Machine Learning Project

Comparison of Statistical Machine Learning Techniques for Laon Default Prediction

submitted by

Sahra Ghalebikesabi

December 2018

Abstract

This paper assesses the performance of various machine learning methods for the classification of loan defaults using a data set provided by the online lending platform Lending Club. While the performances of most approaches do not differ, the gradient boosted machines can be shown to significantly outperform other methods.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AUC	Area under the curve.
CD	Coordinate descent.
CV	Cross-validation.
GAM	Generalised additive model.
GBM	Gradient boosted machine.
kNN	K-nearest neighbour.
LC	Lending Club.
LDA	Linear discriminant analysis.
MER	Misclassification error rate.
MSE	Mean-squared error.
P2P	Peer-to-peer.
Q-Q	Quantile-quantile.
QDA	Quadratic discriminant analysis.
RF	Random forest.
RSS	Root-squared sum.
sa	Searching accuracy.
SVC	Support vector classifier.
SVM	Support vector machine.
w.o.	Without.

1 Risk Assessment in Peer-to-Peer Lending

1.1 Introduction

In recent years, peer-to-peer (P2P) lending has emerged as a new lending model where investors lend a borrower money directly through online trading platforms without any intermediary (Kumar et al., 2016). Sustainability of such platforms requires reliable risk attribution (Malekipirbazari and Aksakalli, 2015). In order to detect in advance whether a loan will be fully paid or charged-off, we consider data available from the largest social lending platform based in the United States `www.lendingclub.com` (Lending Club, 2018a). This data set has been discussed thoroughly in recent years (Serrano-Cinca and Gutiérrez-Nieto, 2016; Emekter et al., 2015; Malekipirbazari and Aksakalli, 2015). While the latter propose the usage of random forests, Emekter et al. (2015) suggest logistic regressions. Regarding the lack of literature assessing the performance of various models, we will identify the one machine-learning method rendering the lowest misclassification error rate (MER). In order to achieve this aim, we first describe our data, then present some methods for classification and subsequently identify the best performing algorithm. The last subsection concludes.

1.2 Data Description

For our purpose we collect loan data from January 2007 to September 2011 from Lending Club (2018b) where each instance is a loan with information on its characteristics (e.g. loan amount) and the corresponding borrower (e.g. annual income).
Based on domain knowledge we clean the data and extract relevant parameters. Our final data set consists of 17 variables described in Table 3 on page 7. We further scale numeric data and convert categorical into binary variables as further discussed in subsection A.1 on page 6 in the appendix.
Since the raw data set comprises 42,538 loans of which only 6,428 are charged-off, we randomly sample 6,428 instances from the fully-paid loans such that our final data set consists of 12,856 records. Although this approach bears information loss, this drawback affects all models equally such that our research question is not touched.

1.3 Comparison of Diverse Machine-Learning Algorithms

In order to cover the most important machine learning methods, we restrict our analysis to those algorithms applied in common credit default detection literature (Carter and Catlett, 1987; Huang et al., 2007; Yeh and Lien, 2009) and those algorithms presented in Wu et al. (2008), who identify the most influential data mining algorithms in the research community. We present the mean MER of the best performing configurations obtained in a 10-fold cross-validation (CV) in Table 1 on the following page. Please refer to subsection A.2 on page 8 for an elaboration on our evaluation methods and subsection A.3 on page 9 for more detailed results, both in the appendix. For each configuration we choose a discrimination threshold of 0.5.
The logistic (as well as probabilistic) regression only performs ordinarily. Including different settings of polynomial terms up to degree 3 or settings of splines with a

Approach	MER	Approach	MER
GBM, $\lambda = 0.008$	0.2872	RF, $n.trees = 425$	0.3640
GBM, $\lambda = 0.001$	0.2997	Logit (`annual_inc` as spline)	0.3765
Logistic polynomial regression	0.3474	Elastic net, $\alpha = 0.02$	0.3805
Logistic regression	0.3525	Neural network (5 and 3 nodes)	0.3901
Linear discrimininant analysis	0.3538	Logit group lasso	0.4009
Logit AdaBoost, $n = 550$	0.3542	kNN, $k = 9$	0.4090
SVC, $cost = 200$	0.3570	weighted kNN, $k = 10$	0.4157
Logit adaptive lasso	0.3629	Naïve Bayes classifier	0.4241

Table 1: Error rates of best performing configurations in ascending order of respective MER's

degree of freedom of 6 do not improve the performance suggesting a linear relationship. The adaptive lasso according to Zou (2006) and the elastic net with $\alpha := \lambda_1$ and $\lambda_2 := (1-\alpha)/2$, where λ_1 and λ_2 denote the shrinkage parameters of the l_1 and l_2 penalties, reduce the model complexity to only 22 variables when the shrinkage parameters are chosen according to the one standard deviation rule. However since the model includes categorical variables which are binarised for the regression, interpretability requires the use of the group lasso for feature selection, which obviously performs worse (Meier et al., 2008).

