unine
Université de Neuchâtel

Credit applicant classification

Seminar of applied statistics

Aritz Lizoain

June 2022

Abstract

The goal of this study is to determine, in an automated manner, whether new applicants present good or bad credit risk. Several machine learning models are trained to classify applicants based on their credit rating. A random forest is found to be the best performing model, which ascertains the checking account status, credit duration, credit history, average balance in savings account, and credit amount, to be the most decisive information for classification.

Table of Contents

1 Introduction

Approving or denying a credit application requires the lender to exhaustively check the applicant's background and information. Additionally, there exists no simple rule for making a final decision; a combination of numerous variables need to be analyzed together. For this reason, a machine learning model is sought. The goal of the study is to train a model that can accurately classify credit applicants based on their credit rating: good credit risk, or bad credit risk. Moreover, the most important variables will be highlighted.

Following the CRoss Industry Standard Process for Data Mining (CRISP-DM) model (see Figure 1), this work mainly comprehends the phases of data understanding, data preparation, modeling, and evaluation.

Figure 1. CRoss Industry Standard Process for Data Mining (CRISP-DM) model. Digital Image. Wolf, R. *CRISP-DM: Ein Standard-Prozess-Modell für Data Mining.* (2012). https://statistik-dresden.de/archives/1128.

The data used to train the model corresponds to 1000 credit applicants in Germany. It comprises 30 variables for each applicant, with information about their credit history, purpose of credit, credit amount, etc. Please see Appendix A for more information on the data variables.

Nine different machine learning models are trained to classify the applicant's credit as good or bad. Besides, the main focus of the models is to correctly classify applicants presenting bad credit risk, since approving an application that should be denied is considered to be seriously harmful, and must be avoided at all costs. Finally, the models are evaluated comparing their predictions to the true credit rating of the applicants.

2 Modeling

2.1 Data cleaning

Before starting to explore the dataset, there are several aspects to inspect. The data does not contain any missing values. Nevertheless, it does contain errors and outliers. Firstly, two invalid inputs are found: one observation with $EDUCATION = -1$, and another with $GUARANTOR = 2$. Both are binary variables, with value 0 or 1. It is not possible to guess what the the correct inputs are, and both observations are consequently removed from the dataset. Secondly, a clear outlier is found: an applicant whose age is 125 years. Although removing the observation is a valid option, it is decided to modify this observation's AGE to the median value of all observations.

As far as variable transformations are concerned, two new variables are created: CAR, and $GENDER$. The first one is a combination of NEW_CAR and $USED\_CAR$, describing whether the applicant has a new, used, or no car. The latter, on the other hand, describes whether the applicant is a female or male, and it is obtained from the $MALE\_DIV$, $MALE\_SINGLE$, and $MALE\_MAR\_or\_WID$ variables. Unfortunately, none of these new variables improves the models. Thus, they are not included in the model training data. The variable $OBS\#$ is not included either, since it provides no relevant information about the applicant.

2.2 Exploratory Data Analysis

The main drawback of the data is its imbalance; 700 observations correspond to applicants presenting good credit risk, while 300 observations correspond to applicants presenting bad credit risk. This negatively affects the models' prediction abilities, since having more information about good credit applicants than bad credit applicants results in better predictions for the first than the latter. Furthermore, as previously mentioned, the main focus of the models is to correctly detect bad credit applicants. Thus, some sort of restriction will be imposed to the predictions.

In order to gain the first insights regarding the influence of each variable on the final prediction, two types of analysis are done: one for numerical variables, and another for categorical variables. Decisive variables are likely to show different tendencies for each response type (good/bad).

Numerical variables are explored visualizing the boxplots, histograms and density plots of applicants with good or bad credit. As an example, see Figure 2, where the variable $AMOUNT$ (credit amount) is inspected. All plots show similar distributions for both response types, although applicants with negative credit risk seem to tally with larger credit amounts.

Similarly, applicants with negative credit risk tally with longer credit duration, and younger age. Please see Appendix B for the visualization of all variables.

On the other hand, the examination of categorical variables is done visualizing the proportion of good/bad credit risk applicants on each category of the variable. As an example, see Figure 3, where the variable CHK_ACCT (checking account status) is inspected. The plot shows a clear distinction between response types; good credit applicants tend to have over 200DM[1], or no checking account, while bad credit applicants tend to have less than 200DM. Additionally, the difference between mean values of bad credit applicants (dashed line), and good credit applicants (solid line), provides coherent discernment. These results imply that the variable CHK_ACCT plays an important role on the final decision.

Similarly, the following variables are found to be potentially influential: $HISTORY$, and SAV_ACCT. Please see Appendix B for the visualization of all variables.

Unfortunately, as previously discussed, the final decision cannot be made based uniquely on one variable. Thus, machine learning models are considered, which allow the analysis of all applicant information combined.

[1] official currency of West Germany from 1948 until 1990

Figure 2. Boxplot, histogram and density plot of applicants presenting good (blue) or bad (red) credit risk, for the numerical variable $AMOUNT$.

