Area Under the ROC Curve

A comparison of performance metrics for classification problems

—

Radoslaw Bartman & Jonathan Schmitz

Fachbereich Wirtschaftswissenschaften
Spandauer Str. 1
10099 Berlin

Contents

Introduction

In any company decision-making is a key operation among all levels of hierarchy. The application of Machine Learning (ML) methods to support or automate decision-making is becoming a common tool in Business Intelligence. In banking for instance, ML-model based analysis to predict credit default is gaining relevance. The goal of these credit scoring systems is to give a proper estimate of credit risk for potential customers. A new customers data is analyzed by a trained algorithm which classifies the customer into a high or low risk of credit default. Such a classification system would be perfect, if it would classify all default customers as high risk and all non-default customers as low risk. In practice, perfect classification is usually not the case, leading to the problem of misclassification. False predictions result in two problematic cases: giving a credit to a high risk customer and not giving a credit to a low risk customer. Thus the correct evaluation of ML methods is a very important aspect of the modeling process, especially since the process is supposed to support business decisions.

The evaluation af a ML model relies on which metric is employed to asses the method. The *Receiver Operating Characteristic* curve displays the relationship between true positive rate (positive events correctly classified) and the true negative rate (negative events correctly classified), giving aid to select a decision threshold. In case we want to compare different classification algorithms and various model specifications, it would be unhandy to compare various ROC curves. Therefore the single value metric Area Under the ROC Curve (AUC) metric was introduced. In this paper we examime the use of AUC for the evaluation of machine learning algorithms, particulary looking at AUC as a measure of classifier performance. Ultimately we are facing the general problem "of how to accurately evaluate the performance of a system that learns by being shown labelled examples" (Bradley, 1997, pp.3). Any classification problem faces two major decisions in terms of modeling: which classification method should be used and how should that model be evaluated regarding accuracy. However both aspects highly depend on the given data and the type of the problem, which makes it hard to give any kind of guide line for choosing the 'right' estimator and its right 'metric'. Our goal is to show that in case of imbalanced data the ROC/AUC metric has some advantageous properties over qualitative classification metrics, especially accuracy. For our case study, we are evaluating the performance of six different classification algorithms applied on a class imbalanced credit scoring data set. To provide a better basis for comparison we generated two additional balanced data sets through oversampling techniques. For the evaluation we are using 8 different metrics, 3 rank

metrics and 5 threshold metrics. As the AUC metric has the role of a benchmark metric, we particularly focus our comparison on the pairs ROC-AUC/Accuracy and ROC-AUC/PR-AUC. Thereby we aim to investigate the position of AUC within rank and threshold measures. To compare the performance of the metrics themselves, we apply a version of the Friedman test to ROC-AUC and Accuracy and for the comparison of ROC-AUC and PR-AUC we compare their respective graphs.

The paper is organised as follows: in the next Section we look at some related work. In Section 3, we introduce our benchmark metric and other related metrics for classification problems. In the following Sections 4 and 5 we explain the methodology of the experiment and analyze the results. In the last section we draw a conclusion.

Related Work

In this section, we summarize previous literature which is related to our work. As related we consider papers which deal with unbalanced class distribution in machine learning, use of machine learning to solve credit scoring problems, comparison of different metrics and comparison of various classifiers.

There existis a great amount of literature on the comparison of evaluation measures and a lot of research has been done in this field of machine learning. The first notable work in this field is (Flach, 2003) where he compares different metrics using the ROC isometric space. He analysed 4 evaluation measures: accuracy, precision, weighted relative accuracy and F-Measure using decision trees as a classification algorithm. The first large scale comparison of classifiers had been made by (STATLOG, 1995). They compared the performance of different algorithms on twelve datasets from different real-world problems (including credit scoring). Nevertheless, they did not compare performance using different metrics. What is especially interesting for our work, they have two data sets for the classification of credit risk problems.

The first paper which made a large scale empirical comparison of evaluation methods on several classifiers and data sets was published by (Caruna & Niculescu-Mizil, 2004). They noticed that classifiers differ in their goals. Learning methods can perform well on one criterion and perform badly on other criteria. Some classifiers, for instance boosted trees, perform well on metrics like ROC-AUC and accuracy but perform badly on probabilistic classifiers. Caruana and Niculescu-Mizil suggest choosing metrics depending on which goals we want to achieve. They also provide the

existence of measures for general purpose metric as such metrics to use when more specific criteria are not known. We can use the root-mean-squared-error (RMS) or a newly created measure called SAR, which is a simple combination of the most popular measures of the three metric types (accuracy, AUC-ROC and RMS).

There has been subsequent work comparing performance measures for classification, see (Fuernkranz & Flach, 2005), (Buja et al., 2005) and (Huang & Ling, 2007). However, all of these papers rather focus on theoretical aspects then practical applications.