To increase the models' accuracy, we apply diverse boosting methods according to Drucker (1997) on the best performing regression model, the logistic regression including all variables. AdaBoost with weight updating coefficient according to Freund et al. (1996) and 550 iterations for which boosting is run renders the smallest MER.

We also apply a linear discriminant analysis (LDA) approach on all variables. Since this method requires Gaussian input, we consider only continuous variables in a next step. Please refer to the Q-Q plots in Figure 6 on page 15 to verify that these variables follow approximately a normal distribution. For $k = 1, ..., 10$ we also train the best LDA model that uses only k variables. However misclassification increases. Quadratic discriminant analysis (QDA) performs even worse.

For the sake of completeness a backpropagation neural network with two hidden layers of 5 and 3 nodes as proposed by Surkan and Singleton (1990) is also trained on the data set. In our setting, however, it exhibits one of the highest MERs.

The Naïve Bayes classifier and the k-nearest neighbour (kNN) method perform worst with an MER higher than 0.4, where the latter approach's performance is optimised for $k = 9$. Using a distance weighted kNN algorithm based on the Epanechnikov kernel and Minkowski distance 1 even impairs the performce of the kNN classifier.

Further we train support vector machines (SVMs) with Gaussian radial kernels by first choosing the cost parameter, before considering the gamma parameter based on 10-fold CV. The support vector classifier (SVC) outperforms all settings for the cost set to 200 suggesting a linear decision boundary.

Within the context of random forests (RFs) splitting is done with respect to the Gini Index. Since the MER does not decrease up to the third decimal place, when the size

of the random forest is increased from 425 to 450, the RF with 425 trees should be considered as dominating configuration considering computational arguments. The maximal interaction depth is found to be optimised by 4 in a 10-fold CV.

1.4 Discussion of the Final Approach

Based on the 5×2 CV paired t-test proposed by Dietterich (1998), which is presented in subsection A.2 on page 8 in the appendix, we find that the gradient boosted machine (GBM) with $\lambda = 0.008, n = 425$ and a maximal interaction depth of 4 outperforms all other models at a 5% significance level with an MER of only 0.2872.

Considering Table 9 on page 13 we notice that the type I error rates of the GBM tend to decrease as λ decreases. As the configurations of the GBM do not differ significantly with respect to MER, we suggest a cost sensitive analysis of the different approaches. Since accepting a loan that will be charged-off carries much greater risk than rejecting a loan that will be fully-paid, we propose as a final approach a GBM with $\lambda = 0.01$, maximal interaction depth of 4 and 425 trees since this model minimises the costly type I error while only increasing the MER by 0.0125 compared to the same model with $\lambda = 0.08$.

More thorough parameter fine tuning could also increase the efficiency of other methods as SVM or neural networks. Drawbacks of these methods, however, are the required computational time and their black-box nature. In comparison GBMs are faster and allow for interpretation through variable importance plots according to Friedman (2001). Please refer to Figure 4 on page 14 for that plot of our final approach.

After having validated our final approach, we use the GBM as specified above, train it on the old data set and test it now on new data from LC covering the months from October 2011 to September 2013. This data is cleaned as before. Since the new data is highly unbalanced, we assess the AUC (area under the curve) instead of the MER. With an AUC of 0.68 the model performs nearly as good as the RFs presented by Malekipirbazari and Aksakalli (2015). Since Schebesch and Stecking (2005) assume that a type I error is 5 times as expensive as a type II error, we suggest the use of GBMs with higher discrimination thresholds. Using a threshold of 0.64 was found to be optimal in a 10-fold CV on the data from 2007 to 2011.

1.5 Conclusion

The findings show that loan default data is only weakly non-linear. Apart from that it was found that many classification techniques such as logistic regression or SVCs yield performances which are quite competitive with each other. Using GBMs, however, we achieve MERs nearly as low as those of Malekipirbazari and Aksakalli (2015). We tested only a limited number of SVMs and neural networks in our study. It is reasonable that careful parameter fine tuning could improve their performance.

Further research could ensemble the best presented classifiers to a unified approach where multiple classification algorithms are combined and classifications are made based on voting schemes (Twala, 2010). Another promising research venue would be to study genetic algorithm based approaches to optimise the parameters and feature subset simultaneously according to Huang and Wang (2006).

