Computer Science and Information Systems 16(1):155–178 https://doi.org/10.2298/CSIS180110037A

An insight into the effects of class imbalance and sampling on classification accuracy in credit risk assessment

Kristina Andrić[1], Damir Kalpić[2], and
Zoran Bohaček[3]

[1] Privredna Banka Zagreb, Radnička 50,
10000 Zagreb, Croatia
kristina.andric@fer.hr
[2] University of Zagreb Faculty of electrical engineering and computing, Unska 3,
10000 Zagreb, Croatia
damir.kalpic@fer.hr
[3] Croatian Banking Association, Nova Ves 17,
10000 Zagreb, Croatia
zoran.bohacek@alumni.unizg.hr

Abstract. In this paper we investigate the role of sample size and class distribution in credit risk assessments, focusing on real life imbalanced data sets. Choosing the optimal sample is of utmost importance for the quality of predictive models and has become an increasingly important topic with the recent advances in automating lending decision processes and the ever growing richness in data collected by financial institutions. To address the observed research gap, a large-scale experimental evaluation of real-life data sets of different characteristics was performed, using several classification algorithms and performance measures. Results indicate that various factors play a role in determining the optimal class distribution, namely the performance measure, classification algorithm and data set characteristics. The study also provides valuable insight on how to design the training sample to maximize prediction performance and the suitability of using different classification algorithms by assessing their sensitivity to class imbalance and sample size.

Keywords: credit risk assessment, imbalanced data sets, class distribution, classification algorithms, sample size, undersampling.

1. Introduction

The aim of credit scoring is to differentiate borrowers and classify them into two groups: good clients (non defaulters) and bad clients (defaulters), using predictive models and historical information on clients' repayment behavior. It is widely used in financial institutions for lending decisions and asset quality monitoring and is one of the most widely used applications of statistical models and data mining methods in practice [1]. Therefore, making a correct assessment of the likelihood of the client defaulting is of utmost importance for financial institutions [2] and increasing accuracy in default prediction would be of great interest for financial institutions [3].

Class imbalance is a common occurrence in credit scoring, where the good clients greatly outnumber the bad clients. As one of the key challenges in data mining [4], it has received a lot of attention in recent years. Performance of standard classification algorithms tends to deteriorate when class imbalance is present as the cost of misclassifying a case in credit scoring is greatly asymmetrical. Frequent occurrence of class imbalance in credit risk assessment indicates the need for additional research efforts in handling imbalanced datasets. Several methods have been proposed to improve prediction accuracy in these settings, such as altering class distribution [5].

Despite having access to vast amounts of data related to customer behavior, standard practice is to develop predictive models using a sample of data, i.e. representative of the overall population. Even though a lot of research has been focused around prediction accuracy of different classification algorithms, topics of data sample design have been largely neglected regardless of the fact that data sample preparation is the cornerstone and the most time consuming step of the model development process [6]. Little research effort has been devoted to assessing the impact of class distribution on credit scoring, in particular when dealing with real life data samples, that can be very heterogeneous and differ in terms of size, imbalance ratio (IR, to be defined later), the number of features etc. Majority of studies use benchmark data whose characteristics are not representative of real-life data samples, which may introduce a bias, or use a very limited number of data samples that contain only a few independent variables [7], [8]. Sample size is especially important for low default portfolios, where, due to the lack of a systematic empirical evaluation, it is unclear what is the minimum number of instances that would justify model development. Nevertheless, this topic has not been evaluated systematically with empirical experiments, despite its apparent importance and applicability.

To address the observed research gap, a study of the interdependence of class distribution, dataset characteristics and prediction accuracy in a real life credit scoring environment was performed using logistic regression, neural network and gradient boosting algorithm on a wide array of real life data samples. Class imbalance was increased progressively to determine how classification accuracy was affected by different IRs. Performance of the proposed framework was evaluated with several evaluation measures. The second part of the study addresses the question of how sample size impacts classification accuracy and finding the minimal sample size (namely the number of defaulters) needed to develop a model with adequate prediction accuracy. The main contribution from this study can be summarized as: (1) finding the optimal class distribution that maximizes classification accuracy, (2) assessment of different classification algorithm performance when dealing with different imbalance ratios and samples of different size, (3) assessment of gradient boosting algorithm in credit risk domain, (4) finding the minimal data sample size that can be used for model development with adequate prediction accuracy and (5) analysis of the intrinsic characteristics of the data samples (IR, number of minority class cases, number of total instances).

The rest of this paper continues as follows: Section 2 provides a literature review. The classification algorithms are described in Section 3, and the experimental framework in Section 4. Results of the study are presented and discussed in Sections 5. and 6. Conclusions and possible future research direction are presented in Section 7.

2. Background work

A lot of research effort has been committed to evaluating classification algorithms in credit scoring, ranging from traditional statistical methods, such as logistic regression [1], to non-parametric algorithms, such as neural networks [9]. In the recent years there has been an increased interest for using hybrid and ensemble classifiers in credit risk, such as boosted regression trees, random forests, deep learning methods and other [10], [11], [12], [13] [14]. A number of benchmark studies have been performed, comparing classification accuracy of different classification algorithms [15], [8]. However, there is no consensus among researchers as to which algorithms yield the best performance. In fact, it seems that the choice of algorithm should take into account the domain, dataset and evaluation criterion [1], [16].

Class imbalance is inherent to credit scoring. In [17] the authors analyzed how class imbalance impacts several classifiers by steadily increasing the IR. They concluded that gradient boosting and random forest achieved good performance, in particular when the IR was very high. Logistic regression also achieved high accuracy, unlike other methods, such as SVM and C4.5. One of the methods proposed to alleviate the adverse impact of using imbalanced data sets includes using pre-processing techniques, which alter the original imbalanced data sample to produce a more balanced class distribution [18]. With oversampling, the number of defaulted clients is increased, whereas with undersampling the number of good clients is decreased. A few comparisons of different sampling techniques were performed, but with different conclusions reached on which technique yields highest accuracy. In fact, research suggests that intrinsic data set characteristics and the application domain have a greater impact on the performance than a particular pre-processing technique, indicating that further analysis is needed [19], [20], [21].

Adjusting class distribution can lead to disregarding potentially valuable information from the sample or overfitting. Therefore, finding the optimal class distribution and its empirical evaluation is of great significance. Although not related to credit scoring, in [22] the authors studied the C4.5 classifier, concluding that balanced class distribution in general achieves the best results. However, in credit scoring the number of defaulted clients is usually very small and a balanced class distribution is hardly ever encountered in real life. In [23] the authors analyzed the effect of different class distributions in credit risk assessment. Class distribution was altered by using random undersampling to produce various imbalance ratios. The results stressed the importance of assessment criterion, reaching a conclusion that performance deteriorated with all classification algorithms used when faced with larger class imbalance.

A common approach in credit scoring is to develop predictive models using a smaller sample of the overall population, that should resemble the target population as much as possible, due to various reasons. As first, these models are subjected to restrictive out of sample validation procedures, implying that all the data available cannot be used for model development. As second, using larger samples increases the cost, in terms of time and computational resources needed to develop a model. As third, when population drift occurs, where characteristics of the population change over time, the entire historical sample cannot be used for model development, since it is not representative of the target population. Population drift has become an especially important topic in recent years, since during the financial crisis of 2007 the overall macroeconomic environment changed and both customers and financial institutions changed their behavior. Therefore, sample size and construction are of utmost importance for the quality of predictive models.

Despite its relevance, studies on choosing the optimal sample size are limited. There is a common understanding among practitioners that a sample containing around 1500 cases of each class should suffice to build predictive models [6]. However, in [20] the authors performed an empirical study on sampling in credit scoring, using two datasets and four classification algorithms. The study indicated that using samples larger than those recommended in practice increases prediction accuracy.

The focus of research so far has been proposing new algorithms and comparing classification accuracy of different classification algorithms, whereas the issues of sample design were largely ignored. This study aims to address the observed gaps and provide insight into credit risk modelling with empirical evidence on how to design the training sample and which algorithm might be appropriate in given settings.

3. Classification algorithms

Classification algorithms used in the study include logistic regression, one of the most commonly used methods for credit scoring [20], neural networks, a classifier whose good performance was demonstrated in many studies [9], [15] and gradient boosting, a relatively new method, yet to be thoroughly evaluated in the credit risk domain.