A further important branch of empirical research was constituted through (Caruna & Niculescu-Mizil, 2006) and (Ferri at al., 2009). The first paper compared performance measures and the second compared learning algorithms. In (Ferri at al., 2006) eleven supervised learning methods were compared, some of the predictions were calibrated with Platt Scaling and Isotonic Regression. For evaluation the same evaluation measures as (Caruna & Niculescu-Mizil, 2004) were employed. The performance of some of the classifiers improved dramatically due to the calibration (boosted trees, or SVMs) and for others (Neural nets or Logistic Regression) no improvement was noticed.

(Ferri at al., 2009) has been the largest comparison of metrics as so far. They compared eighteen different evaluation metrics on 6 learning methods. This comparison is remarkable, if we consider that there are 35 data sets, whereas some of them deal with multiple classification problems. Furthermore the analysis of results was specified by regarding properties of the employed data sets, for instance if a data set contains a two-class or multiclass outcome, if its size is small or large or if it its outcome variable is balanced or imbalanced. Their results happened to be complementary to the work of (Caruna & Niculescu-Mizil 2004). In our interes lies their result of observing a large difference in correlation between measures concerning class distribution. Some measures behave differently if the class distribution is imbalanced, compared to when it is balanced.

Furthermore there are various papers which perform comparison of measures for classification related to a specific field of application. (Liu & Shriberg, 2007) for instance compare evaluation measures for sentence boundary detection with the use of IT-specific classifiers; (Jeni et al., 2011) compare metrics in recognizing facial expression.

In this paper we compare evaluation measures on an imbalanced dataset in a credit scoring context. The first major work in empirical evaluation and comparison of classifiers was introduced by (Baesens et al., 2003). They compared classifiers and

Table 1: Literature table

Author & publishing year	# Data sets	# Classification Methods	Classification Type	# Metrics
Baesens et al. 2003	8	17	binary	2
Caruana & Niculescu-Mizil 2004	7	7	binary	10
Sokolova et al. 2006	1	2	binary	5
Caruana & Niculescu-Mizil 2006	11	10	binary	8
Liu & Shriberg 2007	2	5	binary	6
Hulse et al. 2007	35	11	binary	7
Ferri et al. 2009	30	6	binary & multiple	18
Nanni et al. 2011	3	4	binary	4
Jeni et al. 2013	3	6	binary	6
Lessmann et al. 2015	8	41	binary	6

the performance of various state-of-the-art classification algorithms applied to eight real-life credit scoring data sets. The evaluation of these algorithms was made by two measures ROC-AUC and accuracy.

A similar approach in credit scoring problems was made by (Brown & Mues, 2012). They analysed seven different classifiers but only on imbalanced datasets. They concluded that the gradient boosting and random forest classifiers yield a very good performance at extreme levels of class imbalance, whereas the SVM sees a reduction in performance as a larger class imbalance is introduced. They evaluated the classifiers with the AUC-ROC measure.

(Lessmann et al., 2015) published an actualisation of the general findings of data mining in credit scoring problems. It contains a summary of empirical studies on credit scoring problems until 2015. The empirical research was performed with a larger number of datasets, more developed classifiers algorithms and evaluation methods. Finally, the overall best results were achieved for an ensemble selection approach that integrates the principles of bootstrap sampling with a greedy hill-climbing algorithm.

For a concise overview on the related literature see Table 1.

Methodology

In this section, we introduce three types of performance metrics, of which we employ two typesin our analysis. We especially focus on our AUC-ROC measure, as it is the

benchmark metric of our analysis. We briefly describe the details of this measurement tool.

In our work we will use eight different performance measures: Area under the Receiver Operating Characteristic Curve (AUC-ROC), Area under the Precision-Recall Curve (AUC-PR), Average Precision (APR), Accuracy (ACC), Precision (PRE), Recall (REC), F-Measure (FME) and Kappa Statistic (KAP). Those metrcis are usually categorized into two groups: threshold metrics and rank metrics. In our experiment we will particularly focus on the comparison of ROC-AUC/Accuracy and ROC-AUC/PR-AUC, to cover both type of metrics.

Rank metrics

Rank metrics quantify the quality of rankings. Rank metrics depend only on the ordering of the predictions, not the actual predicted values. If the ordering is preserved it makes no difference if the predicted values range between 0 and 1 or any other range in this interval, e.g. between 0.50 and 0.51. We use three of those metrics in our work.

AUC-ROC

The *Receiver Operating Characteristic* investigates and employs the relationship between sensitivity (the true positive rate) and specificity (true negative rate) of a classifier. The ROC curve is a plot, where sensitivity on the y-axis is plotted against $1-$specificity (the false positive rate) on the x-axis. To calculate the area under the ROC Curve is the most common method to interpret the ROC curve because it represents the expected performance to a single value.

$$Sensitivity = \frac{TP}{TP + FP} \tag{1}$$

$$Specificity = \frac{TN}{FP + TN} \tag{2}$$

The ROC-AUC is an ordering metric, which measures how well positive cases rank above negative cases. It does not depend on the actual predicted values but only on the ordering of the predictions. The values of the AUC are in an interval between 0 and 1. $AUC = 1$ is the best possible result and means that the classifier scores every positive value higher than every negative. Accordingly, if the $AUC = 0$ is the

worst possible result and imply that the classifier scores every negative higher then every positive. The AUC statistic is similar to the Gini coefficient, which is equal to $2 \times (AUC - 0.5)$.