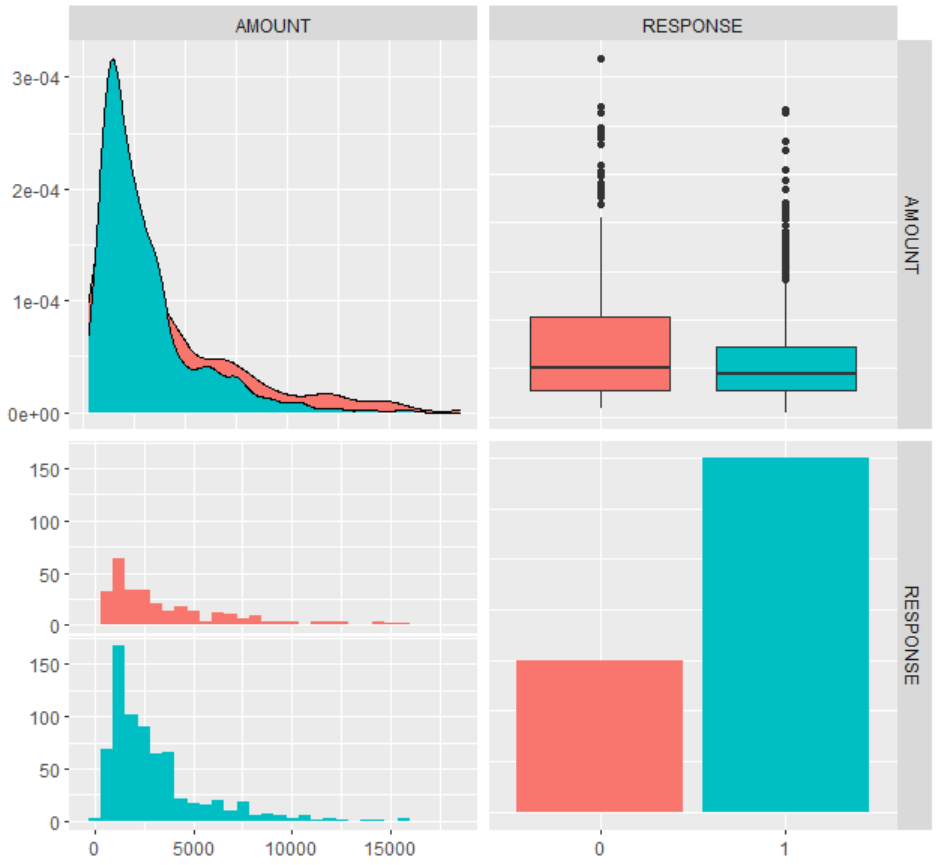
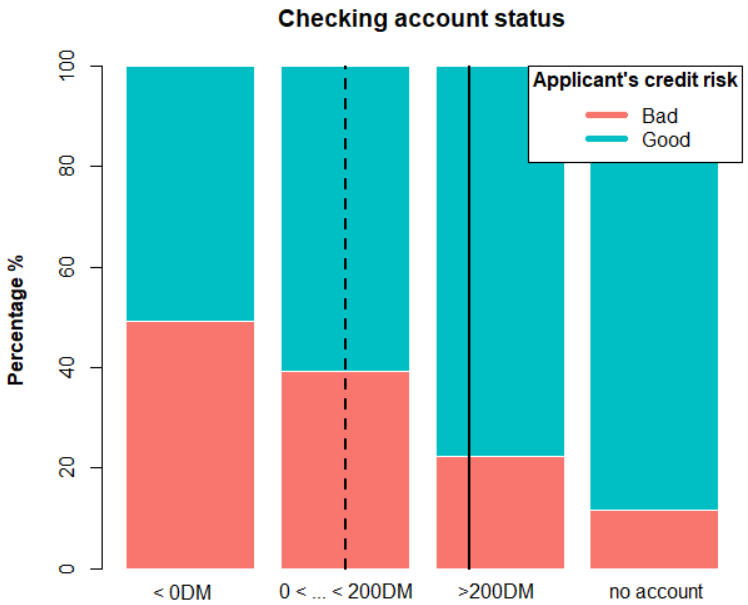


Figure 3. Proportion of applicants presenting good or bad credit risk, for the categorical variable CHK_ACCT. The dashed line represents the mean value of bad credit applicants, and the solid line represents the mean value of good credit applicants.



2.3 Train-test split

In order to evaluate the models fairly, the data is split into two subsets: a training set, and a test set. The training set is the data that is fed to the model, from which it 'learns'. On the other side, the test set is the data - unseen by the model during the 'learning' process - used to evaluate the model's performance. The training set comprises 75% of the cleaned data (749 observations), and the test set comprises 25% (249 observations). All models use the same training and testing set.

