second most important parameter from the bank's point of view. Therefore, among the examined algorithms, the Gradient Boosting Decision Trees turned out to be "the best" in both cases. They had high Recall score and, at the same time, very low Type I error. Besides, for both datasets, the models with the worst results were the SVM with radial kernel, and the K-nearest neighbors.

Then, we used two related methods, to create meta-classifiers. However, it turned out that this did not bring the expected (positive) results. The Recall score and the Type I error turned out to be worse than in the case of individual models. However, we observed improvement of results for the Give Me Some Data (which is very unbalanced) for the MCC and the AUC metrics. This means that these two simple meta-classifiers are an interesting solution, but for our problem it is not a good solutions. In our future work we focus on more complicated meta-classificators, such as these presented in [23] and we try to establish if this methodology can result in a better Recall score. We will report on our work in subsequent publications.

References

1. Raport kredyt trendy. pages 1–45, 2019.
2. J. Abellán and F. Castellano. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, 73, 12 2016.
3. M. Alaraj and M. Abbod. A systematic credit scoring model based on heterogeneous classifier ensembles. 09 2015.
4. M. Alaraj and M. Abbod. Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems*, 104, 04 2016.
5. M. Alaraj and M. Abbod. A new hybrid ensemble credit scoring model based on classifiers consensus system approach. *Expert Systems with Applications*, 64, 07 2016.
6. Aleksandra Hernik, Jerzy Balicki. Metodyka porównania algorytmów klasyfikacji z pomoca benchmarków do oceny wiarygodności kredytowej, 2018. accessed via email J.Balicki@mini.pw.edu.pl.
7. E. I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, pages 589–609, 1968.
8. A. Ben-David and E. Frank. Accuracy of machine learning models versus "hand crafted" expert systems - a credit scoring case study. *Expert Systems with Applications*, 36:5264–5271, 04 2009.
9. A. Bequé and S. Lessmann. Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications*, 86, 05 2017.
10. J. Bergstra, D. Yamins, and C. D. D. Hyperopt: A python library for optimizing thehyperparameters of machine learning algorithms. 2013.
11. M. Bourel, C. Crisci, and A. Martínez. Consensus methods based on machine learning techniques for marine phytoplankton presence-absence prediction. *Ecological Informatics*, 42, 09 2017.

12. L. Breiman. Some infinity theory for predictor ensembles. *Technical Report 577*, 2000.
13. L. Breiman. Consistency for a simple model of random forest. *Technical Report 670*, 2004.
14. A. Byczynska, M. Ganzha, M. Paprzycki, and M. Kutka. Evidence quality estimation using selected machine learning approaches. pages 1–8, 03 2020.
15. N. Chen, B. Ribeiro, and A. Chen. Financial credit risk assessment: a recent review. *Artificial Intelligence Review*, 45, 10 2015.
16. L. Cleofas, V. García, A. Marqués, and J. Sánchez. Financial distress prediction using the hybrid associative memory with translation. *Applied Soft Computing*, 44:144 – 152, 04 2016.
17. C. Cortes and V. V. Support vector machine. *Machine Learning*, pages 1303–1308, 1995.
18. T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13:21–27, 1967.
19. R. Florez and J. Ramon. Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. a correlated-adjusted decision forest proposal. *Expert Systems with Applications*, 42, 08 2015.
20. Y. Freund and S. Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and system sciences*, pages 119–139, 1996.
21. J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 10 2001.
22. J. H. Friedman. Greedy function approximation: A gradient boosting machine. *IMS 1999 Reitz Lectures*, 1999.
23. M. Ganzha, M. Paprzycki, and S. Jakub. Combining information from multiple search engines - preliminary comparison. *Information Sciences*, 180:1908–1923, 2010.
24. A. Hanic, E. Žunić Dželihodžić, and A. Dzelihodzic. Scoring models of bank credit policy management. *Economic Analysis*, 46:12–27, 01 2013.
25. W. Henley and D. J. Hand. A k-nearest-neighbour classifier for assessing consumer credit risk. 1996.
26. S. Lessmann, B. Baesens, H.-V. Seow, and L. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, (doi:10.1016/j.ejor.2015.05.030), 05 2015.
27. M. Machowiak. Personal communication.
28. M. Paprzycki and M. Machowiak. Development of e-banking in poland. pages 1–8, 2000.
29. R. E. Schapire and K. Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*, 42:164–166, 02 2013.
30. S. H. Shashi Dahiya and N. Singh. A feature selection enabled hybrid-bagging algorithm for credit risk evaluation. *Expert Systems*, page 34(6), 2017.
31. S. Sohn, D.-H. Kim, and J. Yoon. Technology credit scoring model with fuzzy logistic regression. *Applied Soft Computing*, 43:150–158, 06 2016.
32. L. Thomas, R. Oliver, and D. Hand. A survey of the issues in consumer credit modelling research. *Journal of the Operational Research Society*, 56:1006–1015, 09 2005.
33. B. Twala. Multiple classifier application to credit risk assessment. *Expert Systems with Applications*, 37:3326–3336, 04 2010.
34. B. Wang, Y. Kong, Y. Zhang, D. Liu, and L. Ning. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Systems with Applications*, 128, 03 2019.
35. G. Wang, J. Ma, L. Huang, and K. Xu. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, 26:61–68, 02 2012.
36. J. C. Wiginton. A note on the comparison of logit and discriminant models of consumer credit behavior. *The Journal of Financial and Quantitative Analysis*, pages 757–770, 1980.
37. P. Wysiński. The use of credit scoring in credit risk management. *International Business and Global Economy*, pages 253–268, 2013.
38. Y. Xia, C. Liu, B. Da, and F. Xie. A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, 93, 2017.