A more detailed explanation of the model can be found in (Flach, 2016) and (Fawcett, 2006).

AUC-PR

AUC-PR is calculated in a similar way like the AUC-ROC curve. The difference is that the Precision-Recall curve plots precision on the y-axis with recall on the x-axis. The definition of precision and recall is given below. For more information see (Boyd et al, 2013).

Average Precision

APR summarizes a Precision-Recall Curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$APR = \sum_{n} (R_n - R_{n-1}) P_n \tag{3}$$

Where P_n and R_n are the precision and recall at the nth threshold.

Threshold metrics

Threshold metrics are sensitive to a good choice of threshold. It is not important how close a prediction is to a threshold, only if it is above or below a threshold. These measures are preferable if a model is supposed to minimize the number of errors. We use five of them in our work.

Accuracy

Accuracy is the percentage of the correctly classified positive and negative examples. It approximates how effective the algorithm is by showing the probability of the true value of the class label. It assesses the overall effectiveness of the algorithm:

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

Precision

Precision is defined as the percentage of correctly classified positive examples in examples that were classified positive. It estimates the predictive value of a label, either positive or negative, depending on the class for which it is calculated; in other words, it assesses the predictive power of the algorithm:

$$PRE = \frac{TP}{TP + FP} \tag{5}$$

Recall

Recall is defined as the percentage of correctly classified positive examples in all positive examples. It approximates the probability of the positive (negative) label being true; in other words, it assesses the effectiveness of the algorithm on a single class.

$$REC = \frac{TP}{TP + FN} \tag{6}$$

F-Measure

F-Measure is the harmonic mean of precision and recall.

$$FME = \frac{2 \times PRE \times REC}{PRE + REC} \tag{7}$$

Kappa

Kappa Statistic indicates the proportion of agreement beyond that expected by chance, that is, the achieved beyond-chance agreement as a proportion of the possible beyond-chance agreement. It takes the form:

$$KAP = \frac{P_o - P_c}{1 - P_c} \tag{8}$$

Where P_o is the proportion of observed agreements and P_c is the proportion of agreements expected by chance (Sim & Wright 2005).

Furthermore there exists a third type of measure - probability metrics. These measures quantify the deviation of the estimated probability with respect to the actual probability. We can interpret the predicted value of each case as the conditional probability of that case. For our comparison we are not taking this kind of measure into account, since the other two types are recommended to use for the evaluation of binary classification problems (Kuhn, & Johnson, 2013, Ch. 11), (Branco et al., 2016).

Table 2: Variable summary

skim_variable	n_missing	numeric.mean	numeric.sd
default	0	0.067	0.251
unsecure_lines	0	0.000	1.000
age	0	0.000	1.000
nr_past_due30	0	0.000	1.000
debtratio	0	0.000	1.000
monthlyincome	0	0.000	1.000
nr_open_credits	0	0.000	1.000
nr_re_loans	0	0.000	1.000
nr_family	0	0.000	1.000
profit	0	56.292	3653.142

Experimental Design

The data set used for our experiment was obtained through the course instructor. It contains customer data of some financial institution and the share of good clients amounts about 93 percent. A bad customer is defined by two characteristics:

DEFAULT A person who experienced 90 days past due delinquency or worse.

PROFIT A person who generates a negative cash flow, approximated through total cash flow of the loan

To perform our experiment we choose DEFAULT as our target variable. A summary of the data set is given below in Table 2. It appears to be that the data set is already prepared for modeling; there are no missing values, all variables are numeric and all variables, except the target variables, are standardized. In Figure 1 we see a visualization of the severe class imbalance of DEFAULT and a correlation plot of default and all predictor variables, showing a low correlation over all. Since the data set is already relatively prepared we only applied a few further preprocessing steps: some variables were renamed, PROFIT was dropped and DEFAULT was transformed into a factor variable.