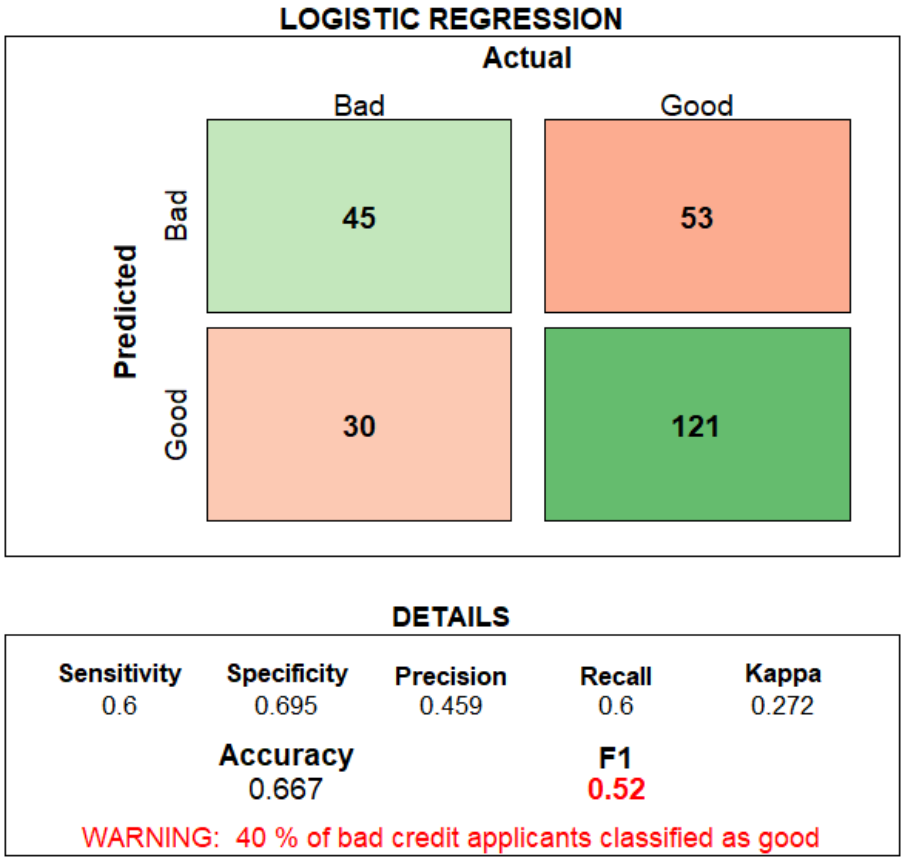
2.4 Threshold

As stated before, the aim is to correctly detect bad credit applicants, and a restriction is imposed: a threshold. Along with the prediction, models provide a corresponding probability. The threshold determines the minimum probability required for applicants to be classified as good. On a conventional model, the applicant is classified as good if the prediction probability for 'good credit' is above 0.5, and as bad if the prediction probability for 'good credit' is below 0.5. A conservative model will require a higher confidence to grant a credit, yielding to a lower false negative rate (i.e. bad credit applications that are approved). Thus, a threshold of 0.75 is set; the models will only classify the applicant as good if the prediction probability for 'good credit' is above 0.75.

2.5 Logistic regression

Logistic regression is widely used in binary classification problems. Similar to linear regression, the result is the impact - or weight - of each variable, on the odds ratio of the response variable. The performance of the logistic regression model can be assessed from its corresponding confusion matrix (see Figure 4).

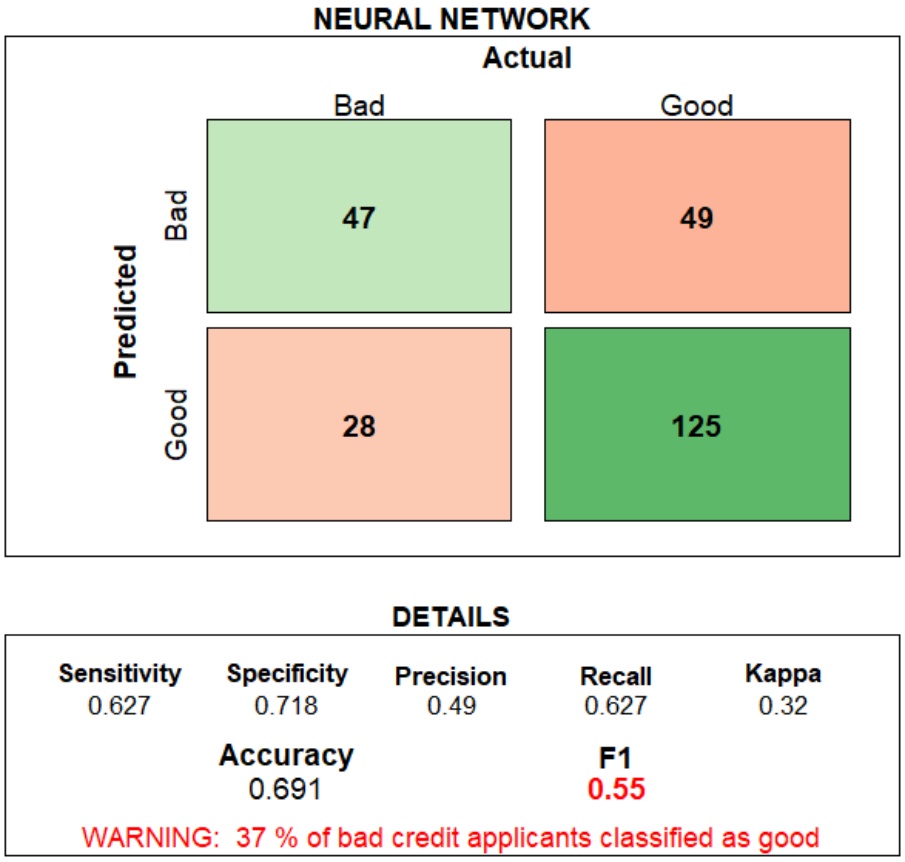
Figure 4. Confusion matrix of the logistic regression model.