References

Carter, C. and Catlett, J. (1987). Assessing credit card applications using machine learning. *IEEE expert*, 2(3):71–79.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Drucker, H. (1997). Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115.

Emekter, R., Tu, Y., Jirasakuldech, B., and Lu, M. (2015). Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending. *Applied Economics*, 47(1):54–70.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Citeseer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Huang, C.-L., Chen, M.-C., and Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, 33(4):847–856.

Huang, C.-L. and Wang, C.-J. (2006). A GA-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with applications*, 31(2):231–240.

Kumar, V., Natarajan, S., Keerthana, S., Chinmayi, K., and Lakshmi, N. (2016). Credit risk analysis in peer-to-peer lending system. In *Knowledge Engineering and Applications (ICKEA), IEEE International Conference on*, pages 193–196. IEEE.

Lending Club (2018a). How does an online credit marketplace work? `https://www.lendingclub.com/public/how-peer-lending-works.action`. Accessed: 2018-11-30.

Lending Club (2018b). Lending Club Statistics. `https://www.lendingclub.com/info/download-data.action`. Accessed: 2018-11-30.

Malekipirbazari, M. and Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications*, 42(10):4621–4631.

Meier, L., Van De Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71.

Schebesch, K. B. and Stecking, R. (2005). Support vector machines for credit scoring: Extension to non standard cases. In *Innovations in classification, data science, and information systems*, pages 498–505. Springer.

Serrano-Cinca, C. and Gutiérrez-Nieto, B. (2016). The use of profit scoring as an alternative to credit scoring systems in peer-to-peer lending. *Decision Support Systems*, 89:113–122.

Surkan, A. J. and Singleton, J. C. (1990). Neural networks for bond rating improved by multiple hidden layers. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 157–162. IEEE.

Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications*, 37(4):3326–3336.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37.

Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

A Appendix

A.1 Data Cleaning

The unmodified data set has 64 variables from which we delete those with an NA ratio greater than 0.5, that is more than 21,269 NA values. Please refer to Table 2 for an overview of these variables and the remaining variables with NA values.

Number of NAs	Variables
42,378	settlement_amount, settlement_percentage, settlement_term
38,887	mths_since_last_record
26,929	mths_since_last_delinq
1,368	pub_rec_bankruptcies
148	collections_12_mths_ex_med, chargeoff_within_12_mths
108	tax_liens
32	delinq_2yrs, inq_last_6mths, open_acc, pub_rec, total_acc, acc_now_delinq, delinq_amnt
7	annual_inc
3	loan_amnt, funded_amnt, funded_amnt_inv, installment, dti, revol_bal, out_prncp, out_prncp_inv, total_pymnt, total_pymnt_inv, total_rec_prncp, total_rec_int, total_rec_late_fee, recoveries, collection_recovery_fee, last_pymnt_amnt, policy_code
1	title

Table 2: Number of NAs of variables with a positive number of NAs

Next we omit further 43 variables since

1. they are based on information that is not available before the issue date (e.g. funded_amnt),
2. they impose multicollinearity (e.g. grade),
3. they have too imbalanced classes (e.g. tax_liens),
4. obviously unrelated to the loan status (e.g. member_id) or
5. they are not suitable for the basic machine learning methods we use (e.g. desc).

The remaining 16 variables are adjusted for our purpose (e.g. binary variables are set equal to either 0 or 1). Please refer to Table 3 on the facing page for a description of the used variables according to Lending Club (2018b).

Variable	Description
addr_state	The state provided by the borrower in the loan application.
annual_inc	The self-reported annual income provided by the borrower during registration.
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Possible values are RENT, OWN and MORTGAGE.
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries).
int_rate	The interest rate on the loan.
loan_amnt	The listed amount of the loan applied for by the borrower.
loan_status	Current status of the loan, either fully paid or charged off.
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records.
purpose	A category provided by the borrower for the loan request. Possible values are credit_card, debt_consolidation, car, education, home_improvement, house, major_purchase, medical, moving, renewable_energy, small_business, vacation, wedding and other.
revol_util	Revolving line utilisation rate, or the amount of credit the borrower is using relative to all available revolving credit.
sub_grade	LC assigned loan sub grade from A1 to G5. In our data set the sub grades are enumerated from 1 to 35 assuming equal distances between subsequent sub grades.
term	The number of payments on the loan. Our values are either 36 months or 60 months.
total_acc	The total number of credit lines currently in the borrower's credit file.
verification_status	Indicates if income or income source was verified or not verified by LC.