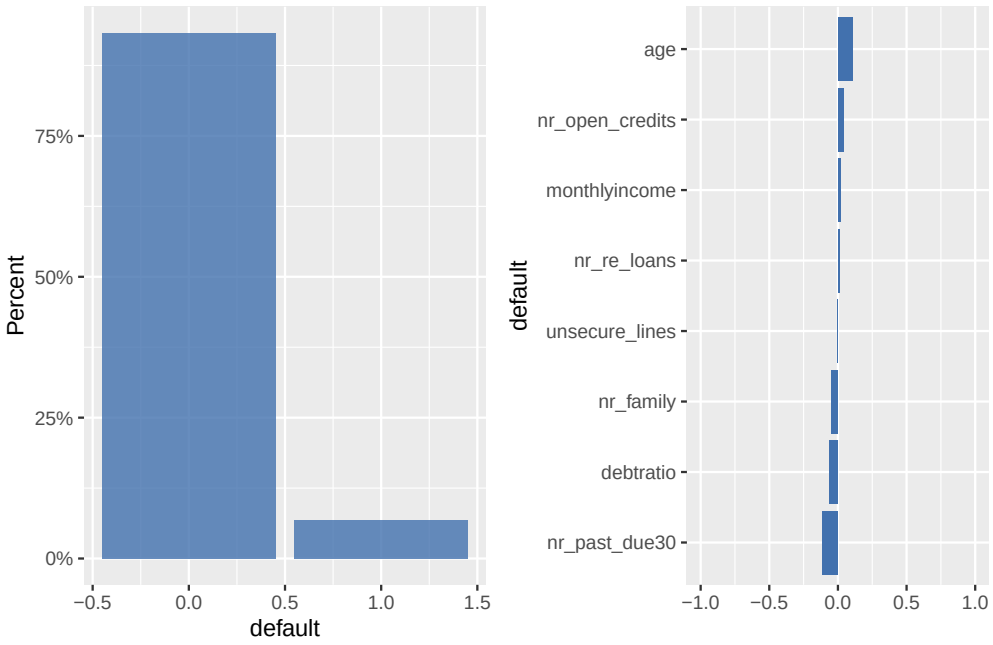

Figure 1: Class Imbalance and Correlation

For the modeling process we split the data set into a training and a test set (with a ratio of 75/25). Due to the severe class imbalance the sample was stratified by the target variable. Additionally we further splitted the training set into a training and a validation set (ratio 80/20) for tuning model parameters. We did not use N-fold cross validation due to computational restrictions. Furthermore we created two additional data sets through over-sampling, since as it is our goal to compare the performance of the AUC and other metrics, especially Accuracy and PR-AUC. Therefore we used two different over-sampling techniques, *down-sampling* and *SMOTE*. Down-sampling involves randomly removing observations from the majority class to prevent it from dominating the learning algorithm. SMOTE (Synthetic Minority Oversampling Technique) is an up-sampling technique, which generates new synthetic instances from existing minority cases. Both techniques result in a perfectly balanced data set. The raw test set contains 1954 bad and 26275 good customers, for the down-sampled set both classes contain 1954 observations and for the SMOTE set both contain 26275 cases. By using differently balanced versions of the data set we extend the range of our performance test, since we can now compare the performance of each metric for balanced and imbalanced data.

The goal of our paper is to explore the AUC metric und compare it with other metrics. Therefore our experiment aims to compare the performance of a set of performance metrics applied in the context of credit scoring. ML algorithms are employed in credit scoring to solve a classification problem: on the basis of available data, every customer is assigned a binary label, as for instance 'customer likely defaults' or 'customer likely not defaults'. Since a classification model results in a binary outcome, the target variable y can either take on the value $y = 1$ if the customer likely defaults, or $y = 0$ if the customer likely not defaults. To model the response based on our data sets we selected six classifiers which are commonly used for credit scoring (Baesens et al., 2003), (Brown & Mues, 2012). For the modeling and tuning process we used the `tidymodels` framework in R; each classifier is tuned on a random grid of 25 different hyperparameters (or hyperparameter combination, if there is more then one parameter to tune). A description of the workflow for each model is supplied in the accompanying R-script. We chose the following modeling techniques and grid settings:

Penalized Logistic Regression (LR)

The logistic regression model contains two tuning parameters: a penalty value and a mixing parameter. We set the mixing parameter $\alpha = 1$ and tuned the penalty on a sequence of the range $[0.0001, 0.1]$. For more information see (Friedman et al., 2008).

Random Forest (RF)

The random forest model contains three tuning parameters: a value for the number of predictors that will be randomly sampled at each split, a value for the number of trees and a value for the number of data points that are required for a node to split. We set the tree parameter $= 1000$, the first parameter was tuned in the range of $[2, 8]$ and the last one in $[2, 40]$. A more detailed explanation of the model can be found in (Breiman, 2001).

Support Vector Machine (SVM)

The SVM-model uses a radial basis kernel function for training and prediction and contains two tuning parameters: the cost of predicting a sample within or on the wrong side of the margin and a precision parameter for the radial basis funcion. The first parameter was tuned in the range of $[0.25, 128]$, the second in $[0.001, 0.01]$. For more information see (Noble, 2006).