2.6 Neural network

Feed-forward neural networks can be understood as a nonlinear regression model. First, the input values are pre-processed: re-scaled to [0,1]. This is done in order to speed up learning and reach convergence. Then, the inputs are weighted within a set of hidden layers to produce the output. Compared to logistic regression, neural networks are more flexible, although this can also result in overfitting, due to the large number of parameters to estimate. The *size*, number of units in the one and only hidden layer of the network, is chosen via 10-fold cross validation. Additionally, the *decay*, regularization parameter to avoid overfitting, is also chosen via cross validation. The highest accuracy is obtained with $size = 1$ and $decay = 0.1$. The performance of the neural network model can be assessed from its corresponding confusion matrix (see Figure 5).

Figure 5. Confusion matrix of the neural network model.



2.7 Tree

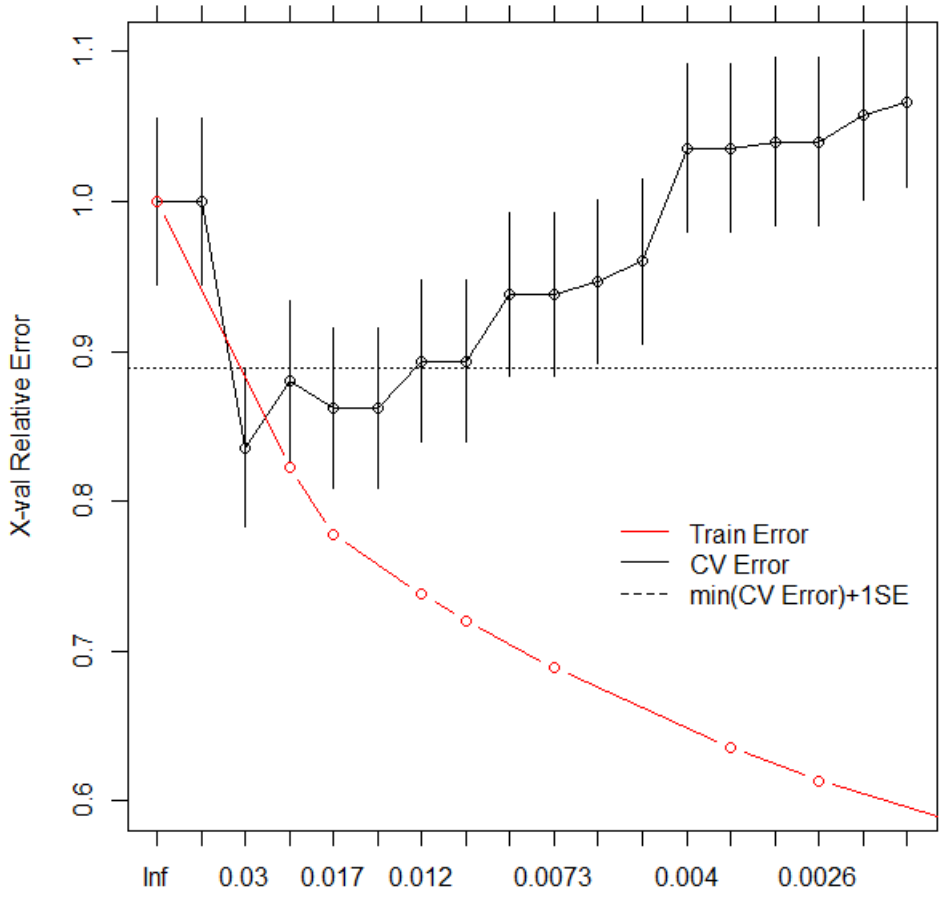
Decision trees break down complex data into more manageable parts; the tree, with a flowchart like structure constructed through repeated splits of subsets (nodes), considers all possible outcomes of a decision and traces each path to a conclusion.

First, a tree is built using all variables from the data. Two main hyperparameters are defined: $minsplit = 4$, and $cp =$ 1e-05. *minsplit* is the minimum number of observations required in a node to be split further. The complexity parameter *cp* is the cost of adding another variable to the decision tree, and it is used to control and select the optimal tree size. These settings result in a tree with 115 nodes.

Such complex tree is likely to overfit the data. Therefore, the tree is pruned, with the goal of reducing overall complexity, without reducing predictive accuracy. The *cp* of the pruned tree is determined as

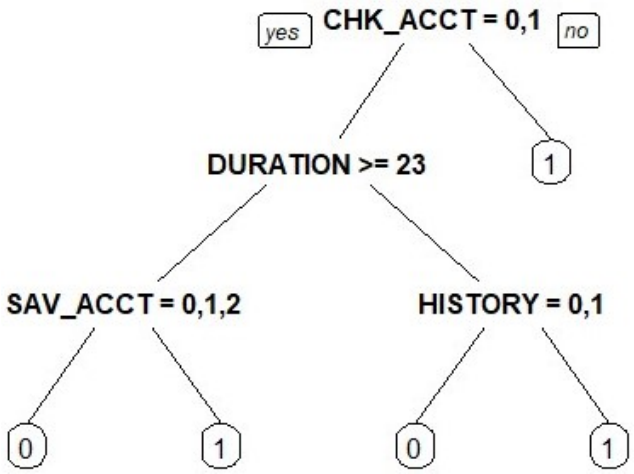
the largest one, whose cross validation error is smaller than the minimum cross validation error + cross validation standard error (see Figure 6). The aforementioned overfit is clearly visible on Figure 6, where the training error always decreases as the complexity of the tree increases, but the cross validation error increases after a certain point.