Table 3: Description of variables employed in the analysis

We turn the categorical variables into binary variables in order to be able to run regressions. Further we rescale the data only allowing for values in $[0,1]$. In a next step we omit those 32 loans with incomplete information such that our final data set has $n = 12330$ entries and 78 variables which is obviously smaller than $\sqrt{12330} \approx 111$. Although one could argue that omitting the variable `addr_state` increases the computational efficiency of the models, MERs are significantly higher without.

A.2 Evaluation Methods

In total 6,428 of these loans were charged-off such that we have a ratio of 0.1781 charged-off loans to fully-paid loans. Since the data is not balanced, we could either restrict ourselves to the analysis of AUC or randomly undersample our data set, that is include all charged-off loans and bootstrap 6,428 fully-paid loans. Because of computational advantages, we decided for the second approach. Although this method entails information loss, it biases our results to the benefit to identifying charged-off loans.
In the first instance we are interested in the MER. However, since the type I error bears much higher costs than the type II error, results for these rates are also represented.
Since leave-one out cross-validation is computationally not feasible for our data sets, we obtained our train and testing data set in two different ways:

1. We tried common splitting ratios employed in machine learning tasks: 80/20, 70/30 and 60/40.
2. We also employed common cross validation approaches, that is to say the 10-fold and 5-fold cross-validation.

To assess the superiority of our final approach we conduct the 5×2 CV paired t tests introduced by Dietterich (1998).

1. In five replications, the available data are randomly partitioned into two equal-sized sets, S_1 and S_2.
2. Each learning algorithm (A or B) is trained on each set and tested on the other set rendering four error estimates for each replication $i \in \{1,...,5\}$: $p_{A,i}^{(1)}$ and $p_{B,i}^{(1)}$ (trained on S_1 and tested on S_2) and $p_{A,i}^{(2)}$ and $p_{B,i}^{(2)}$ (trained on S_2 and tested on S_1).
3. The resulting test statistic
$$T = \frac{p_{A,1}^{(1)} - p_{B,1}^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}}$$
with $s_i^2 = 2((p_{A,i}^{(1)} - p_{B,i}^{(1)})/2 - (p_{A,i}^{(2)} - p_{B,i}^{(2)})/2)^2$ is t distributed with 5 degrees of freedom

A.3 Some Evaluation Results

In the subsequence, we present the results of most of our models. Please refer to the main part for the best performing configurations.

A.3.1 Regression

Approach	MER	Type I error	Type II error
Logistic regression	0.3525	0.3888	0.316
Probabilistic regression	0.3519	0.379	0.3248
logit_poly	0.3474	0.3496	0.3452
probit_poly	0.3477	0.35	0.3454
Adaptive lasso	0.3568	0.5132	0.2006
Group lasso	0.3609	0.3978	0.3562
Elastic net, alpha=0	0.3991	0.6676	0.1306
Elastic net, alpha=0.1	0.3805	0.5948	0.1662
Elastic net, alpha=0.2	0.383	0.6016	0.1646
Elastic net, alpha=0.3	0.3838	0.6084	0.159
Elastic net, alpha=0.4	0.384	0.6092	0.1588
Elastic net, alpha=0.5	0.3845	0.6176	0.1514
Elastic net, alpha=0.6	0.3844	0.6146	0.1542
Elastic net, alpha=0.7	0.3841	0.6164	0.1518
Elastic net, alpha=0.8	0.3841	0.616	0.1522
Elastic net, alpha=0.9	0.3875	0.6306	0.1444
Elastic net, alpha=1	0.3835	0.5662	0.2008
Elastic net poly, alpha=0	0.4063	0.3818	0.4308
Elastic net poly, alpha=0.2	0.3749	0.5062	0.2436
Elastic net poly, alpha=0.4	0.4108	0.4098	0.4118
Elastic net poly, alpha=0.6	0.3822	0.5602	0.2042
Elastic net poly, alpha=0.8	0.3824	0.5582	0.2066
Elastic net poly, alpha=1	1	2	2

Table 4: Error rates for regression models

Approach	MER	Type I error	Type II error
GAM w.o. s	0.3809	0.5912	0.1706
GAM, s=loan_amnt	0.3785	0.586	0.171
GAM, s=int_rate	0.3798	0.5874	0.172
GAM, s=sub_grade	0.3781	0.5794	0.1768
GAM, s=emp_length	0.3789	0.5866	0.1712
GAM, s=annual_inc	0.3765	0.5788	0.174
GAM, s=delinq_2yrs	0.3798	0.5884	0.1714
GAM, s=inq_last_6mths	0.3807	0.589	0.1724
GAM, s=open_acc	0.3811	0.5908	0.1714
GAM, s=pub_rec	0.3806	0.5898	0.1714
GAM, s=revol_util	0.3799	0.5906	0.1692
GAM, s=total_acc	0.3778	0.5866	0.169
GAM s except for open_acc	0.3878	0.6132	0.1624