Multi-layer Perceptron (MLP)

For our MLP-model (Neural Network) we choose the penalty parameter over dropout, which leaves us with four tuning parameters: a value for the number of units in the hidden layer, a penalty value for the amount of weight decay, a value for the number of training iterations (epochs), a type of activation function, which connects hidden layer and input variables. The hidden units parameter was tuned in the range of $[1, 10]$, the penalty value in $[0.0001, 0.1]$; batch size was fixed set 20 and as activation function we chose relu. A more detailed explanation of the model can be found in (Atiya, 2001).

Gradient Boosting (XGB)

The gradient boosting model uses the XGBoost algorithm and contains a total of 7 hyperparameters of which we tuned the following four: the number of predictors at each split, in the range of $[2, 16]$; the number of trees, in the range of $[1000, 2000]$; the maximum depth of a tree, in the range of $[1, 5]$; and the rate at which the boosting algorithm adapts from iteration to iteration, in the range of $[0.01, 0.1]$. For more information see (Chen & Guestrin, 2016).

Naive Bayes (NB)

The naive bayes model contains two tuning parameters: a regularization value and a correction parameter for smoothing low-frequency counts. The first parameter was tuned in the range of $[0.5, 1.5]$, the second in the range of $[0, 3]$. A more detailed explanation of the model can be found in (Lewis, 1998).

All 25 specifications of each model were evaluated using the validation set and each model was applied to all three data sets, resulting in 125 models per data set. The 25 specifications of each of the six models were then evaluated by eight performance metrics and the best value for each classification model and metric was selected, getting us 48 results for each data set.

To Asses the performance of multiple classification models on multiple datasets it is recommended to perform a statistical test for the chosen evaluation metric (Japkowicz & Shah, 2011). For our comparisson of AUC and accuracy we therefore conducted two non-parametrical tests on each metric, following the approach of (Demšar, 2006), and compared the results for both metrics. However, as the number of data sets and algorithms in our experiment was rather small, we followed the recommendation of

(Garcia et al., 2010) using Friedman's Aligned Rank test instead of the regular Friedman test; instead of Nemenyi's post hoc test we used Friedman's Aligned Ranks post hoc test to assess all the pairwise differences between algorithms.

In Friedman's Aligned Rank test "a value of location is computed as the average performance achieved by all algorithms in each data set. Then, it calculates the difference between the performance obtained by an algorithm and the value of location" (Garcia et al., 2010, p.2051). After repeating this step for each algorithm and data set, the so called *aligned observations*, the resulting differences, are then ranked from 1 to kn relative to each other. "Then, the ranking scheme is the same as that employed by a multiple comparison procedure which employs indepen-dent samples; such as the Kruskal–Wallis test" (Garcia et al., 2010, p.2051). This results in the following test statistic, which is compared with a chi-square distribution for $k-1$ degrees of freedom:

$$T = \frac{(k-1)[\sum_{j=1}^{k} \hat{R}_{\cdot j}^2 - (kn^2/4)(kn+1)^2]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k)\sum_{i=1}^{n} \hat{R}_{i\cdot}^2} \tag{9}$$

Then $\hat{R}_{\cdot j}$ is the rank total of the jth algorithm and equals $\hat{R}_{i\cdot}$ the rank total of the ith data set. Furthermore (Garcia et al., 2010) recommend to proceed with a post hoc test, if the null hypothesis is rejected. Friedman's Aligned Rank post hoc test is based on a p-value comparison, where thoses values are obtained from the Friedman's Aligned Rank test; A more detailed explanation of both tests can be found in their paper. For our experiment we used the test functions provided by the `scmamp` package.

Therefore we statistically compared all 48 results of each metric and tested the significance of difference in rank between the individual classification models using the mentioned tests. For comparison of AUC-ROC and PR-AUC we simply looked at the plotted curves.

Empirical Results

In this section we will look at the results of our experiment, comparing the performance of our set of metrics and particularly discuss the differences between the pairs of Accuracy & ROC-AUC and PR-AUC & ROC-AUC.

Table 3: Mean difference per classifier for each pair of data sets

Metrics	Raw & Down	Raw & SMOTE	SMOTE & Down
Accuracy	0.143	0.086	0.057
Avg Precision	-0.025	-0.012	-0.013
F-Measure	-0.202	-0.218	0.016
Kappa	-0.134	-0.160	0.027
AUC-PR	-0.004	0.029	-0.033
Precision	0.311	0.233	0.077
Recall	-0.678	-0.571	-0.108
AUC-ROC	-0.036	-0.027	-0.009

General findings

Table 4 (on the following page) reports the best values for each metric and classifier model and for each data set. Probably the most obvious result is the impact of class imbalance on most metrics. For instance the values for precision and recall altered significantly, with the range of values for recall changing from $(0.003, 0.16)$ for inbalanced data to $(0.46, 0.91)$ for balanced data. A similar result holds for the kappa metric, for imbalanced data it does not reach the 0.2 threshold once, whereas it is in the $(0.2, 0.3)$ for most classifiers in the balanced data sets. Almost the same holds for the F-measure, except that it reaches 0.221 for one classifier. If we group those metrics which perform significantly different on imbalanced and those which do not, we encounter an earlier made categorization: threshold metrics show siginificant performance variance, whereas ranking metrics do not. In Table 3 we measured the difference of each metric value for each data set $(DS^1_{ij} - DS^2_{ij})$. It clearly shows that for threshold metrics the deviation for each measure never is higher then about 3.5%, it mostly is below. For threshold metrics the table shows great variation between imbalanced and balanced data sets. This result is not very surprising, as it is well documented that threshold metrics do not perform well on imbalanced data (Jeni et al., 2013), (Ferri et al., 2009).

