Evaluating Performance of Six Machine Learning Algorithms to Predict Creditworthiness of Bank Loan Customers

Tanvi Chheda and Philipp Kogelnik

ABSTRACT

Six Machine Learning Algorithms are used to predict the creditworthiness of loan seekers- Naïve Bayes, Linear Regression, Decision Trees, Random Forests, Extremely Randomized Trees and Support Vector Machines. Parameters such as type of kernel, measure of information etc. are tuned to find the best performer. Predictions are cross validated by dividing data into training and test samples, accuracy of 70-75% is obtained. Relative feature importance gives an insight into characteristics of a person that would determine his/her borrowing and payment behavior. Algorithms are compared on the basis of computation time, stability, accuracy, F1 score and interpretability.

INTRODUCTION

Creditworthiness- the probability of a person meeting his/her payment obligations in future is usually determined from credit history: a record of successfully paid credits, defaults, insolvencies, etc. Although agencies that provide personal credit scores such as FICO in the US and SCHUFA in Germany maintain their algorithm to predict risk a trade secret [1], it is known that they use demographic and socio-economic data such as age, stability at employment and residence, savings and stock holdings, etc. in addition to payment history. The goal of their algorithms is to identify patterns in data, formulate strategies and provide objective decision support to minimize risk and maximize profit.

This paper gives some insights in how Machine Learning Algorithms (MLAs) can be used to predict creditworthiness based on payment history, demographic data and socio-economic data of clients. To accomplish this task different methods are compared based on both qualitative and quantitative metrics.

Which machine learning methods are used for comparison and which problems have to be tackled for the analysis can be found in the section, “Conceptualization”. The data analysis pipeline and methodology is explained in detail in the section “Methodology”. This section also includes information about the data used for the experiments.

Outcomes of the different methods based on different metrics are provided and discussed in the section “Results” and a

bottom line about the work is given in the last section - “Conclusion”.

CONCEPTUALIZATION

As already mentioned the goal of this work is to compare different methods in terms of performance for prediction of creditworthiness. According of the “No free lunch” theorem there is no methods which works best for every problem. [2] Therefore, we selected six different machine learning algorithms to find out which one works best for this type of application: Linear Regression (LR), Decision Tree (DT), Random Forest (RF), Extremely Randomized Trees (ERT), Naïve Bayes (NB) and Support Vector Machine (SVM).

One thing we want to investigate is the usefulness of less interpretable models in this context. Businesses most of the time prefer an easy model over more complex ones. In our experiments LR, DT and NB can be seen as easier interpretable models. RF and ERT internally consist of multiple trees, which makes it already harder to interpret. [3] SVM is the most complex classifier and it is difficult to understand how it works internally because of the mathematical complexity. [4]

Training and testing of the classifiers is done in two different ways. The first one uses a fixes split of training and testing data (hold-out validation). The second one uses k-fold cross validation, where we divide the data in *k* separate bags. Then k iterations are performed, every time k-1 bags are used for training and the remaining bag for testing. In the end the results get averaged.

When using a hold-out validation, there are some problems which might occur. With fixed train and test sets some parts of the data are never used for training, which causes the training data to be smaller. This can be an issue with just a small dataset available. Moreover, the performance of the classifiers with fixed sets highly relies on the samples included in the test set. [5]

We are performing our experiments in both ways to find out if using all the data is able to improve the overall performance of some methods. Additionally, we are trying to comprehend if the eventually improved performance justifies the increase in training time because of the *k* iterations which have to be performed.

METHODOLOGY

In general, the analysis process consists of four steps to get comparable results. At first a suitable dataset has to be chosen, which is able to answer our questions. This step also includes exploratory data analysis to verify correctness and cleanness of the containing samples.

In the second step we preprocess the data to prepare them for being used in different analysis and machine learning methods. Step three comprises of the training of the different models which are used for classification and should be compared in the end. In the final step the models are used for predicting creditworthiness for

previously unseen data. Different metrics are calculated to evaluate the performances and compare them.

Step 1: Selection of Dataset

For conducting our analysis, we perform experiments on German Credit Data obtained from Pennsylvania State University's Data Mining course [6].