Figure 6. Cross validation (CV) error for different complexity parameter trees (x-axis).



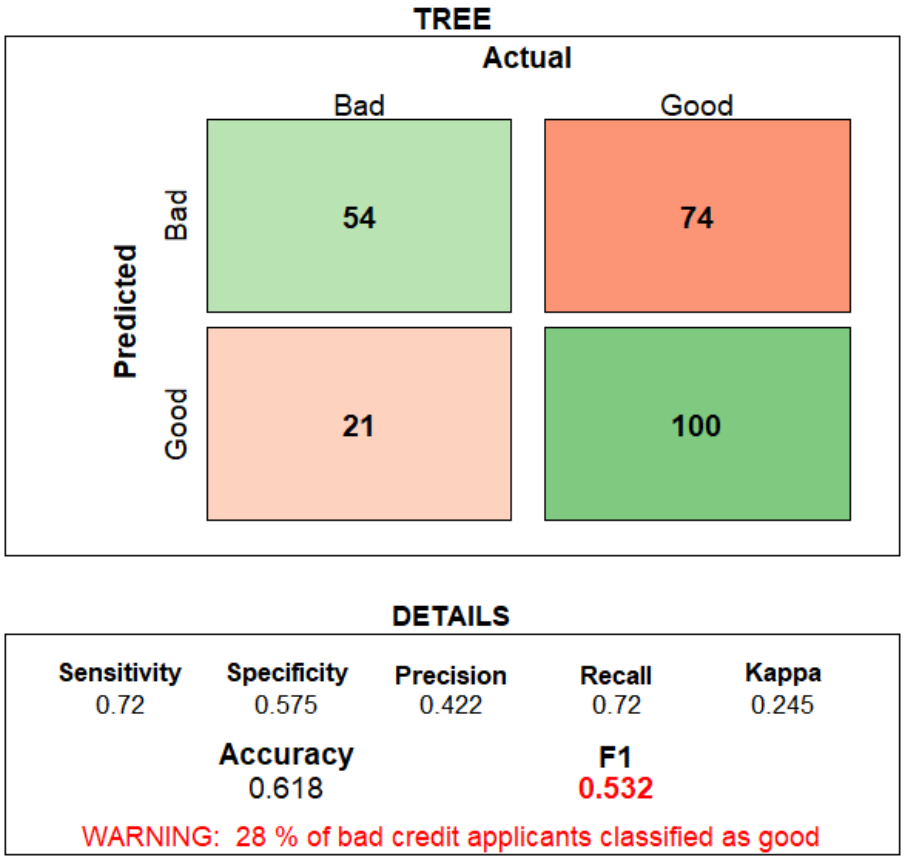
The complexity parameter of the pruned tree is $cp = 0.02$, and it results in a tree with 5 nodes (see Figure 7).

Figure 7. Pruned tree with complexity parameter $cp = 0.02$.



The performance of the pruned decision tree can be assessed from its corresponding confusion matrix (see Figure 8).

Figure 8. Confusion matrix of the pruned decision tree model.



2.8 Random forest

As an extension of decision trees, random forest models utilize bagging (random sampling of the training data) and randomized feature selection to combine uncorrelated decision trees, which yields a more accurate and stable performance. There are several hyperparameters to define, the first one being *ntree*: the number of trees to grow. Figure 9 shows how accuracy improves with more trees, and stabilizes after approximately 100 trees. Another hyperparameter is *mtry*: the number of variables randomly sampled as candidates at each split. Lastly, *nodesize* is the minimum size of terminal nodes. In this study *ntree* = 500, $mtry = 5 \approx \sqrt{\#variables}$, and $nodesize = 10$. The performance of the random forest can be assessed from its corresponding confusion matrix (see Figure 10).

Figure 9. Prediction error of random forest models composed of different number of trees.