3.1. Logistic regression

Logistic regression (LR) is the standard method used for credit scoring among practitioners and one of the most commonly used methods for credit scoring [20]. LR estimates $p(y = 1|\mathbf{x})$. y represents the dependent variable, equal to 1 if the client is in a default status. $\mathbf{x}$ is a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, or a collection of n independent variables. $p(y = 1|\mathbf{x})$ then represents the conditional probability that the outcome of the client defaulting is present. The relationship between the independent variables and the dependent variable is described by a logit function [24]:

$$\text{logit}\,(p(y = 1|\mathbf{x})) = \ln\left(\frac{p(y = 1|\mathbf{x})}{1-p(y = 1|\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \,. \quad \textbf{(1)}$$

β_0 is the intercept parameter and $\beta_1, \ldots, \beta_n$ are the model parameters. The model parameters are estimated through minimizing the log-likelihood function:

$$min\, l = -\sum_{i=1}^{n} y_i \, ln\big(p(y = 1|\mathbf{x})\big) + (1 - y_i)\, ln\big(1 - p(y = 1|\mathbf{x})\big)\,. \quad \textbf{(2)}$$

3.2. Neural networks

Neural networks (NN) are nonlinear classifiers modelled after the human brain [25]. The multilayer perceptron (MLP) is one of the most commonly used types of neural networks. It contains an input layer, a hidden layer, and an output layer. In each of the layers neurons process their inputs into output values, used by the neurons in the next layer. The independent variables are the input layer neurons. Initial weights are assigned to connections between neurons and adjusted during training. This iterative process can be

based on various methods, such as gradient descent, Quasi-Newton etc. Output of the hidden neuron h_i is determined through an activation function f:

$$h_i = f\ (b_i + \sum_{j=1}^{n} w_{ij} x_j)\ . \quad (3)$$

x represents a collection of n independent variables, **x** = ($x_1, x_2, \ldots, x_n$), w_{ij} is the weight connecting input j to neuron i and b_i is the bias term. Output of the output layer is determined using a similar function**:**

$$o = g\ (b_i + \sum_{j=1}^{m} v_i h_i)\ . \quad (4)$$

m is the number of neurons in the hidden layer and v_i is the weight connecting hidden neuron h_i to the output neuron. MLP use sigmoidal f and g functions. The model parameters w_{ij}, v_i, b_i are determined through minimizing the loss function using gradient-based algorithms.

3.3. Gradient boosting

Gradient boosting (GB), also referred to as stochastic gradient boosting, is an ensemble model that consists of a series of simple decision trees. Ensemble models aim to improve accuracy by combining predictions of multiple base models and minimizing the error term in an iterative manner. After the initial base model (tree) is set up, each subsequent base model is fitted to the residuals of the previous model to minimize the error term and avoid errors of the current ensemble [26]. Since it is a homogeneous ensemble classifier, additional randomness is introduced by bootstrap sampling [8]. The model is described as:

$$F(x) = G_0 + \sum_i \beta_i L_i\ (x)\ . \quad (5)$$

β_i are coefficients for the respective tree node L_i fitted to the residuals of the previous algorithm and G_0 the first value for the series.

4. Experimental design

4.1. Data samples

To conduct the experiments, ten different data sets were used. Their characteristics are outlined in Table 1. Japanese and German data set are public data sets, available in the UCI Machine Learning Repository [27] and include data related to credit card applications. The German dataset has 1000 cases, 300 of which are defaulted, with 20 features. The Japanese data set has 690 cases, 329 of which are defaulted, with 13 features. Data sets DS1 - DS8 are real-life samples obtained from a commercial bank. Even though

a limited number of real-life data sets were used, datasets used for credit scoring usually show a high level of comparability across different credit institutions [20]. This notion stems from the fact that types of consumer data gathered by institutions broadly represent same types of information for similar types of portfolios. For a number of countries (for example EU countries) using the types of data that were used in this study is also a legal requirement [28], [29]. Consequently, institutions are required to take into consideration a broad set of information, including borrower risk characteristics, (borrower type, demographics, financial information, etc.), transaction risk characteristics, (product types, collateral information, etc.) and behavior patterns (historical patterns on repayment behavior etc.).

Table 1. Data sets

Size	Data set	No of cases	No of features	No of good clients	No of bad clients	Imbalance ratio (IR)	
Small	German	1,000	20	700	300	2	Low
	Japanese	690	13	361	329	1	
	DS1	3,053	206	2,757	296	9	
Mid	DS2	25,226	218	25,016	210	119	Severe
	DS3	67,800	162	64,831	2,969	22	Moderate
	DS4	65,535	160	62,542	2,993	21	Moderate
	DS5	26,590	198	26,397	193	137	Severe
Large	DS6	205,720	160	198,772	6,948	29	Moderate
	DS7	197,462	160	189,479	7,983	24	Moderate
	DS8	232,275	160	223,569	8,706	26	Moderate

DS3, DS4 and DS6 - DS8 data sets include information on retail clients. Features include sociodemographic characteristics of the client (20 features) such as age, occupation etc.; income and other available funds information (29 features); loan characteristics (14 features) such as products type, collateral details or seasoning information; behavior patterns such as balances (39 features), repayment behavior (47 features) and external information (11 features). DS1-DS2 and DS5 data sets include information on corporate clients. Features reflect details on borrower characteristics such as type, industry, region and size (27 features), financial information such as balance sheet (40 features), or profit and loss details (33 features), cash flows analysis (41 features), debt details (8 features), various ratios (29 features) and trend indicators (14 features), as well as external information (11 features). Every data set includes a variable that indicates whether a default event was observed during the performance period of one year. Clients with outstanding debt for more than 3 months were marked as defaulted.

German and Japanese data sets are frequently used as benchmark datasets, especially for evaluating performance of different methods. Unlike the benchmark datasets, real-life data sets are characterized by a much greater number of features (over 150 features), as well as much greater size and class imbalance. The key characteristics analyzed in this study are sample size, class imbalance and the number of minority class cases. The extent of class imbalance is defined with imbalance ratio (IR), the ratio of the number of majority class cases and minority class cases. Data sets were divided by size, the number of bad clients and imbalance ratio. Those containing less than 10,000 observations were marked

as small sized, while the ones with more than 10,000 but less than 100,000 observations were marked as midsized data sets. Data sets with more than 100,000 observations were marked as large. Except for the Japanese data set, all the data sets are imbalanced, displaying different imbalance ratios. Data sets with IR smaller than 10 were categorized as having a low IR, those with IR between 10 and 50 as moderately imbalanced, while the data sets with IR higher than 50 were categorized as severely imbalanced.

4.2. Data transformation

A graphical illustration of the research process is presented in Figures 1. and 2. Fig 1. illustrates the first part of the study focusing on the impact of class imbalance on accuracy of different algorithms, whereas Fig. 2 illustrates the second one focusing on the impact of sample size. The analysis was performed in the context of a specific algorithm since different classification algorithms are expected to exhibit varying sensitivity on both class distribution and size.

Repeated stratified random sampling, or Monte Carlo cross-validation [30] was used, where the data sets were split randomly into a training and a validation data set four times, generating training samples T_1-T_4 and validation samples V_1-V_4. In each iteration, a case was assigned to either the training data set or the validation data set. Stratified sampling was used, with the cases being assigned either to the training or the validation data set randomly, but keeping the original imbalance ratio constant. The classifiers were built on the training data set, whereas the validation data set was used for performance evaluation. For both experiments the data sets described in Table 1 were split randomly into a training (T_i) and a validation (V_i) data set using the 70%/30% ratio, as shown under Step 2 in Fig. 1 and Fig. 2. The same approach was used throughout the study for all the algorithms.