Table 4: Best metrics for each data set

Metrics	LR	RF	SVM	MLP	XGB	NB
Raw data						
Accuracy	0.933	0.935	0.933	0.935	0.935	0.933
Avg Precision	0.204	0.331	0.171	0.285	0.332	0.254
F-Measure	0.016	0.189	0.016	0.100	0.221	0.005
Kappa	0.014	0.168	0.014	0.088	0.185	0.003
AUC-PR	0.533	0.328	0.170	0.284	0.330	0.253
Precision	0.500	0.667	0.500	0.667	0.576	0.167
Recall	0.008	0.116	0.008	0.056	0.164	0.003
AUC-ROC	0.723	0.840	0.630	0.799	0.841	0.788
Down-sampled data						
Accuracy	0.754	0.758	0.784	0.772	0.765	0.910
Avg Precision	0.262	0.331	0.274	0.283	0.321	0.256
F-Measure	0.263	0.301	0.287	0.282	0.298	0.330
Kappa	0.174	0.216	0.204	0.196	0.214	0.270
AUC-PR	0.263	0.329	0.272	0.487	0.320	0.254
Precision	0.164	0.186	0.184	0.177	0.186	0.315
Recall	0.728	0.786	0.759	0.913	0.780	0.458
AUC-ROC	0.782	0.840	0.797	0.796	0.837	0.789
SMOTE-sampled data						
Accuracy	0.721	0.910	0.783	0.818	0.931	0.924
Avg Precision	0.259	0.296	0.283	0.288	0.282	0.240
F-Measure	0.251	0.355	0.289	0.302	0.349	0.310
Kappa	0.159	0.293	0.205	0.224	0.294	0.260
AUC-PR	0.260	0.294	0.281	0.373	0.279	0.238
Precision	0.153	0.319	0.184	0.202	0.446	0.373
Recall	0.728	0.550	0.738	0.741	0.471	0.550
AUC-ROC	0.778	0.826	0.796	0.800	0.818	0.767

Problems of threshold metrics: Accuracy

Especially the measure of accuracy is problematic for imbalanced data, because accuracy only reports the share of correct responses, consisting in the sum of true positive and true negative values. If the outcome class is severly unbalanced, for instance following a distribution of 99% negatives to 1% positive, then we could just guess the majority class and we would obtain a 99% accurate classifier. This is not surprising, as accuracy by default assumes a 50% threshold between a positive and negative classification. If we now look at Table 4, we see an accuracy of about 93% for all classifiers. Figure 1 in the last section reminds us of the class distribution of the outcome variable: about 7% of the clients were defaulters. Most likely the accuracy for all classifiers applied on the imbalanced data just mimiced the class distribution of default. In case of credit scoring this could be highly problematic, since accuracy does not hold much information on the False Negative rate, those customers which were predicted as 'non-default' but who were actually defaulters. In our data set the average default generates cost of $\$11145.56$, the average non-default results in a profit of $\$866.61$. Minimizing the amount of false negatives by only a few percent would decrease a lending companies loss on credit default significantly.

Statistical comparison

If we now reconsider the question of model selection for production, it would be impossible to draw any conclusion on the imbalance data set only by looking at accuracy. A different result occurs for the AUC metric. As we can see in Table 4, regarding

Table 5: Friedman Aligned Rank test

Metric	statistic	p.value	parameter
AUC	11.58	0.04	5
Accuracy	6.31	0.28	5

AUC, the random forest model and the gradient boosting classifier perform better then all of the other classifiers and they due so in any of the data sets. To statistically test if there is a difference in performance we applied the Friedman Aligned Rank test for Accuracy and AUC; H_0 is defined 'All algorithms perform the same'. For the results of the tests for both metrics see Table 5 below. We can reject the null hypothesis for

AUC, but not for Accuracy. Hence, statistically there is no performance difference among the selected algorithms, if we chose accuracy as evaluation metric.