Table 5: Error rates for GAMs

Approach	MER	Type I error	Type II error
SAMME	0.3628	0.3288	0.3926
Ada_boost Breiman	0.3555	0.326	0.3838
Ada_boost Freund, n=100	0.3575	0.3316	0.3838
Ada_boost Freund, n=50	0.3597	0.3418	0.3776
Ada_boost Freund, n=150	0.3565	0.3296	0.3836
Ada_boost Freund, n=200	0.358	0.337	0.3792
Ada_boost Freund, n=250	0.3572	0.3292	0.3852
Ada_boost Freund, n=300	0.3556	0.3346	0.3766
Ada_boost Freund, n=350	0.3559	0.3264	0.3854
Ada_boost Freund, n=400	0.3562	0.329	0.3834
Ada_boost Freund, n=450	0.3567	0.3332	0.3802
Ada_boost Freund, n=500	0.3574	0.332	0.3828
Ada_boost Freund, n=550	0.3542	0.3282	0.3802

Table 6: Error rates of adaptive boosting of for binomial regression

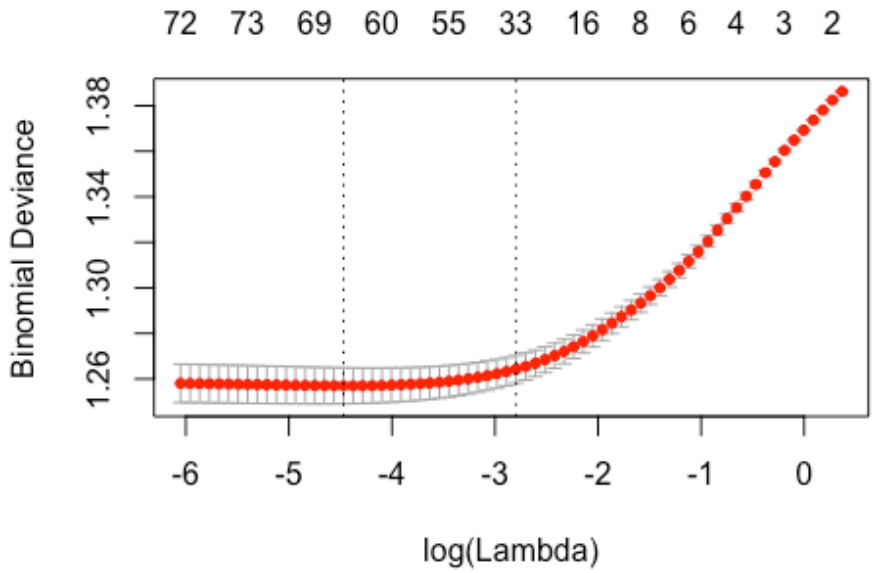

Figure 1: Binomial deviance in dependence of $log(\lambda)$ in the elastic net approach

A.3.2 Support Vector Machines

Approach	MER	Type I error	Type II error
SVC, cost=0.001	0.4228	0.6628	0.1828
SVC, cost=0.01	0.3854	0.5036	0.2672
SVC, cost=0.1	0.3653	0.4216	0.309
SVC, cost=1	0.3599	0.401	0.3188
SVC, cost=5	0.3577	0.3958	0.3196
SVC, cost=10	0.3577	0.394	0.3216
SVC, cost=100	0.3567	0.3882	0.3252
SVC, cost=200	0.357	0.3878	0.3262
SVC, cost=500	0.3569	0.3872	0.3266
SVM, cost=10, g=0.5	0.4051	0.4024	0.4078
SVM, cost=10, g=1	0.4169	0.4136	0.4202
SVM, cost=10, g=2	0.4348	0.4334	0.4362
SVM, cost=10, g=3	0.4463	0.4578	0.4346
SVM, cost=10, g=4	0.4542	0.463	0.4454