In order to assess the impact of gradually changing the level of imbalance, the initial training data sets were gradually undersampled to reach target imbalance ratios. Target IR ranged from 90/10, with 10% of the defaulted cases, to 20/80, with 80% of the defaulted cases. Class distribution was altered by random undersampling, where either non-defaulted clients or both defaulted and non-defaulted clients were removed to reach the target imbalance ratio. In other words, the defaulted clients' ratio, for all the data sets was altered by a factor of 10% to create greater disparities in the level of imbalance. Eight new data sets T_{i1}- T_{i8} with different IR were therefore created for each original data set, resulting in overall 320 different train data samples. This stage is illustrated in Fig. 1 under Step 2. It is important to emphasize that only the training data sets were altered, leaving the validation data sets and their underlying class distribution unchanged. Classifiers C_1-C_8 were trained using train samples T_{i1}- T_{i8} (Fig. 1, Step 4.) and applied to the validation set V_i (Fig. 1, Step 5.). The performance metrics, outlined in Chapter 4.6., were calculated for each V_i (Fig. 1, Step 6.). However, with Monte Carlo cross-validation, the final performance metrics, reported in the remainder of the paper, were computed as an average over all validation sets V_i to ensure reliable estimates of the results (Fig. 1, Step 7.).

For the study on the impact of sample size a similar approach with repeated stratified random sampling was used. Data set DS8 was used, as a data set representative of real-life behavioral data samples. Under a specific IR, the size of the sample was progressively decreased, while keeping the IR constant (Fig. 2, Step 3.). For each IR, the sample size was decreased by a factor of 10% until reaching 20% of the original size, followed by a reduction of 5% until reaching 5% of the original size with an additional reduction to 3%

of the original size. As a result, for each subsample T_1- T_8 new train subsamples T_{1j}- T_{8j} of different sizes were created. Classifiers C_{1j}- C_{8j} trained using train samples T_{1j}- T_{8j} (Fig. 2, Step 4.) were applied to the validation set V_i (Fig. 2, Step 5.). The final performance metrics, reported in the remainder of the paper, were computed as an average over all validation sets V_i (Fig. 1, Step 7.).

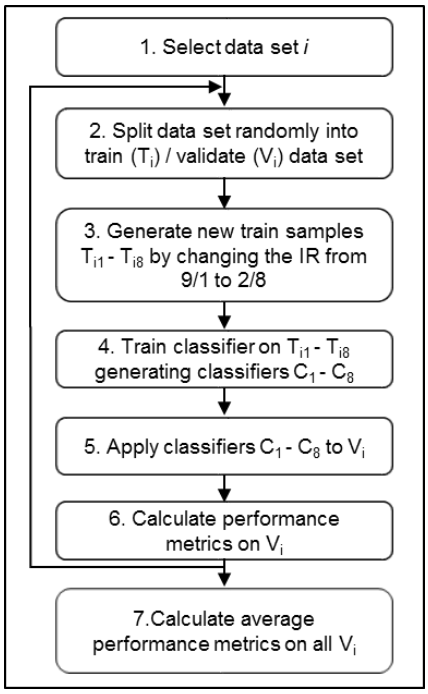

Fig. 1. Class imbalance impact evaluation

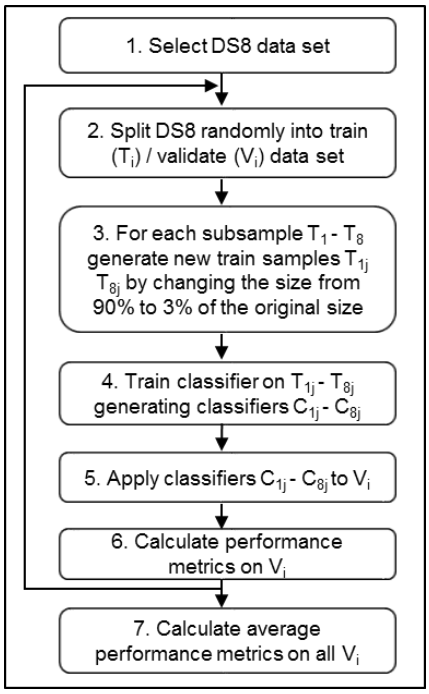

Fig. 2. Sample size impact evaluation

4.3. Classification algorithms and feature selection

For logistic regression we used the stepwise method to select independent variables. To determine the weights for neural networks we used the standard backpropagation algorithm. The hidden layer was set to include 3 units with no direct links between input and output nodes. For gradient boosting we used binary trees and the number of terms in the boosting series equal to 50. Features selection was performed by using Chi-square and R-square analysis. To prevent overfitting, the number of features was limited to 7.

4.4. Performance evaluation metrics

Choosing the right performance metrics when dealing with imbalanced data sets is crucial for appropriate evaluation of the classifier's accuracy [7]. Traditionally, the confusion

matrix was used to compare the predictive outcome of the algorithm with the true values [31].

Table 2. Confusion matrix

		Prediction		Accuracy rate (acc) = (TP+TN)/(TP+TN+FP+FN) True positive rate(TPR) = (TP)/(TP+FN) True negative rate(TNR)= (TN)/(FP+TN) False negative rate(FNR) =(FN)/(TP+FN) False positive rate(FPR) = (FP)/(FP+TN)
		Positive	Negative	
Actual	Positive	TP (true positive)	FN (false negative)	
	Negative	FP (false positive)	TN (true negative)	

From the confusion matrix, several metrics can be induced such as the *Accuracy rate (acc)*, that measures the ratio of correct predictions of the algorithm. When dealing with imbalanced data sets, performance measures insensitive to class distribution should be used. Classical measures, such as *acc*, exhibit an inherent bias toward the majority class, since for example the error of ignoring the minority class completely would be only 2% if the data set contained only 2% of minority class cases. Other, more appropriate metrics of performance can be used, monitoring accuracy for both classes independently. *True positive rate (TPR)* represents the ratio of all the defaulted clients that were correctly classified by the algorithm among all defaulted clients. It is also referred to as sensitivity or recall. *True negative rate (TNR)* is the ratio of all the non defaulted clients that were correctly classified by the algorithm among all non defaulted clients. It is also referred to as specificity. The goal of a classifier is to maximize the true positive and true negative rates. In credit risk assessment, default prediction is of greater interest since misclassifying a bad client attracts a much higher cost than misclassifying a good client.

In this study measures that consider prediction accuracy for both classes were used, the Receiver Operating Characteristic *(ROC)* and corresponding area under the ROC Curve *(AUC)*, as well as geometric mean of the true rates *(GM)*. The two measures represent different types of performance indicators and assess the predictive performance of the classifier from different angles. *ROC* is one of the most frequently used measures. *AUC* represents the probability that a randomly chosen positive case (a defaulted client) will be ranked higher than a randomly chosen negative case (a non defaulted client) [7]. It visualizes the trade-off between sensitivity and 1-specificity [32]. An algorithm that classifies all cases correctly would include point (0, 1) and a random algorithm point (0.5, 0.5). The geometric mean of the true rates measure *(GM)* on the other hand combines measures of correctness of the binary classification predictions, allowing for simultaneous maximization of the prediction accuracy for both classes. It is defined as follows [31]:

$$\text{GM} = \sqrt{TP_{rate} * TN_{rate}} \quad \textbf{(6)}$$

5. Empirical results - the impact of class distribution

5.1. Using geometric mean *(GM)* as performance metric

Tables 3, 4 and 5 summarize the results for all data sets at different IR, with logistic regression shown in Table 3, neural network in Table 4 and gradient boosting in Table 5. All are calculated using the validation data sets. The results show the rank of a given classifier for each imbalance ratio. The imbalance ratio where the highest result is achieved is marked by "1". Average result across all the data sets for a given class distribution is also shown, together with the coefficient of variation (*CV*), or the ratio of the standard deviation of *GM* to the mean *GM*. Fig. 3 presents how the *GM* is affected by the changes in the underlying class distribution for small, medium and large data sets.

Logistic regression - impact of sample size and imbalance ratio. The best results in terms of *GM* are achieved when the data sample is more or less balanced, or when the imbalance ratio ranges from 6/4 to 4/6. Similar results are observed for all data samples, regardless of their size. When analyzing the average *GM* across all data sets for a given imbalance ratio, the best results are achieved when the imbalance ratio is equal to 1. This can be observed also for small sized data sets, whereas for midsized data sets slightly better results on average are obtained when the imbalance ratio is greater than 1, 6/4 and for larger data sets when the imbalance ratio is smaller than 1. Choosing the optimal class distribution increases the *GM* by 55% on average. Overall, logistic regression did not show high sensitivity to IR.

If the absolute number of bad clients is taken into account, it can be observed that data samples with a smaller number of bad clients achieve the best results when the number of good clients is larger than the number of bad clients (6/4). Data samples with an adequate number of bad clients benefit most from a balanced class distribution and the ones with a higher number of bad clients when the imbalance ratio is smaller than 1, i.e. 4/6.