The data consists of 20 variables for 1000 loan applicants. Our analysis is supervised form of learning since we have a target/ dependent variable to be predicted- the creditworthiness encoded 0 for customers who are likely to default and 1 for those who are likely to pay.

The predicting variables and their categories are as follows:

- Account Balance: No account (1), No balance (2), Some Balance (3)
- Payment Status: Some problems, paid up, no problems (in this bank)
- Savings/Stock Value: None, below 100 DM, 100 to 1000 DM, above 1000 DM
- Employment Length: Below 1 year (including unemployed), [1, 4), [4, 7), above 7 years
- Sex/Marital Status: Male divorced/single, male married/widowed, female
- No of Credits at this bank: 1, more than 1
- Guarantor: None, Yes
- Concurrent Credits: Other banks or department stores, none
- Purpose of Credit: New car, used car, Home related, other
- Foreign Worker and Telephone variables are dropped from the study.

The data we obtained was pre-processed to combine levels that had too few observations. For example, if only 0.9% of customers had purpose of credit as 'Vacation', it was combined with the category 'Others'.

Step 2: Cleaning & Formatting

Data consists of both categorical and continuous variables. For categorical predictors with more than two categories, we need to define dummy variables [7]. Keeping Sex/Marital status as 1 = Male single, 2 = Male married, 3 = Female would imply an ordered relationship between them (Female category is somehow "twice" the Male married category). To avoid this, we create another dummy variable.

Male single is coded as Marital = 0 & Female = 0. Male married is coded as Marital = 1 & Female = 0.
Female is coded as Marital =0 & Female =1

We need k-1 dummy variables for a categorical feature with k levels. In the above example, two dummies capture all of the information.

Step 3: Classification Model Training

Selection of one or more algorithm specific parameters that are adjusted to the given data and intended application optimizes performance.

Naïve Bayes: NB estimates conditional probabilities by "naively" assuming that for a given class the inputs are independent of each other. It is used as a base level

classifier for comparison with other algorithms.

Linear Regression:

Multiple LR fits a line through a multi-dimensional cloud of data points such that the line has smallest possible value for the sum of squares of residuals.

Assumption: Feature or predictor variables are independent random variables. This assumption might not be good because whether a person is employed or married does depend on his age. More detailed work would also study collinearity between X variables.

In our experiment we calculate a creditworthiness score. If the score >= 0.5, we classify as creditworthy. If score < 0.5, the person is not creditworthy.

Decision Tree:

In DT algorithm, population is split into sets based on most significant attributes or independent variables to make distinct groups. We control minimum sample split size to be 9 to avoid overfitting. We tried both Gini impurity and entropy (information gain) criteria and the latter gave less error.

Advantages: DTs require less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values very much. It can handle both continuous and categorical features.

Disadvantages: While working with continuous numerical variables, decision tree loses information when it categorizes them [8]. Over fitting is one of the biggest drawbacks for DT models. This problem gets solved by use of random forests discussed next.

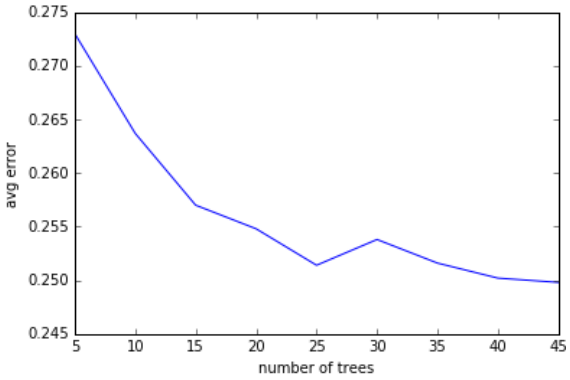

Fig. 1: Average error decreases as number of trees increase. After a threshold of about 25 trees there is no significant error reduction at the expense of computation.

Random Forest: A problem with decision trees is a high variance when growing the tree very deep. Random forest tries to overcome this issues with a similar bias but a dramatically decreased variance. [9] Hence, using this method makes the model harder to interpret.

Each DT of RF is learned on a random bootstrapped sample. Bagging generates training data for each tree by sampling with replacement a number of samples equal to the number of samples in the source dataset. Number of trees, minimum sample split size and tree depth were varied to study error reduction-computation time tradeoff.