Table 7: Error rates for SVC and SVM

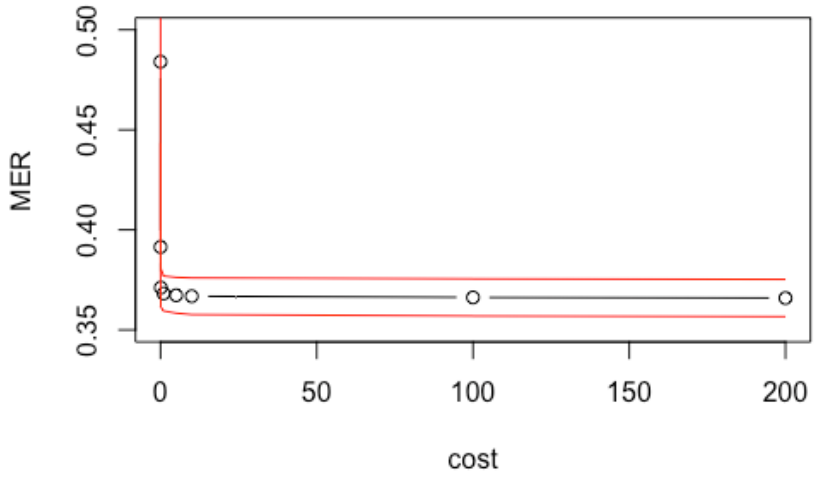

Figure 2: MER in dependence of the cost parameter for SVMs with radial kernel

A.3.3 Random Forests and Gradient Boosted Machines

Approach	MER	Type I error	Type II error
RF, n.trees=50	0.3653	0.3494	0.3812
RF, n.trees=75	0.3657	0.3448	0.3866
RF, n.trees=100	0.3639	0.3418	0.3858
RF, n.trees=125	0.3665	0.349	0.384
RF, n.trees=150	0.3668	0.3494	0.3842
RF, n.trees=175	0.3651	0.3492	0.3808
RF, n.trees=200	0.3658	0.3428	0.3888
RF, n.trees=225	0.3649	0.3436	0.3862
RF, n.trees=250	0.3652	0.3454	0.3848
RF, n.trees=275	0.3642	0.3438	0.3844
RF, n.trees=300	0.3663	0.3478	0.3848
RF, n.trees=325	0.365	0.3454	0.3846
RF, n.trees=350	0.3664	0.3486	0.3842
RF, n.trees=375	0.3659	0.3472	0.3846
RF, n.trees=400	0.3652	0.3478	0.3826
RF, n.trees=425	0.364	0.3474	0.3806
RF, n.trees=450	0.3661	0.3518	0.3804
RF, n.trees=475	0.3653	0.3472	0.3834
RF, n.trees=500	0.3648	0.3452	0.3844

Table 8: Error rates of RF for different number of trees n.trees

Approach	MER	Type I error	Type II error
GBM, λ=0.001	0.2997	0.2479	0.3515
GBM, λ=0.002	0.2952	0.2502	0.3399
GBM, λ=0.003	0.2922	0.2531	0.3312
GBM, λ=0.004	0.2903	0.2560	0.3246
GBM, λ=0.005	0.2894	0.2604	0.3185
GBM, λ=0.006	0.2875	0.26	0.3150
GBM, λ=0.007	0.2875	0.2626	0.3127
GBM, λ=0.008	0.2872	0.2632	0.3112
GBM, λ=0.009	0.2876	0.2658	0.3095
GBM, λ=0.01	0.2875	0.2652	0.3098

Table 9: Error rates of GBM for different lambda and a maximal possible interaction depth of 4