Neural networks - impact of sample size and imbalance ratio. Neural networks display similar performance, where the best result in terms of *GM* is achieved when the data sample is more or less balanced, more specifically when the imbalance ratio ranges from 6/4 to 4/6. However, neural networks show less sensitivity to changes in class imbalance than logistic regression, as can be seen by observing smaller disparities in the coefficient of variation (*CV*). When analyzing the average *GM* across all data sets we observed that a balanced distribution yields the highest accuracy. The same result can be observed for both small and large data sets, whereas for midsized data sets slightly better results on average are obtained when the imbalance ratio is greater than 1, i.e. 6/4. Choosing the optimal class distribution increases the *GM* by 54% on average.

If the absolute number of bad clients is taken into account, data samples with a smaller number of bad clients achieve the best results when the number of good clients is larger than the number of bad clients (6/4). Others benefit most from a balanced class distribution.

Table 3. *GM* rank for logistic regression

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	8	7	5	3	1	2	4	6	41%
Japanese	8	7	6	4	1	2	3	5	6%
DS1	8	7	4	2	1	3	5	6	41%
DS2	7	5	4	1	2	3	6	8	30%
DS3	8	7	5	3	1	2	4	6	27%
DS4	8	7	6	4	1	2	3	5	23%
DS5	6	3	2	1	4	5	7	8	40%
DS6	8	7	6	3	2	1	4	5	40%
DS7	7	6	5	4	2	1	3	8	50%
DS8	8	7	5	3	1	2	4	6	23%
AVG	7,6	6,3	4,8	2,8	1,6	2,3	4,3	6,3	16%

Table 4. *GM* rank for neural network

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	8	7	5	3	1	2	4	6	26%
Japanese	8	7	6	5	2	1	3	4	6%
DS1	8	7	4	2	1	3	5	6	32%
DS2	8	6	4	1	2	3	5	7	39%
DS3	8	6	3	1	2	4	5	7	23%
DS4	8	7	4	2	1	3	5	6	17%
DS5	7	4	2	1	3	5	6	8	40%
DS6	8	6	4	2	1	3	5	7	26%
DS7	8	7	5	3	1	2	4	6	40%
DS8	8	7	4	1	2	3	5	6	24%
AVG	7,9	6,4	4,1	2,1	1,6	2,9	4,7	6,3	12%

Table 5. *GM* rank for gradient boosting

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	8	7	5	4	2	1	3	6	40%
Japanese	8	7	6	1	2	3	3	3	8%
DS1	6	5	7	7	1	2	3	4	86%
DS2	5	5	5	5	1	2	3	4	120%
DS3	5	5	5	5	2	1	3	4	119%
DS4	5	5	5	5	3	1	2	4	119%
DS5	5	5	5	5	2	1	3	4	119%
DS6	5	5	5	5	2	1	3	4	119%
DS7	5	5	5	5	2	1	3	4	119%
DS8	5	5	5	5	2	1	3	4	119%
AVG	5,7	5,4	5,3	4,7	1,9	1,4	2,9	4,1	75%

Table 6. Highest average *GM*

	Imbalance ratio							
SAMPLE	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8
ALL	NN	NN	NN	NN	GB	GB	GB	GB
SMALL	NN	GB	GB	LR	LR	GB	GB	NN
MID	NN	NN	NN	NN	GB	GB	GB	GB
LARGE	NN	NN	NN	NN	GB	GB	GB	GB
REAL LIFE	NN	NN	NN	NN	GB	GB	GB	GB

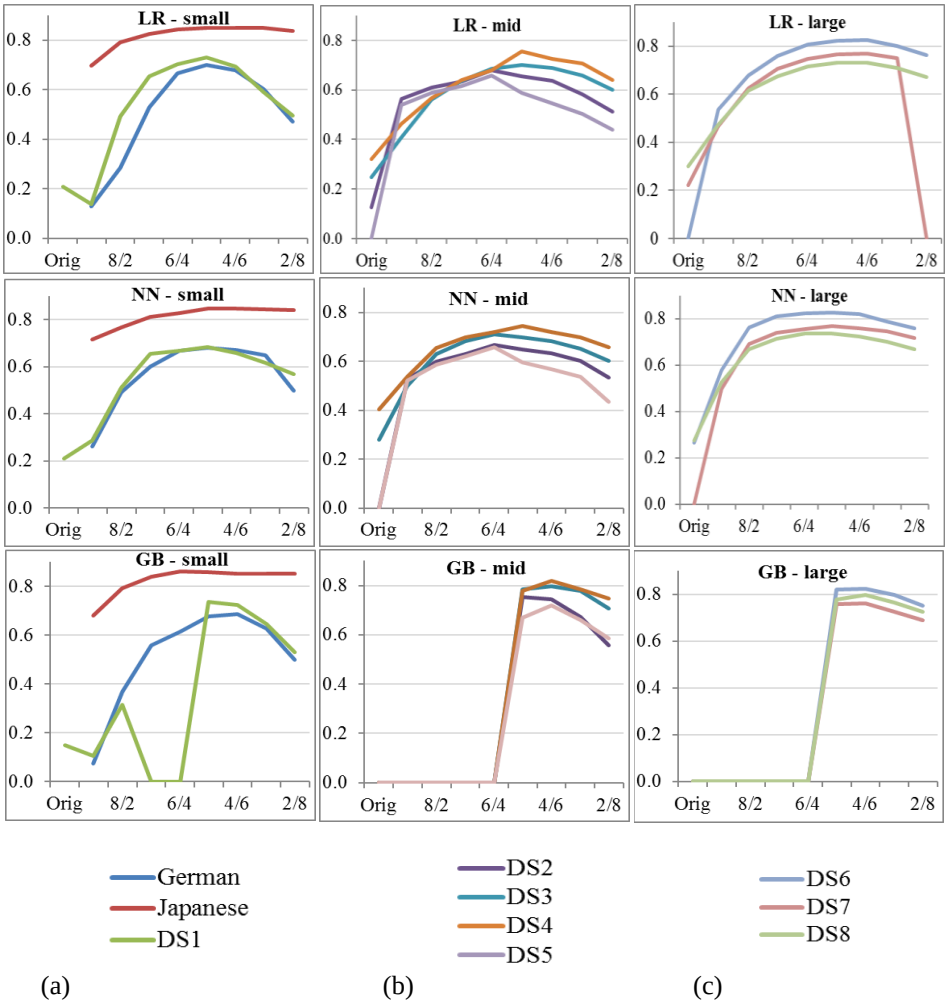

Fig. 3. *GM* for logistic regression (LR), neural network (NN) and gradient boosting (GB) for (a) small data sets, (b) mid data sets and (c) large data sets

Gradient boosting - impact of sample size and imbalance ratio. Table 5 clearly demonstrates that with gradient boosting the best results are also achieved when the data

sample is more or less balanced, with better results achieved when the imbalance ratio is smaller than one, 4/6. The same result can be observed for both mid and large data sets, whereas for small sized data sets slightly better results on average are obtained when the imbalance ratio is equal to 1. Gradient boosting displays much greater sensitivity to class imbalance than the previous two methods, as can be seen from the *CV*, but also from the fact that in vast majority of the cases when class imbalance is greater than 1, not a single bad client was identified by gradient boosting. As a result, *GM* is equal to 0%. This is especially relevant for real-life data sets (DS1-DS8). Choosing the optimal class distribution increases the *GM* by 75% on average. Similar results can be observed if the absolute number of bad clients is analyzed.

Comparison of class distribution across all classifiers. Table 6 shows the classification algorithm achieving the best results on average for a given imbalance ratio. The results are shown separately for all the data sets, depending on the size of the data set, and for real life data sets. This finding suggests that when imbalance ratio is greater than 1, neural networks in general achieve the best results, whereas for imbalance ratios smaller than 1 gradient boosting achieves the best results.