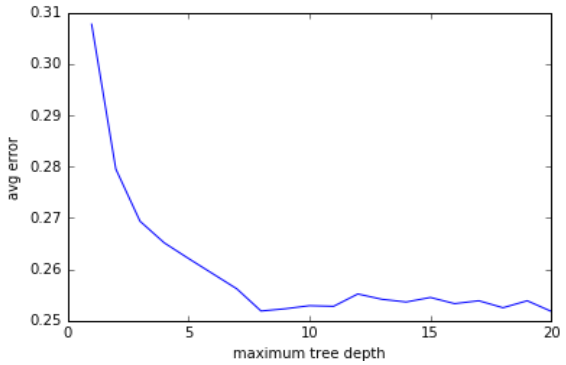

Fig. 2: Error decreases as we increase the tree depth or number of features used until a threshold value, beyond which there is little accuracy gain for high computational cost. Deeper trees are also susceptible to overfitting. Hence, the optimal value is used in final calculation.

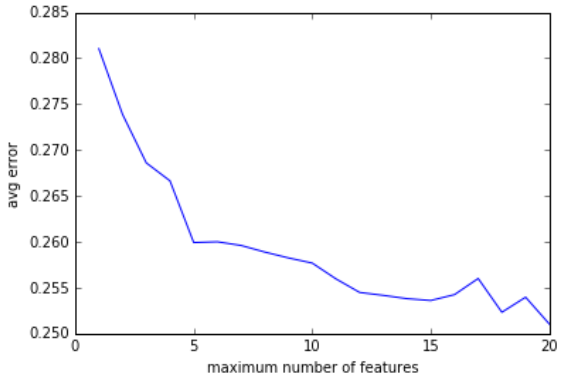

Fig. 3: Error decreases as we increase the number of features used until a threshold value. For the final calculation we use the optimal value

Extremely Randomized Trees:

In addition to Random forest like bagging, in ERT the splitting of learning tree is also randomized instead of making a split on the basis of Gini index or entropy. Its strength lies in computational efficiency but it might have more bias than RF [10].

Support Vector Machines:

Support vectors are the co-ordinates of data points in a space whose dimension is equal to number of features used in classification. SVM algorithm divides the groups by finding a hyperplane that maximizes distance on each side from plane to support vector.

SVM only uses points that are nearest (a subset of training sample) to hyperplane so is memory efficient for big datasets- say if we had credit data for the whole country. In our study, the algorithm used around 180 out of 900 points in each iteration of the 10-fold CV.

The default kernel in python is 'rbf' (Radial Basis Function) but we use 'linear' kernel instead for two reasons: 1. Error in 'rbf' is 0.312 as opposed to 0.260 for linear 2. The weights or co-efficient of linear hyperplane are more interpretable since they represent importance of features (Table 1). Feature importance in SVM and Linear Regression has a very similar pattern.

Step 4: Evaluating Model Performance:

As already mentioned before we use a separate test set for assessing the model performance. For our experiments we use a 75%-25% split of training and testing samples on the one hand. On the other hand, a k-fold cross validation is performed, where we use 10 as the value for *k*. With using 10 bags, we increase the number of training samples from 750 to 900 per iteration. The drawback of the decreased test set size should be overcome

with averaging the results over 10 iterations.

Mean Absolute Error (MAE) was used to compare prediction with creditworthiness values in test. Root Squared Mean Error (RSME) would also give the same result since the predictions are 0-1 binary. MAE of Naïve Bayes is algorithm is 0.316 and is seen as a base case. Marked improvement from this number represents how good other algorithms are.

In binary classification accuracy might not always be the best choice for assessing the performance of a classifier [11]. Therefore, we incorporate precision and recall in our evaluation framework. To get a metric which is better comparable, the F1-score is used. It combines both precision and recall to a single value, which makes it easier to directly compare two different results than with two values.

Penn State course documents- where data was acquired from, had already conducted Logistic Regression and Discriminant analysis so instead of repeating these we decided to use other methods. Their linear discriminant analysis has an accuracy of 60.8%, quadratic DA of 58.2% and logistic regression an accuracy of 64%. Five of our six algorithms give significantly better prediction accuracy (>= 70%) than these.