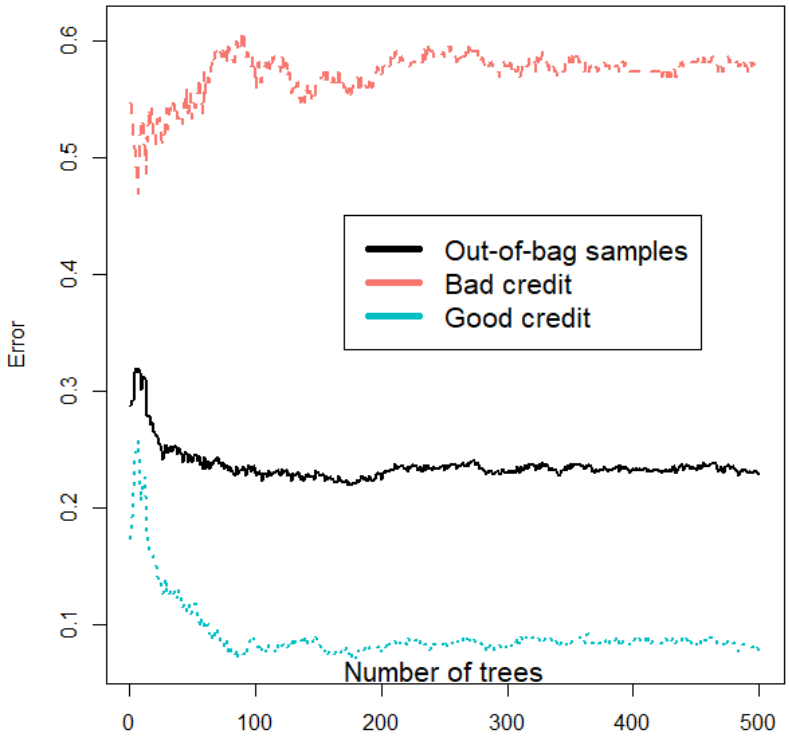
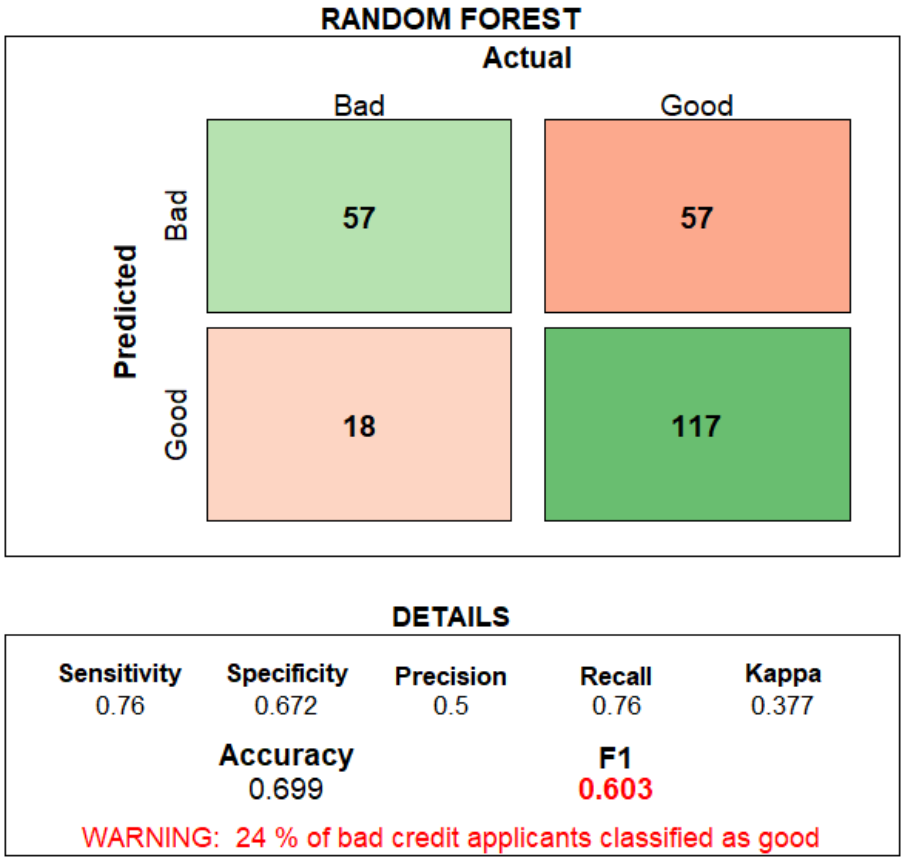


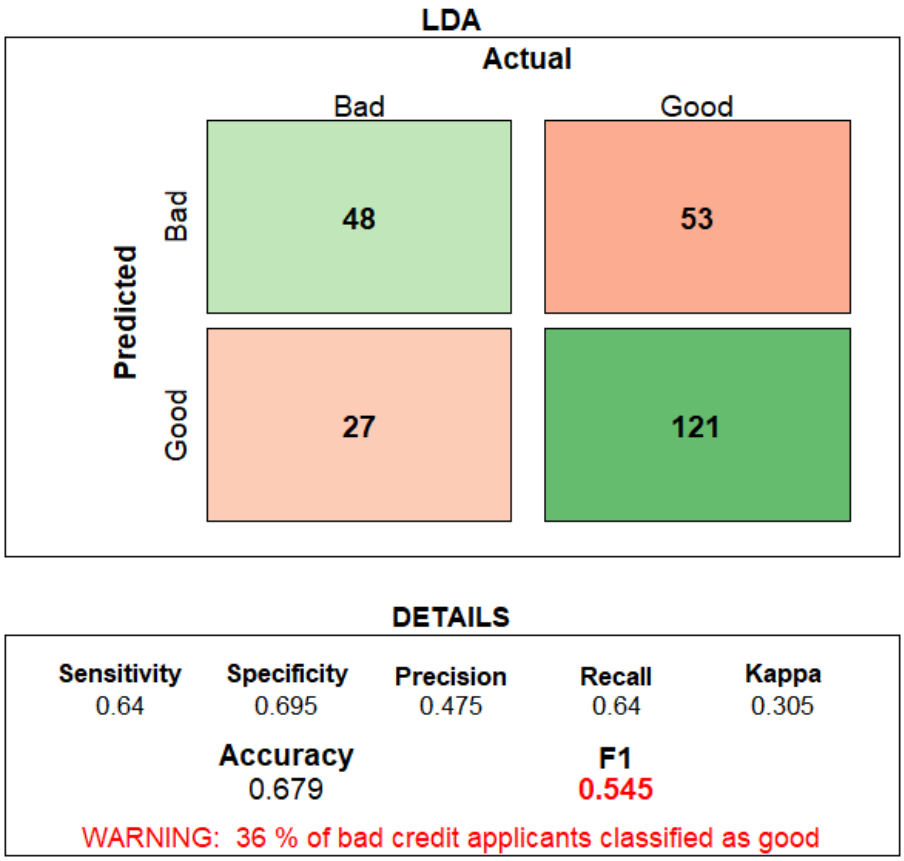
Figure 10. Confusion matrix of the random forest model.