Kendall's coefficient of concordance (*W*) was used to compare the *GM*s with different class distributions. It is a non-parametric statistics based on the average ranked performance of the classification, and is calculated as follows:

$$\mathrm{W} = \frac{12S}{m^2(k^3-k)} \quad . \tag{7}$$

where *S* is the sum of squared deviations, equal to:

$$S = \sum_{i=1}^{k}(R_i - R)^2 \, . \tag{8}$$

R_i is the average rank across all classifiers, *m* the number of classifiers used to perform ranking (3) and *k* is the number of assessments subject to ranking (8). *W*= 0.81 has shown a level of agreement for all the algorithms, with *p-value* = 0.0174 < .05 = α, allowing rejection of the null hypothesis that there is no agreement among the different classification algorithms.

5.2. Using Area under the ROC curve (*AUC*) as performance metric

A similar analysis is displayed in Tables 7, 8 and 9, with *AUC* as the evaluation measure.

Logistic regression - impact of sample size and imbalance ratio. Logistic regression has shown to be insensitive to class distribution and a fairly robust technique. There is no single class distribution that yields the best performance for all the samples. However, larger disparities can be observed when the sample size and the actual number of defaulted cases is taken into account, especially with real-life data sets. For larger data samples the highest *AUC* is achieved when IR is smaller than 1 (more specifically 3/7).

For midsized data samples, the best results are achieved when the data sample is balanced, 5/5, whereas small data sets produce the best results when the IR is greater than 1 (6/4).

Table 7. *AUC* rank for logistic regression

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	8	6	5	1	4	3	1	7	1%
Japanese	5	1	4	3	6	2	7	8	0%
DS1	1	4	3	2	5	6	8	7	2%
DS2	1	2	5	4	7	3	6	8	3%
DS3	8	7	6	2	4	2	1	5	1%
DS4	8	7	6	4	1	2	4	3	0%
DS5	4	5	7	2	1	3	6	8	2%
DS6	8	7	5	5	4	3	1	1	0%
DS7	8	7	5	5	2	2	1	2	1%
DS8	7	6	5	8	3	1	3	2	1%
AVG	5,8	5,2	5,1	3,6	3,7	2,7	3,8	5,1	0%

Table 8. *AUC* rank for neural networks

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	7	1	2	3	4	5	6	8	3%
Japanese	8	3	1	6	2	4	5	7	1%
DS1	2	3	1	4	5	7	5	8	2%
DS2	2	1	3	4	7	4	6	8	3%
DS3	2	1	4	5	3	7	6	8	0%
DS4	2	1	5	6	4	3	8	7	0%
DS5	1	3	5	2	4	7	6	8	2%
DS6	2	2	1	2	5	5	7	8	0%
DS7	2	4	2	6	6	4	1	8	0%
DS8	1	1	6	3	4	7	4	8	0%
AVG	2,9	2,0	3,0	4,1	4,4	5,3	5,4	7,8	1%

Table 9. *AUC* rank for gradient boosting

Data set	Imbalance ratio								
	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8	CV
German	8	2	3	4	6	1	5	7	3%
Japanese	2	1	5	4	3	7	8	6	1%
DS1	5	2	7	8	3	1	3	6	2%
DS2	7	6	7	5	1	1	3	4	6%
DS3	6	7	8	5	4	3	1	2	4%
DS4	8	6	7	5	4	2	1	3	4%
DS5	5	7	8	6	4	1	2	3	6%
DS6	8	6	7	5	4	1	2	3	15%
DS7	8	6	6	5	3	2	1	4	14%
DS8	8	7	6	5	4	2	1	3	15%
AVG	6,5	5,0	6,4	5,2	3,6	2,1	2,7	4,1	7%

If the actual number of defaulted cases is taken into account, it can be observed that data samples with a smaller number of bad clients achieve the best results when the number of good clients is much larger than the number of bad clients (8/2). Data samples with an adequate number of bad clients benefit most from a balanced class distribution and the ones with a higher number of bad clients benefit when the imbalance ratio is smaller than 1, 3/7.

Neural networks - impact of sample size and imbalance ratio. Neural network exhibits different behavior. With neural networks the highest *AUC* is achieved when the IR is significantly greater than 1, or more specifically when the ratio of good/bad clients ranges from 9/1 to 7/3. The only exception is the DS7 data set, where the distribution of good/bad clients of 3/7 yields the highest *AUC*. Similar to logistic regression, it seems that the sample size plays an important role when it comes to determining the optimal class distribution. For larger data samples the best result is obtained when the imbalance ratio is less than one, 3/7, whereas midsized data samples tend to produce the best results when the class distribution is equal to 8/2. Small data sets produce the best results when the imbalance ratio is 7/3. Nevertheless, neural networks did not show significant variation in performance depending on the imbalance ratio, as demonstrated by the coefficient of variation (*CV*), which is relatively small for all data sets.

If the actual number of minority class examples is considered, data samples with a smaller number of bad clients achieve the best results when the number of good clients is much larger than the number of bad clients (8/2).

Gradient boosting - impact of sample size and imbalance ratio. For gradient boosting class distribution ranging between 5/5 to 3/7 generally yields the best performance in terms of *AUC*. Gradient boosting is more sensitive to class imbalance than neural networks and logistic regression. When it comes to choosing the optimal class distribution, greater disparities are observed, depending on the sample size, especially with real-life data sets. Gradient boosting produces the best results on average when the imbalance ratio is greater than 1, or more specifically when the imbalance ratio equals 3/7 for larger data samples and 4/6 for midsized data samples. Class distribution of 8/2 yields best performance in terms of *AUC* for smaller data sets.

Data samples with a small number of bad clients achieve the best results when the number of good clients is smaller than the number of bad clients (4/6). Other data samples show the best results when the imbalance ratio is even lower (3/7).

Comparison of class distribution across all classifiers. We observed that regardless of the classification algorithm, changes in the underlying class distribution do not have a great impact on *AUC*. In fact, it seems that imbalance per se does not necessarily cause bad performance. As expected, when presented with large class imbalances, algorithms do not exhibit high classification accuracy, especially when a small number of minority class observations exist. In general, using a more balanced class distribution improves performance, which is especially evident with gradient boosting. Table 10 shows the classification algorithm achieving the best results on average for a given imbalance ratio. The results are shown separately for all the data sets, depending on the size of the data set, and for real-life data sets. This finding suggests that when imbalance ratio is greater than 1, neural networks in general achieve the best results, whereas for imbalance ratios smaller than 1 gradient boosting achieves the best results.

Table 10. Highest average *AUC*

	Imbalance ratio							
SAMPLE	9/1	8/2	7/3	6/4	5/5	4/6	3/7	2/8
ALL	NN	NN	NN	LR	GB	GB	GB	GB
SMALL	LR	GB	GB	LR	GB	GB	GB	GB
MID	NN	NN	NN	GB	GB	GB	GB	GB
LARGE	NN	NN	NN	NN	NN	GB	GB	GB
REAL LIFE	NN	NN	NN	NN	GB	GB	GB	GB