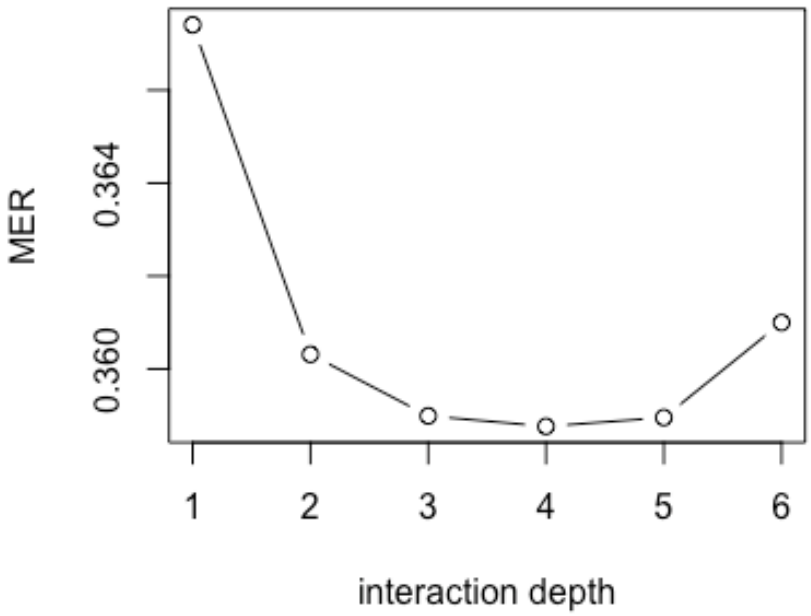

Figure 3: Dependence of MER on the interaction depth

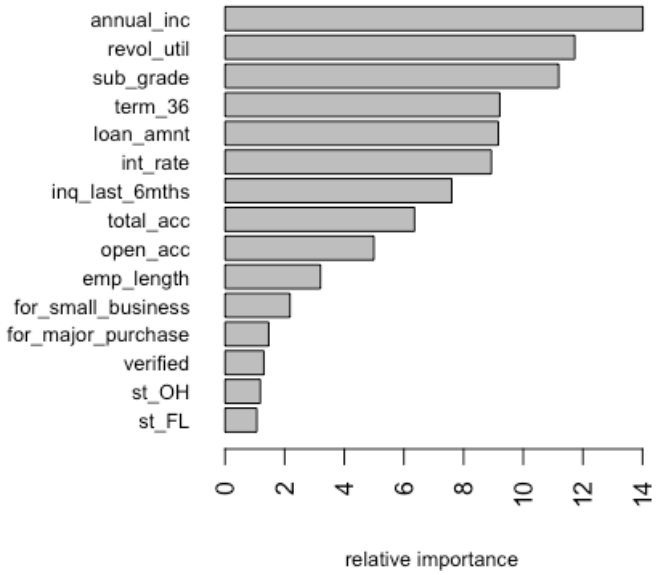

Figure 4: Variable importance plot of GBM with $\lambda = 0.01$, 425 trees and a maximal interaction depth of 4 according to Friedman (2001)