2.9 LDA

Linear Discriminant Analysis (LDA) uses linear combinations of variables to separate the data into categories. LDA can be affected by the scale of the variables, thereby requiring standardization. Thus, numerical data is centered and scaled before being fed to the model. The performance of the LDA model can be assessed from its corresponding confusion matrix (see Figure 11).

Figure 11. Confusion matrix of the LDA model.



2.10 KNN

K-Nearest Neighbors (KNN) models are based on the distances between observations. An observation is classified as the most frequent category among its closest K observations. The value of K is determined via 5-fold cross validation, with values ranging from 1 to 100. The performance of the KNN model can be assessed from its corresponding confusion matrix (see Figure 12).

2.11 SVM

Support-Vector Machines (SVM) create a hyperplane which separates the data into categories. The data is transformed, using a technique called the kernel trick, mapping it to a high-dimensional feature space where data points can be categorized, and the hyperplane is found. SVM models are able to model complex nonlinear decision boundaries. The hyperparameter to define is the kernel type; in this study a linear kernel is chosen. The performance of the SVM model can be assessed from its corresponding confusion matrix (see Figure 13).

Figure 12. Confusion matrix of the KNN model.

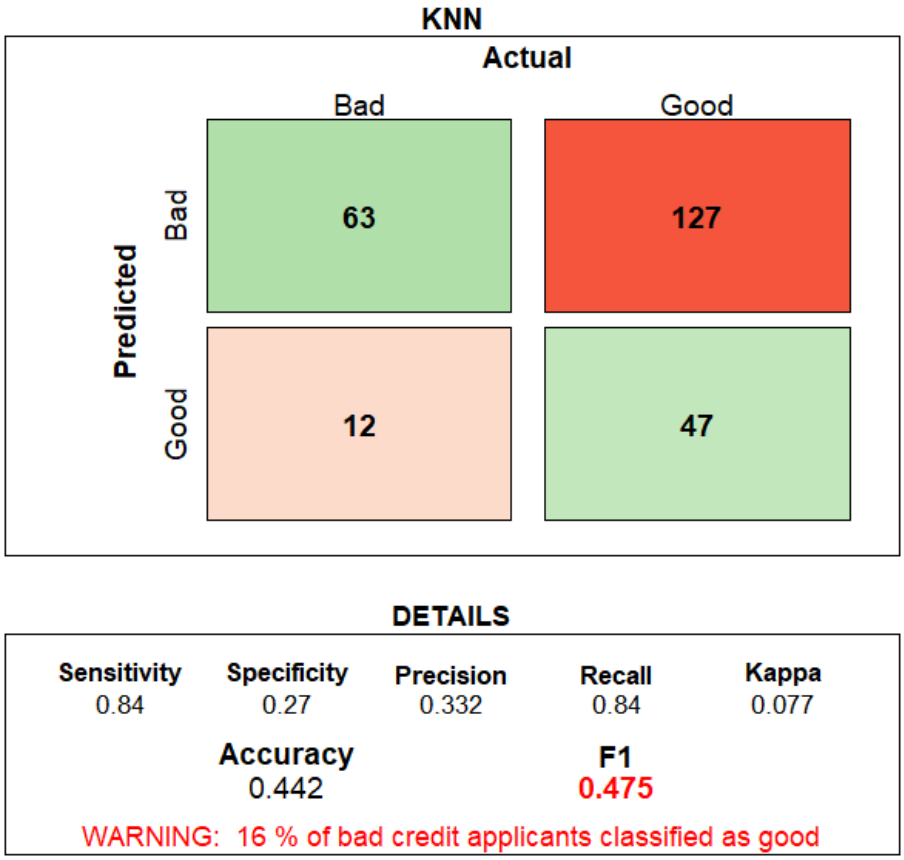
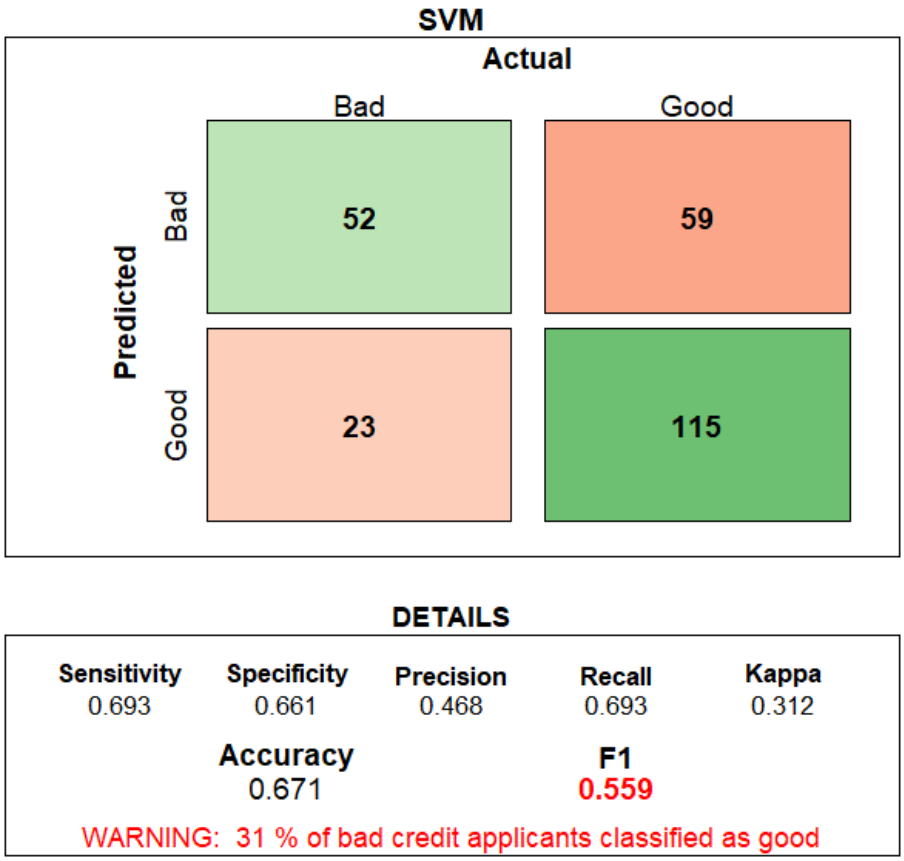