Feature	SVM	LinReg
AccNoBalance	-0.049	0.211
AccwBalance	0.886	0.710
CreditAmt	0.000	0.000
Duration	-0.003	-0.009
SavingsStocks	0.309	0.122
LongEmployed	0.056	0.092
Instalment%	-0.353	-0.164
Guarantors	0.226	0.112
Paid up	0.503	0.720
NoPayDelay	1.000	1.000
ForUsedCar	-0.409	-0.439
ForHomeExp	-0.219	-0.516
OtherPurpose	-0.570	-0.732
MaleMarried	0.397	0.310
Female	0.293	0.222
AssetCar	-0.172	-0.223
AsetInsurance	0.003	-0.123
AsetRealEstate	-0.253	-0.599
AptRented	-0.220	0.050
AptOwned	-0.230	0.271

Table 1: Normalized coefficients of features shaded by importance. Numbers close to 1 in red represent direct relation while negative numbers- blue represent inverse relation between the feature and the predicted outcome- creditworthiness. White represents almost no correlation.

RESULTS

	NB	LR	DT	RF	ERF	SVM L	SVM rbf
Error	0,276	0,256	0,296	0,244	0,264	0,260	0,312
F1 Score	0,817	0,832	0,794	0,836	0,829	0,829	0,814
% Acc	72,40	74,40	70,40	75,60	73,60	74,00	68,80
Time [s]	0,003	0,011	0,007	0,093	0,074	215	0,120

Table 2: Comparison of percentage accuracy, F1 score and computation time of different algorithms with a 75-25 split for training and testing data

	NB	LR	DT	RF	ERF	SVM L	SVM rbf
Error	0,279	0,247	0,319	0,235	0,247	0,244	0,309
F1 Score	0,806	0,835	0,769	0,841	0,837	0,835	0,816
% Acc	72,10	75,30	68,10	76,50	75,30	75,60	69,10
Time [s]	0,380	0,024	0,070	1,090	0,560	1999	0,129

Table 3: Comparison of percentage accuracy, F1 score and computation time of different algorithms using 10-fold cross validation with random data bags

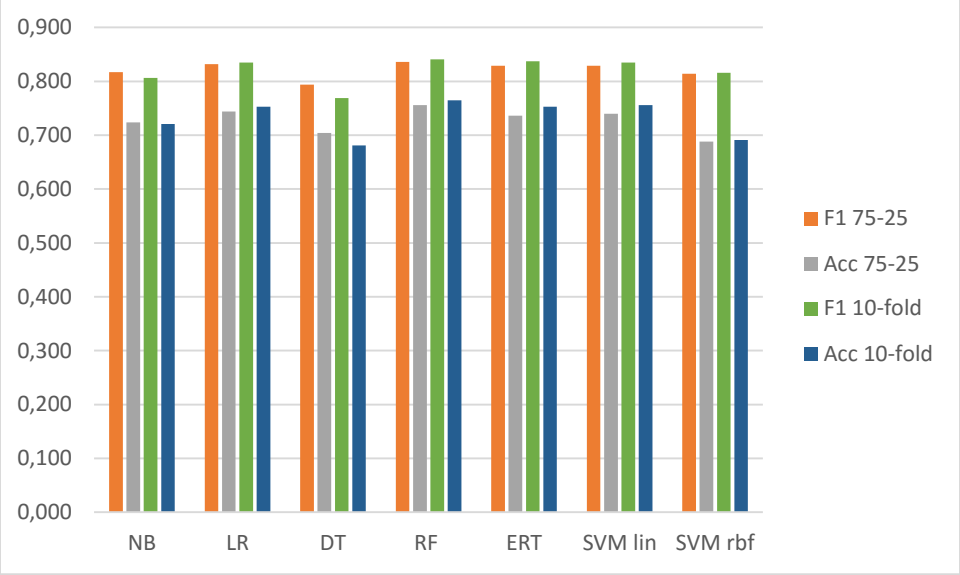

Fig 4: Comparison of accuracy and F1 score for both 75-25 split of train and test set and 10-fold cross validation

The methods are compared on basis of four criteria: stability, ease of use, processing time and prediction accuracy. Additionally, the performance and processing time are compared between fixed split test set and 10-fold cross validation.