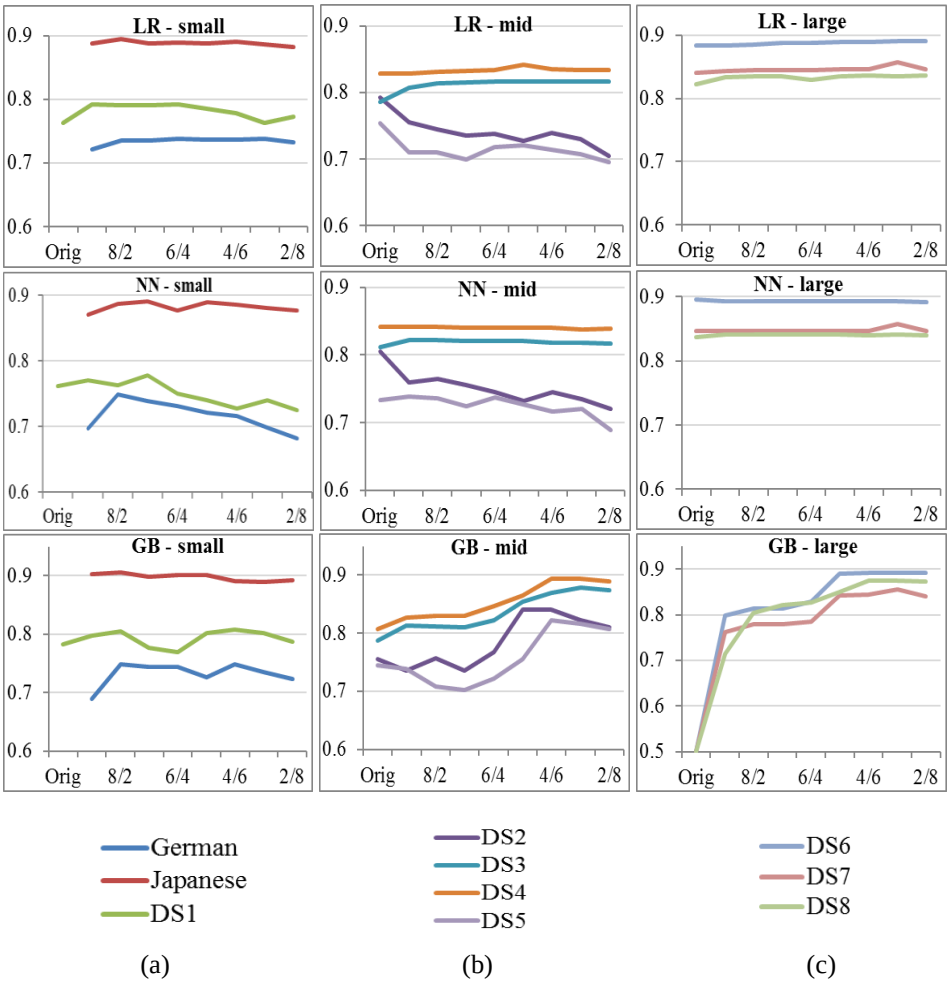

Fig. 4. *AUC* for logistic regression (LR), neural network (NN) and gradient boosting (GB) for (a) small data sets, (b) mid data sets and (c) large data sets

Kendall's coefficient of concordance (*W*) was used to compare the average *AUC* across three classification algorithms and eight class distributions. *W*= 0.24 has shown a low level of agreement for all the algorithms as regarding the ranking of different class

distributions, with *p-value* > .05 = α, not allowing rejection of the null hypothesis that there is no agreement among the different classification algorithms.

In any case, we observed that undersampling increases classification accuracy, with the largest impact observed with the gradient boosting algorithm. The impact of class distribution for samples of different sizes is displayed in Fig. 4.

5.3. Comparison of results across performance metric

The results presented in chapters 5.1. and 5.2. reveal that *GM* and *AUC* in some cases indicate a different imbalance ratio that provides the best performance for the same algorithm and data set. This finding can be explained by the fact that the two measures reflect different aspects of classifiers' performance. *AUC* reflects the ability of the classifier to rank the cases from negative to positive in the correct order. Although having appealing properties, such as averaging the performance over all possible thresholds and being objective (i.e. no subjective input from the user is required), there are some disadvantages to using *AUC*. For example, *AUC* can give misleading results if *ROC* curves cross. It can also provide a somewhat incoherent measure of performance, as *AUC* presumes different misclassification cost distributions for different classifiers [32].

On the other hand, *GM* measures whether a correct binary classification has been made, indicating the balance between classification performances on both classes. As a performance indicator, it is similar to accuracy rate. Even if positive cases are correctly classified, poor prediction accuracy of the negative cases will lower the *GM* value. Consequently, the *GM* measure can be used to avoid overfitting to the positive class, where the negative class becomes marginalized. However, *GM* does not distinguish the contribution of each class to the overall performance, as different combinations of *TPR* and *TNR* can produce the same *GM* value. Also, these types of threshold indicators disregard the absolute values of predictions. If the estimate is higher than the threshold, it is irrelevant what is the actual estimated probability is.

To get a clearer view on the alignment of the two measures, a correlation analysis of the rankings of the imbalance ratios suggested by both measures was performed, outlined in Table 11. The high correlation illustrates overall large agreement between the results for LR and GB on which imbalance ratio would yield the best performance. Somewhat lower agreement can be observed for small data sets. On the other hand, the results for NN are quite different and negative correlation between the measures indicates a disagreement between the measures. Therefore, for NN specifically, *GM* could offer an additional angle from which to prediction accuracy can be assessed.

In general, using different types of evaluation metrics when comparing different classifiers' performance is advisable, because a single metric cannot capture all related important aspects [33]. Nevertheless, when faced with conflicting results presented by *GM* and *AUC*, *AUC* could be considered as more appropriate since it enables measuring performance of a classifier and its discriminatory ability over its entire operating range. Credit institutions stand to benefit a great deal from having relative ranks of the clients determined accurately, since it enables a better differentiation of the degree of credit risk and a more accurate assessment of the portfolio and expected losses [34].

Table 11. Pearson's correlation coefficients among *GM* and *AUC* for imbalance ratios rankings

Data set	LR	NN	GB
German	0.725	0.725	0.405
Japanese	-0.190	0.238	-0.518
DS1	-0.119	-0.232	0.753
DS2	0.048	-0.179	0.946
DS3	0.763	-0.071	0.627
DS4	0.942	-0.286	0.848
DS5	0.429	0.381	0.811
DS6	0.572	0.067	0.811
DS7	0.535	-0.253	0.806
DS8	0.431	-0.313	0.738

6. Empirical results - the impact of sample size and the number of minority class cases

In addition to the impact of the IR, a detailed study of the impact of the sample size was conducted. For each IR, the sample size was decreased by a factor of 10% until reaching 20% of the original size, followed by a reduction of 5% until reaching 5% of the original size. An additional reduction to 3% was performed to illustrate inconsistencies that occur when using very small samples. Obviously, with smaller samples a greater deviation from the original population is introduced.

Based on the results outlined in the previous chapter related to class distribution, the impact of sample size was evaluated for IR ranging from 2.33 (70/30) to 0.43 (30/70), since the best performance for this data set was achieved at these IRs. Thus, for each IR additional 12 data sets of different sizes were created.

6.1. Using geometric mean (*GM*) as performance metric

Fig. 5 presents the impact of varying sample sizes on *GM* under imbalance ratios ranging from 2.33 (70/30) to 0.43 (30/70) by displaying the relative change of *GM* when compared to the sample of the maximum size for each of the algorithms.

The results document several important findings. A general trend of decreasing accuracy when the sample size is decreased can be observed, especially when the sample is more or less balanced. Surprisingly, the impact is much smaller than expected. Also, decreasing the sample size has a much greater impact on NN than LR. In addition to class imbalance, LR has demonstrated low sensitivity to sample size. Decrease in *GM*, compared to using the whole sample with all defaulters (100% of the size) did not exceed 1.5%. These results confirm suitability of using LR with small samples. NN exhibits greater sensitivity to the sample size and number of defaulters, with a much greater decrease in *GM* observed when faced with smaller samples. GB did not exhibit high sensitivity to sample size and seems to be more affected with the class imbalance than the sample size. NN and GB displayed unusual behavior when dealing with samples where

defaulters outnumber non defaulters. It appears that reducing the sample size unexpectedly leads to improvements with and a general trend of increased accuracy.

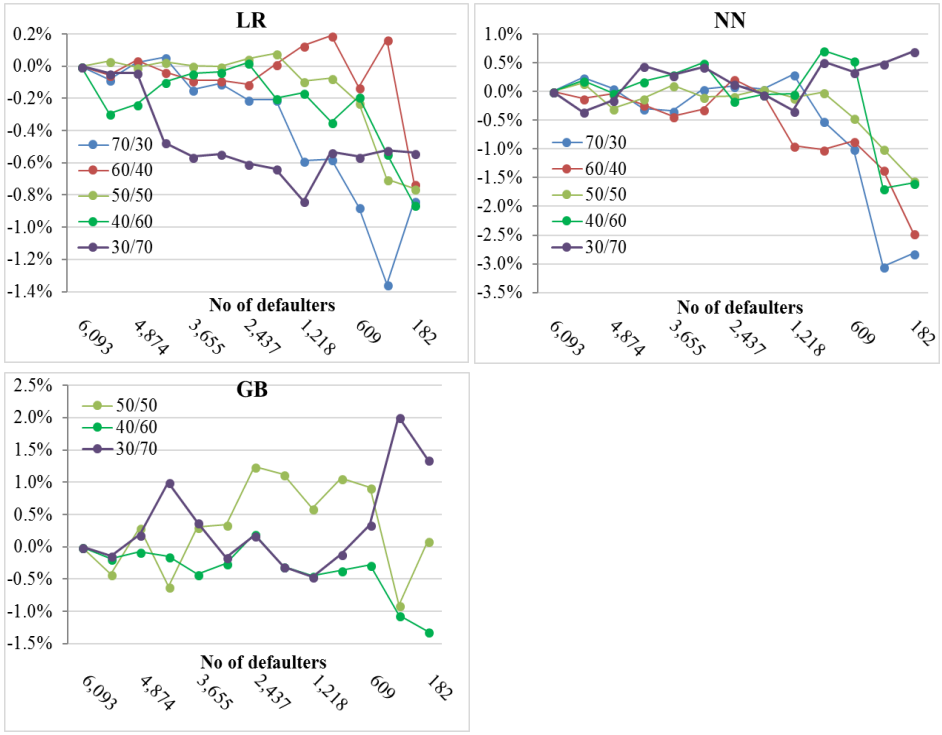

Fig. 5. *GM* % change for IR ranging from 70/30 to 30/70

When measuring performance with *GM*, NN outperforms LR when the good clients outnumber the bad clients, whereas LR outperforms NN when bad clients outnumber the good clients, regardless of the number of defaulters. However, when the sample is balanced, LR outperforms NN when the sample is small (less than 600 defaulters).