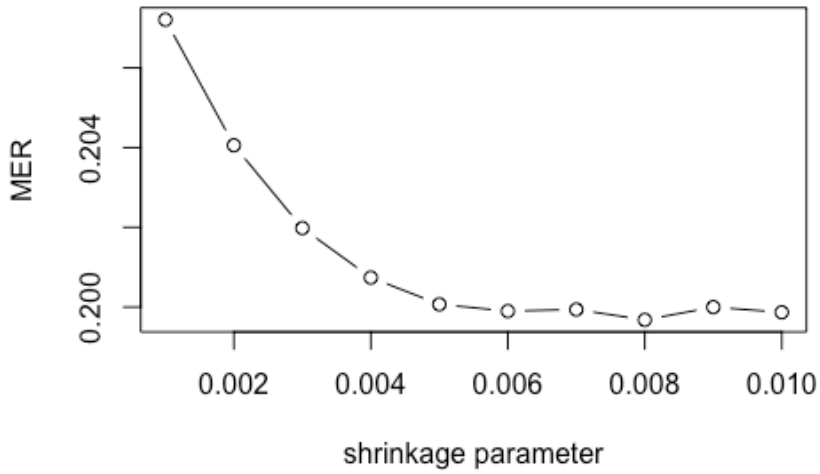

Figure 5: Dependence of MER on the shrinkage parameter

A.3.4 Discriminant analysis

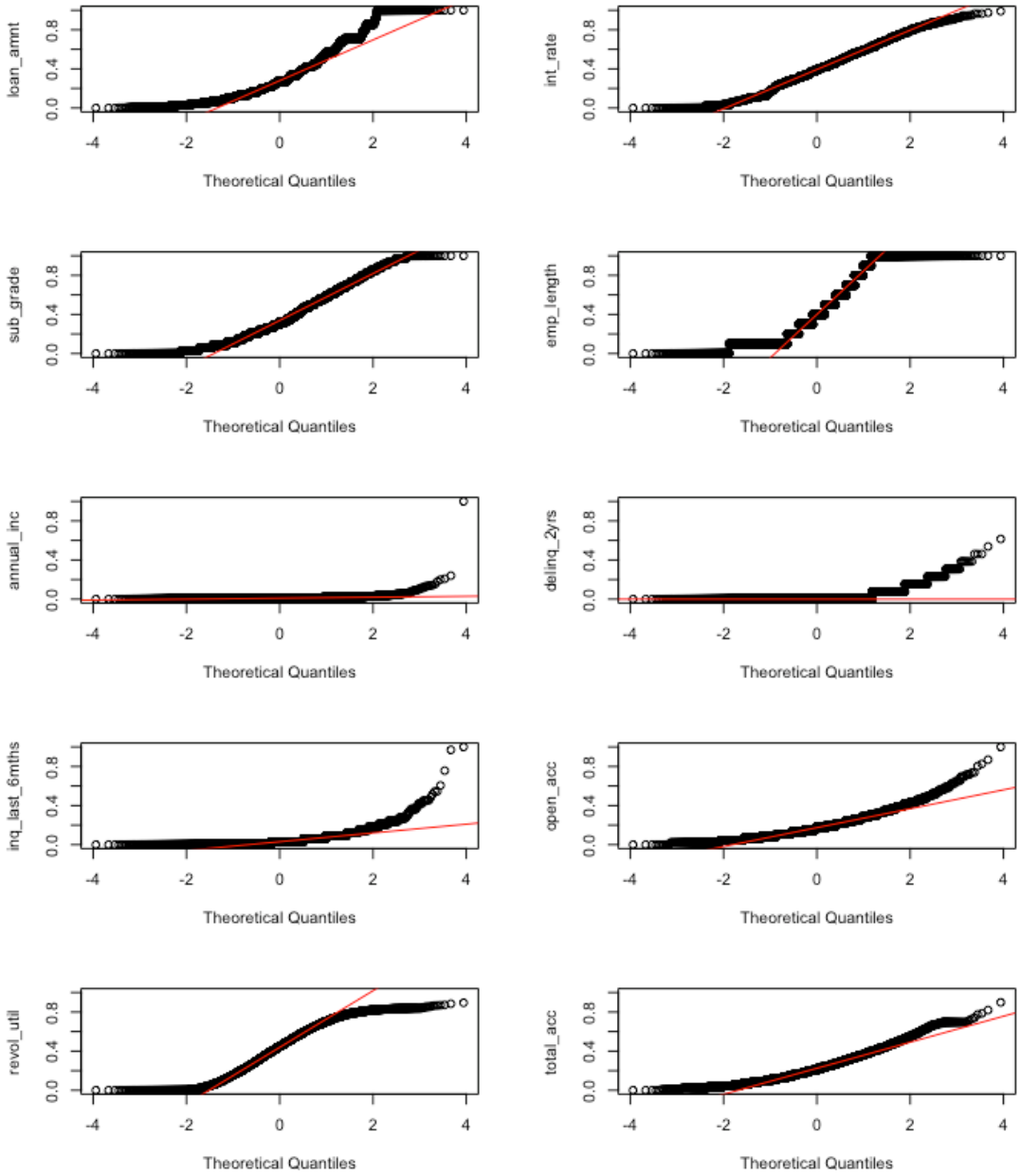

Figure 6: Q-Q plots of the variables used in the discriminant analysis

Approach	MER	Type I error	Type II error
LDA	0.3538	0.375	0.3326
LDA Gaussian	0.3653	0.3814	0.3492
QDA Gaussian	0.3924	0.4784	0.3064
LDA, k=2	0.3852	0.3882	0.3822
LDA, k=3	0.377	0.4392	0.3148
LDA, k=4	0.3785	0.4398	0.3172
LDA, k=5	0.3783	0.4364	0.3202
LDA, k=6	0.3771	0.4352	0.3192
LDA, k=7	0.3801	0.405	0.355
LDA, k=8	0.3805	0.407	0.3542
LDA, k=9	0.3684	0.3872	0.3496
LDA, k=10	0.3695	0.3908	0.3484

Table 10: Error rates of discriminant analysis methods

A.3.5 K- nearest neighbors

Approach	MER	Type I error	Type II error
kNN, k=1	0.4525	0.463	0.442
kNN, k=2	0.4566	0.4636	0.4496
kNN, k=3	0.4292	0.4442	0.4142
kNN, k=4	0.4333	0.4488	0.4178
kNN, k=5	0.425	0.4348	0.4152
kNN, k=6	0.4274	0.44	0.4148
kNN, k=7	0.4176	0.4332	0.402
kNN, k=8	0.417	0.4324	0.4016
kNN, k=9	0.409	0.4274	0.3906
kNN, k=10	0.4148	0.434	0.3956
weighted kNN, k=1	0.4411	0.4566	0.4256
weighted kNN, k=2	0.4411	0.4566	0.4256
weighted kNN, k=3	0.4342	0.4518	0.4166
weighted kNN, k=4	0.4284	0.451	0.4058
weighted kNN, k=5	0.4283	0.4526	0.404
weighted kNN, k=6	0.4246	0.4504	0.399
weighted kNN, k=7	0.4219	0.4474	0.3962
weighted kNN, k=8	0.4181	0.4438	0.3922
weighted kNN, k=9	0.4169	0.4436	0.3902
weighted kNN, k=10	0.4157	0.4434	0.3878

Table 11: MERs of kNN methods