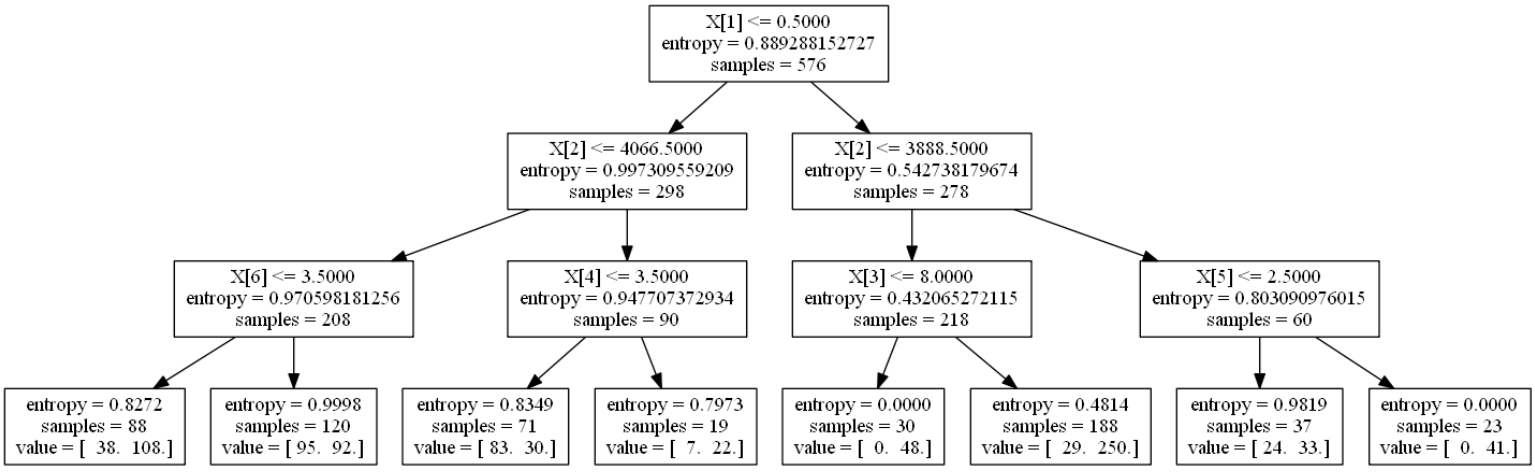

Fig. 5: Decision tree truncated at depth = 3 as an example. Feature 'Account with balance' is the root node (also high importance in LR, SVM). Sum of all values at the bottom is size of sample = 900.

1. Stability: NB, LR and SVM were very stable: producing the same error to the third decimal at each run as well as taking a fairly constant time each instance that the algorithm was run. DT error varied from 0.29 to 0.32 implying the predictions vary by a discernable amount even for a constant given dataset. DT, RF and ERT are not very stable. We iterated these methods 50 times each to obtain the average error and time quoted in Table 2.

2. Ease of use and interpretation: DT gives a very visual result that can be Information gained at each decision node is clear, so are the number of samples being segregated in each category, as shown in Fig. 5. LR and SVM are good at conveying relative feature importance. Multidimensional hyperplane in SVM can be interpreted only in parts when projected to 2D. RF is only somewhat interpretable. MLAs like Neural Network (not used in this paper) though more accurate for large datasets, are like a black box- difficult to trace back or explain by verbal logic.

3. Processing Time: NB, LR and DT take time similar, low time. RF and ERT take an order of magnitude time more, understandably since they are comprised of tens of DTs. SVM linear kernel gives a better result than radial basis function but takes significantly more time than all the algorithms. Time-accuracy tradeoff can be optimized by checking whether a deeper analysis gives proportionate reduction in error- as explored in Fig. 1, 2 and 3. The risk of overfitting should always be considered even if computation time is not a constraint.