6.2. Using Area under the ROC curve (*AUC*) as performance metric

Table 12 presents the impact of reducing sample size to 20% or less of the original size on *AUC* under imbalance ratios ranging from 2.33 (70/30) to 0.43 (30/70). It enables comparison of different algorithms when using the same sample, as well as relative impact on performance when compared to the sample of the maximum size (size=100%) for each of the algorithms. Paired t-tests were conducted to evaluate if the performance when using the reduced sample differs significantly from using the entire sample.

Table 12. *AUC* - Impact of sample size with IR ranging from 70/30 to 30/70

LR											
SIZE	No Bad	IR = 2.33		IR = 1.50		IR = 1.00		IR = 0.67		IR = 0.43	
100%	6,093	0,84	***	0,84	***	0,84	***	0,84	***	0,84	***
20%	1,218	0.84	***	0.84	***	0.84	***	0.84	***	0.84	***
15%	913	0.84	***	0.84	***	0.83	***	0.84	***	0.83	***
10%	609	0,83	*	0,84	***	0,83	***	0,84	***	0,83	***
5%	304	0,83	*	0,83	***	0,83	***	0,83	***	0,83	*
3%	182	0,83		0,83	***	0,83	***	0,83	*	0,83	***
NN											
SIZE	No Bad	IR = 2.33		IR = 1.50		IR = 1.00		IR = 0.67		IR = 0.43	
100%	6,093	0,84	***	0,84	***	0,84	***	0,84	***	0,84	***
20%	1,218	0.84	***	0.84	*	0.84	*	0.84	*	0.84	*
15%	913	0.84	***	0.84	***	0.84	*	0.84	*	0.83	*
10%	609	0,84		0,84	***	0,83	***	0,83		0,83	*
5%	304	0,83		0,83		0,83	*	0,83		0,83	***
3%	182	0,82	*	0,83		0,82		0,81	*	0,82	*
GB											
SIZE	No Bad	IR = 2.33		IR = 1.50		IR = 1.00		IR = 0.67		IR = 0.43	
100%	6,093	0,82	***	0,82	***	0,84	***	0,87	***	0,88	***
20%	1,218	0.82	***	0.82	***	0.85	***	0.87	***	0.87	*
15%	913	0.82	***	0.82	***	0.86	***	0.87	***	0.87	*
10%	609	0,81	*	0,81	***	0,86	***	0,87	***	0,87	*
5%	304	0,82	***	0,82		0,85	***	0,87	***	0,87	*
3%	182	0,81		0,81	*	0,84	***	0,87	*	0,86	*

*** accuracy is not significantly different from the accuracy at size = 100% at α=95%
* accuracy is not significantly different from the accuracy at size = 100% at α=99%

Fig. 6 shows the relative change in *AUC* when compared to the entire sample with all defaulters included graphically.

The results document several important findings. With LR and NN a clear trend of decreasing accuracy when the sample size is decreased can be observed. Surprisingly, the impact is much smaller than expected and in almost all cases accuracy for small sample sizes of 600 or 300 defaulted clients, is not significantly different from the accuracy at size = 100%. This finding is surprising, given that it contradicts the widely recommended using of 1,500 – 2,000 cases. The finding that decreasing the size of the sample with this magnitude does not cause significant changes in performance, represents a novel and very useful discovery for credit scoring. In addition to reducing cost and providing more flexibility when choosing a training sample, it might also encourage practitioners to engage in model development for low default portfolios. For LR the decrease in *AUC*, compared to using the whole sample with all defaulters (100% of the size) never exceeds 1%. In almost all cases accuracy was not significantly different statistically from the accuracy achieved at size = 100%, even for very small samples. NN on the other hand is more affected by the sample size. However, *AUC* did not decrease by more than 4% even when faced with very small samples. When measuring performance with *AUC*, NN again outperforms other algorithms LR when the good clients outnumber the bad clients, whereas GB outperforms NN and LR when bad clients outnumber the good clients,

regardless of the sample size. However, when the sample is balanced, NN outperforms GB when the sample is larger (more than 4000 defaulters). GB did not exhibit high sensitivity to sample size and seems to be more affected with the class imbalance than the sample size. GB displayed unusual behavior when dealing with balanced sample, where reducing the sample size surprisingly lead to improvements in prediction accuracy.

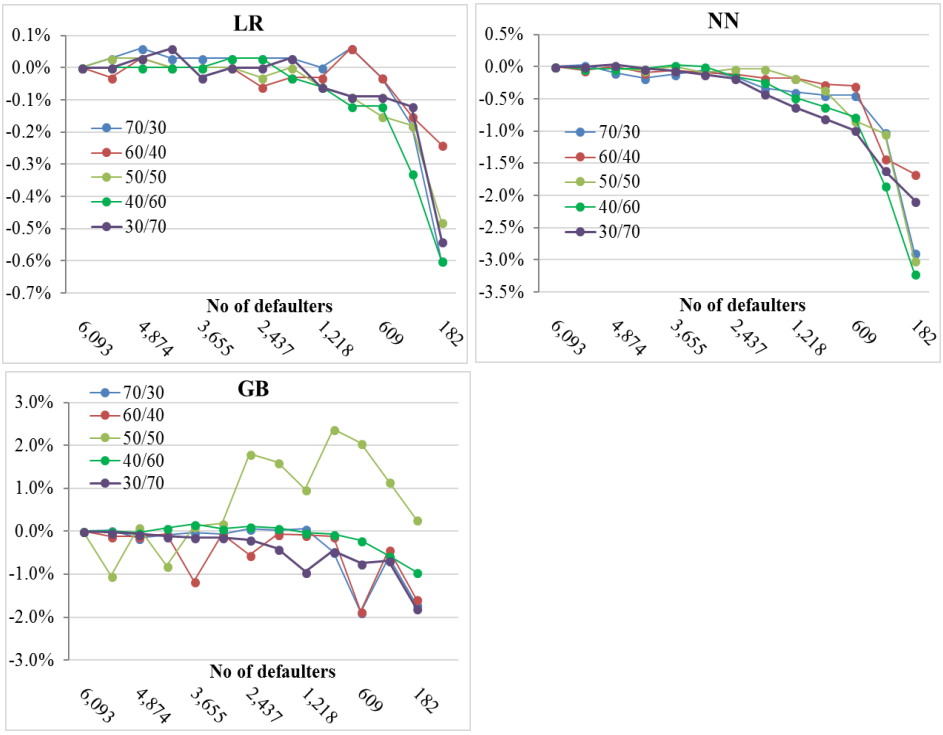

Fig. 6. *AUC* % change for IR ranging from 70/30 to 30/70

The results also confirm that GB is not suitable for dealing with imbalanced data sets, contradicting some of the previous research [17]. LR performs well when using small samples, indicating that samples smaller than the widely accepted benchmark of 1,500 defaulters can be used to develop models of good performance.

7. Conclusion

In this article, a comprehensive experimental study on the effect of sample size and class distribution in credit scoring was presented, using logistic regression, neural network and gradient boosting algorithm on a wide array of real life data samples.