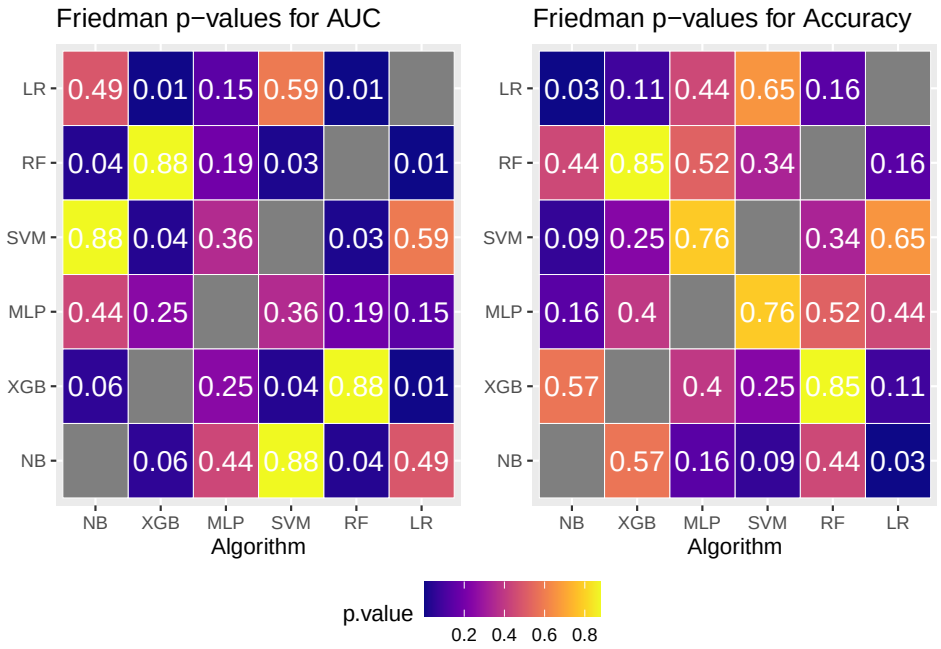

Figure 2: Post hoc test

To illustrate those findings we applied the Friedman Aligned Rank post hoc test to both metrics, even though we could not reject the null hypothesis for accuracy. Taking a look at all pairwise differences between algorithms and for both metrics, we can see in Figure 2. that there are no recognizable differences of performance, except for one pair, if accuracy is the chosen evaluation metric. For AUC on the other hand, we obtain 5 significant differences, which were to expect, since we saw two better performing models in all of the data sets. From Figure n we can see, that the random forest classifier outperforms logistic regression, SVM and naive bayes with $p < 0.05$; same holds for the gradient boosted classifier, except that the p-value for naive bayes is 0.06. It is also not surprising that there is a high correlation between those two classifiers, since their mechanism is very similar.

Rank curves: ROC & AUC-PR

Finally we compare the ROC-curves and the Precision-Recall-curves for all classifiers and data sets. In Figure 3 we can see the difference in performance for balanced and imbalanced data, in both type of curves. Both curves show the advantage, which rank metrics have over threshold metrics: instead of just delivering a single point estimate, we obtain an entire curve of the classifier's performance, which gives us much more information. Therefore we gain flexibility, because the instances get ranked by score first and we can later decide what threshold to use. As in credit scoring our potential goal is to minimize cost, which could mean to minimize the false-negative rate, it is advantagous to have a metric, which supports the finding of good score threshold for deciding positive and negative instances.