4. Accuracy of prediction: It was surprising to note that a simple method like LR gave comparable results to RF and SVM- while being more stable and less computationally expensive than both. Type of dataset- continuous vs categorical variables, number of features, amount of data, spread of observations- outliers, collinearity between features, irrelevant features- all affect which algorithm will better perform in a study.

5. Fixed test set vs. 10-fold cross validation: It can be seen in table 3 that 10-fold cross validation does not change the classifier performances significantly. Most of the accuracy and F1 scores are on the same level in both experiments. Decision tree is the only classifier where the assessed performance drops when using 10-fold cross validation. A more visual comparison is given in figure 4, where the results are given as a bar chart.

It can also be seen in the table that the training time is multiplied approximately with factor 10 for 10-fold CV. For most of the methods this is not a problem because of the small data set. But for SVM with the linear kernel the training time increases from 215 to 1999 seconds. It is obvious that this approach is computationally not reasonable for this small improvement in performance. It can be said that the fixed split train and test sets are the better option for this particular application.

REFLECTION

The experiments in this paper just focus on binary classification – whether a person is creditworthy or not. For further work, can devise a step-like scoring in risk-based pricing. This approach can have two major advantages for businesses.

First of all, when using a finer grain classification, it is possible to find out if a person is close to the edge between creditworthy and non-creditworthy. In those cases, the company can investigate the financial situation of the person more closely, to get a better estimate.

Terms of a loan, including the interest rate offered to borrowers in reality are based on the probability of repayment. Taking higher risk is compensated by higher returns. If using more classes, the terms can be adjusted automatically based on the creditworthiness class.

Hence, for performing multiclass classification a much bigger dataset is needed to get accurate results. A second issue is the need of a pre-labeled data set with all the labels.

CONCLUSION

In our results it can be clearly seen that for this particular application simple models like LR and NB provide a similar performance compared to the more sophisticated ones like SVM. With this metrics in mind it can be said that easy models are the better choice for predicting creditworthiness with the data set we selected for the experiments. They are less computationally expensive and provide an easier-to-understand model. The second point is especially import for presenting the results to businesses.

Given the monumental social impact of lending decisions, MLA results should be interpreted in the light and context of many other factors. If a borrower is likely to pay back the loan, then not lending is a loss to the bank. However, subprime mortgages have in the past spun the world economy into an abyss that is difficult to

rise back from. It is futuristic, almost fictional to ideate MLAs capable of accounting for layered debts, bonds and insurances.

REFERENCES

[1] SCHUFA (2016). *Transparent scoring method*. Retrieved from: https://www.schufa.de/en/about-us/data-scoring/scoring/transparent-scoring-methods/transparent_scoring_methods.jsp

[2] Cai, E. (2016). Machine Learning Lesson of the Day – The "No Free Lunch" Theorem. Retrieved from: http://www.statsblogs.com/2014/01/25/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/

[3] Markham, K. (2015). Comparing supervised learning algorithms. Retrieved from: http://www.dataschool.io/comparing-supervised-learning-algorithms/

[4] Lantz, B. (2013). Black Box Methods – Neural Networks and Support Vector Machines. *Machine Learning with R*. ISBN 978-1782162148

[5] Refaeilzadeh, P., Tang, L., Liu H. (2009). Cross-validation. *Encyclopedia of Database Systems*. DOI 10.1007/978-0-387-39940-9_565

[6] Pennsylvania State University Dept. of Statistics (2016). Retrieved from: https://onlinecourses.science.psu.edu/stat857/node/215

[7] Markham, K., Burroughs, B. (2015). *Linear Regression*. Retrieved from: https://github.com/justmarkham/DAT4/blob/master/notebooks/08_linear_regression.ipynb

[8] Ray, S. (2015). Essentials of Machine Learning Algorithms. Retrieved from: http://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/

[9] Hastie, T., Tibshirani, R., Friedman, J. (2009). Random Forests. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. DOI 10.1007/978-0-387-84858-7_15

[10] Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely randomized trees. *Mach Learn ()*. DOI 10.1007/s10994-006-6226-1

[11] Tryolabs (2013). *Why accuracy alone is a bad measure for classification tasks, and what we can do about it.* Retrieved from: http://blog.tryolabs.com/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/