All algorithms have shown a decrease in performance when challenged with higher imbalance ratios. The results suggest that gradient boosting might not be an appropriate classification algorithm when dealing with imbalanced data sets, while logistic regression

and neural networks are fairly insensitive to changes in the class distribution. Neural networks displayed the best performance on average when dealing with higher imbalance ratios. The results also indicate that class imbalance by itself does not necessarily cause a reduction in classification accuracy, and that sample size and classification algorithm play an important role when it comes to determining the optimal class distribution, especially the absolute number of minority class cases.

Classification algorithms have also shown different levels of sensitivity to sample size. With logistic regression and neural network a clear trend of decreasing accuracy was observed when the sample size was decreased. However, the impact was much smaller than expected, suggesting that decreasing sample size significantly, even down to as few as 300 defaulted cases does not cause a significant decline in performance, and represents an important discovery for credit scoring. Gradient boosting did not exhibit high sensitivity to sample size and seems to be more affected by the class imbalance than the sample size.

Given the recent advances in automating lending decision processes, the ever-growing richness in data collected, the big data trend and capital adequacy optimization, we foresee this to remain a very active research area. Leveraging on these findings, we plan to conduct experiments using other classification algorithms, such as random forests, and other, more sophisticated data pre-processing techniques.

References

1. Thomas, L. C., Edelman, D. B., and Crook, J. N., Credit Scoring and Its Application. SIAM, (2002).
2. Schmid, B., Credit Risk Pricing Models: Theory and Practice. Springer Finance, (2011).
3. Hand, D. J. and Henley, W. E., Statistical Classification Methods in Consumer Credit Scoring: a Review, J. R. Stat. Soc. Ser. A (Statistics Soc., vol. 160, no. 3, pp. 523–541, (1997).
4. Yang, Q., 10 challenging problems in data mining research, Int. J. Inf. Technol. Decis. Mak., vol. 5, no. 4, pp. 597–604, (2006).
5. Van Hulse, J. and Khoshgoftaar, T., Knowledge discovery from imbalanced and noisy data, Data Knowl. Eng., vol. 68, no. 12, pp. 1513–1542, (2009).
6. Siddiqi, N., Credit Risk Scorecards, John Wiley Sons, Inc., vol. 1, pp. 1–210, (2006).
7. García, V., Marqués, A. I., and Sánchez, J. S., An insight into the experimental design for credit risk and corporate bankruptcy prediction systems, J. Intell. Inf. Syst., vol. 44, no. 1, pp. 159–189, (2015).
8. Lessmann, S., Baesens, B., Seow, H. V., and Thomas, L. C., Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, Eur. J. Oper. Res., vol. 247, no. 1, pp. 124–136, (2015).
9. Malhotra, R. and Malhotra, D. K., Evaluating consumer loans using neural networks, Int. J. Manag. Sci., vol. 31, no. 2, pp. 83–96, (2003).
10. Ala'raj, M. and Abbod, M. F., A new hybrid ensemble credit scoring model based on classifiers consensus system approach, Expert Syst. Appl., vol. 64, no. July, pp. 36–55, (2016).
11. Feng, X., Xiao, Z., Zhong, B., Qiu, J., and Dong, Y., Dynamic ensemble classification for credit scoring using soft probability, Appl. Soft Comput., vol. 65, pp. 139–151, (2018).
12. Abellán, J. and Castellano, J. G., A comparative study on base classifiers in ensemble methods for credit scoring, Expert Syst. Appl., vol. 73, pp. 1–10, (2017).
13. Fitzpatrick, T. and Mues, C., An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market, Eur. J. Oper. Res., vol. 249, no. 2, pp. 427–439, (2016).

14. Kvamme, H., Sellereite, N., Aas, K., and Sjursen, S., Predicting Mortgage Default using Convolutional Neural Networks, Expert Syst. Appl., vol. 102, (2018).
15. Baesens, B., Gestel, T., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J., Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring, J. Oper. Res. Soc., vol. 54, no. 6, pp. 627–635, (2003).
16. Peng, Y., Wang, G., Kou, G., and Shi, Y., An empirical study of classification algorithm evaluation for financial risk prediction, Appl. Soft Comput., vol. 11, no. 2, pp. 2906–2915, (2011).
17. Brown, I. and Mues, C., An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl., vol. 39, pp. 3446–3453, (2012).
18. López, V., Fernández, A., García, S., Palade, V., and Herrera, F., An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Inf. Sci. (Ny)., vol. 250, pp. 113–141, (2013).
19. Batista, G., Prati, R., and Monard, M., A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explor. Newsl., vol. 6, no. 1, pp. 20–29, (2004).
20. Crone, S. F. and Finlay, S., Instance sampling in credit scoring: An empirical study of sample size and balancing, Int. J. Forecast., vol. 28, no. 1, pp. 224–238, (2012).
21. García, V., Sánchez, J. S., and Mollineda, R. a., On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowledge-Based Syst., vol. 25, no. 1, pp. 13–21, (2012).
22. Weiss, G. M. and Provost, F., Learning when training data are costly: The effect of class distribution on tree induction, J. Artif. Intell. Res., vol. 19, pp. 315–354, (2003).
23. Andrić, K. and Kalpić, D., The effect of class distribution on classification algorithms in credit risk assessment, in 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016, (2016), p. 7.
24. Hosmer, D. W. and Lemeshow, S., Applied Logistic Regression, *A Wiley-Interscience*. p. 375, 2004.
25. Bishop, C. M., Neural networks for pattern recognition, J. Am. Stat. Assoc., vol. 92, p. 482, (1995).
26. Friedman, J. H., Stochastic gradient boosting, *Computational Statistics and Data Analysis*, vol. 38, no. 4. pp. 367–378, 2002.
27. Dheeru, D. and Karra Taniskidou, E., {UCI} Machine Learning Repository. 2017.
28. European Parliament, Regulation (EU) No 575/2013, Off. J. Eur. Union, no. June 2013, pp. 338–436, (2013).
29. European Banking Agency, Guidelines on PD estimation, LGD estimation and the treatment of defaulted exposures, 2016.
30. Werner Dubitzky, Martin Granzow, D. P. B., Fundamentals of Data Mining in Genomics and Proteomics. Springer, (2007).
31. López, V., Fernández, A., García, S., Palade, V., and Herrera, F., An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Inf. Sci. (Ny)., vol. 250, pp. 113–141, (2013).
32. Hand, D. J., Measuring classifier performance: A coherent alternative to the area under the ROC curve, Mach. Learn., vol. 77, no. 1, pp. 103–123, (2009).
33. Japkowicz, N. and Shah, M., Evaluating Learning Algorithms: A Classification Perspective, Eval. Learn. Algorithms. A Classif. Perspect., (2011).
34. Chen, N., Ribeiro, B., and Chen, A., Financial credit risk assessment: a recent review, Artif. Intell. Rev., vol. 45, no. 1, pp. 1–23, (2016).

Kristina Andric received her MSc degree in computing from the Faculty of electrical engineering and computing, University of Zagreb, Croatia in 2006. She is currently pursuing her PhD in computer science at Faculty of electrical engineering and computing, University of Zagreb. Her primary research interests are data science, data mining and machine learning, with particular interest in algorithms used in credit risk management. She is currently head of the Credit risk models and Credit policy and methodology departments at a commercial bank.

Damir Kalpić graduated, received his PhD degree in computer science and worked as a professor at the University of Zagreb Faculty of electrical engineering and computing, Croatia. His main professional interest is in application of operational research in information systems, and in other applications with humans as direct users. He has published more than 100 internationally reviewed papers in journals and conference proceedings and lead about 40 R&D projects for industry, education, medicine, various services and administration. He has advised a few hundred students at student seminars, projects, graduation and PhD theses. He served as vice dean of the Faculty and was the first head of Department of applied computing after its establishment.

Zoran Bohaček graduated and received his PhD degree in computer science at the University of Zagreb Faculty of electrical engineering and computing, Croatia and his master' degree at Harvard University. He worked at McGill University, Bell Northern Research and served as Director of Operations at Fair Isaac International. He is now the Chief Advisor in Croatian Banking Association, where he also served as a Managing Director, and a member of the Management Board of the Croatian credit registry. He is also an Executive Committee member at the European Banking Federation. He has been teaching graduate course on quantitative methods in risk management at the University of Zagreb Faculty of electrical engineering and computing, Croatia.

Received: January 10, 2018; Accepted: October 10, 2018