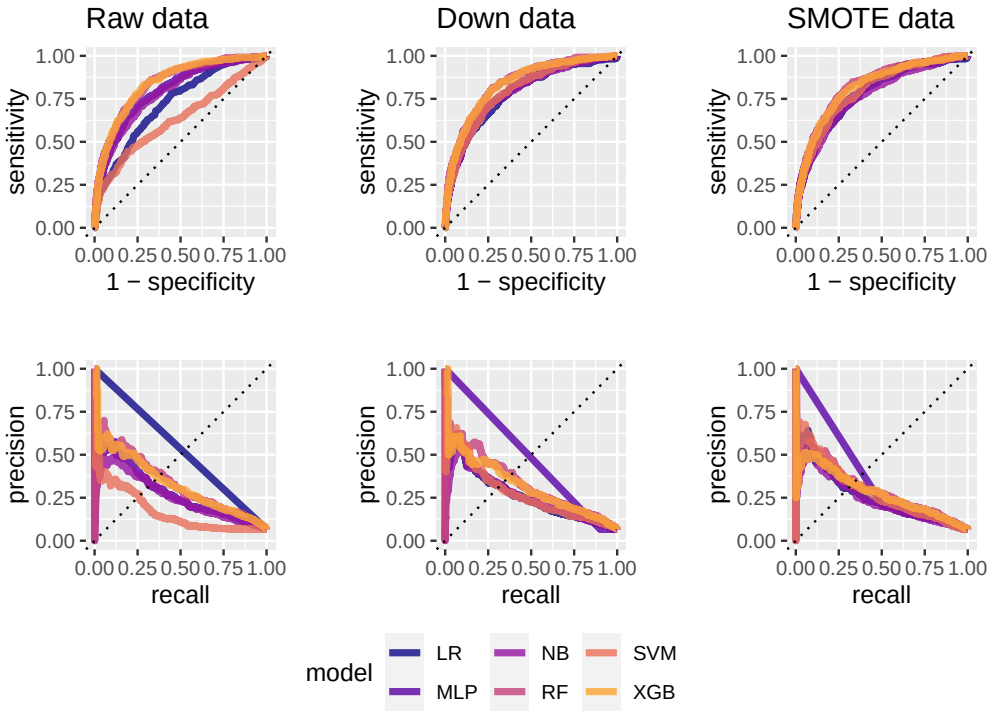

Figure 3: ROC curves (top) and PR-curves (bottom) for all data sets

However if we compare the values of AUC and PR-AUC regarding the question of model performance, we can see in Table 3, that AUC always ranks random forest and gradient boost the highest, but PR-AUC favors logistic regression for imbalanced data and the multi-layer perceptron for balanced data. Keeping in mind that ROC

covers both classes, whereas PR-curve focuses on the minority class, it could be the case that ROC may provide a too optimistic view of the performance (Branco et al., 2016). The curves for both metrics depict this tendency. Regarding the aim of the model it therefore would be necessary to investigate this finding further; one option to optimise performance would be to alternate the threshold for the AUC.

Conclusion

In this study we used a real-world based credit scoring problem to compare different performance metrics to evaluate a number of binary classification models. For this purpose we extended our testing data through resampling techniques. Our benchmark metric in this comparison was AUC, our aim was a general comparison of alternative metrics, but we especially focused on comparing AUC with accuracy and AUC-PR.

Therefore we examined ranking and threshold metrics and our findings generally confirm the expectation we build due to our investigation in related works. The combination of ROC Curve and AUC metric has shown some advantageous properties as a classification performance measure, especially compared to accuracy: it gives an idea of how well separated negative and positive classes are for the decision threshold; it does not depend on a decision threshold; it gives us more information then a single value and therfore more flexibility; and, at least for this kind of data, AUC measured performance differences have been assured statistically.

Though the ROC-AUC metric is a common metric, in particular when it comes to class imbalance, it is important to keep in mind, that there is a potential danger of AUC to give a too optimistic evaluation. As we underlined in the comparison of ROC and PR-AUC, ROC focusses on both classes. This can lead to unreliable estimates "when the problem of class imbalance is associated to the presence of a low sample size of minority instances" (Fernández et al., 2018, p.55). For the field of credit scoring, but in general any kind of ML application field which faces severe class imbalance, it is could be useful to drive the development of AUC/ROC forward by customizing the metric for the one needs and pecularities. As stated by (Brown & Mues, 2012), it would be benifical to investigate further in the effect of over-sampling techniques and its effect on credit scoring. In this regard our paper showed, in a limited scope, that in case of performance evaluation ranking metrics have a certain advantage compared to threshold metrics.

References

[1] Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on neural networks*, 12(4), 929-935.

[2] Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6), 627–635.

[3] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7), 1145-1159.

[4] Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

[5] Boyd, K., Eng, K. H., & Page, C. D. (2013). Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 451-466). Springer, Berlin, Heidelberg.

[6] Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2), 1-50.

[7] Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3), 3446-3453.

[8] Buja, A., Stuetzle, W., & Shen, Y. (2005). Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, November, 3.

[9] Caruana, R., & Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 69-78).

[10] Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).

[11] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[12] Demšar, J. (2006.) Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.

[13] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.

[14] Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1), 27-38.

[15] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (pp. 1-377). Berlin: Springer.

[16] Flach, P. A. (2003). The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 194-201).

[17] Flach, P. A. (2016). ROC analysis. In *Encyclopedia of Machine Learning and Data Mining* (pp. 1-8). Springer.

[18] Friedman, J., Hastie, T. & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software*, Vol. 33(1), 1-22.

[19] Fürnkranz, J., & Flach, P. A. (2005). Roc 'n'rule learning—towards a better understanding of covering algorithms. *Machine Learning*, 58(1), 39-77.

[20] Huang, J., & Ling, C. X. (2007). Constructing New and Better Evaluation Measures for Machine Learning. In *IJCAI* (pp. 859-864).

[21] Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

[22] Jeni, L. A., Cohn, J. F., & De La Torre, F. (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction* (pp. 245-251).

[23] King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, 9(3), 289-333.

[24] Kuhn, M., & Johnson, K. (2013). *Applied predictive modelling Springer*. New York Heidelberg Dordrecht London.

[25] Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4-15). Springer, Berlin, Heidelberg.

[26] Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1), 124-136.

[27] Liu, Y., & Shriberg, E. (2007). Comparing evaluation metrics for sentence boundary detection. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07* (Vol. 4, pp. IV-185). IEEE.

[28] Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology*, 24(12), 1565-1567.

[29] Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence* (pp. 1015-1021). Springer, Berlin, Heidelberg.

[30] Sim, J., & Wright, C. C. (2005). The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3), 257-268.