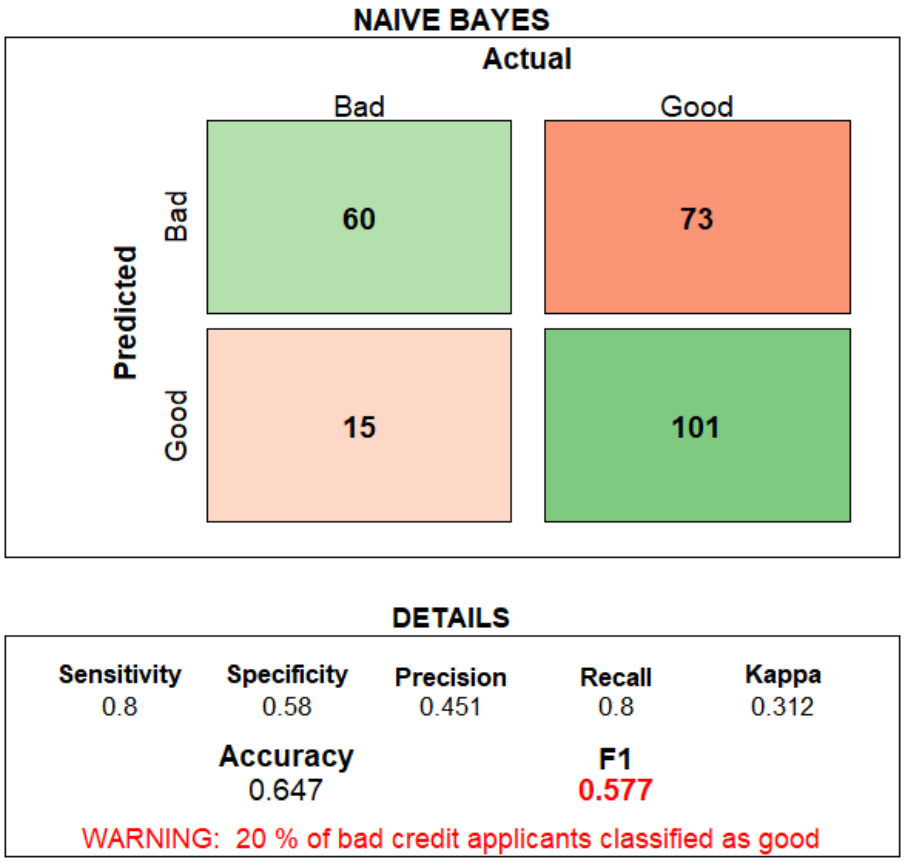
Figure 13. Confusion matrix of the SVM model.



2.12 Naive-Bayes

Naive-Bayes models are probabilistic classifiers that use the Bayes Theorem; they predict the probability that a given observation belongs to a particular category. The Naive-Bayes classifier assumes the variables to be mutually independent, unlike, for instance, LDA models, which allow correlated variables. Due to the imbalance of the data, the prior vector with probabilities of each category is specified as (0.5,0.5). The performance of the Naive-Bayes model can be assessed from its corresponding confusion matrix (see Figure 14).

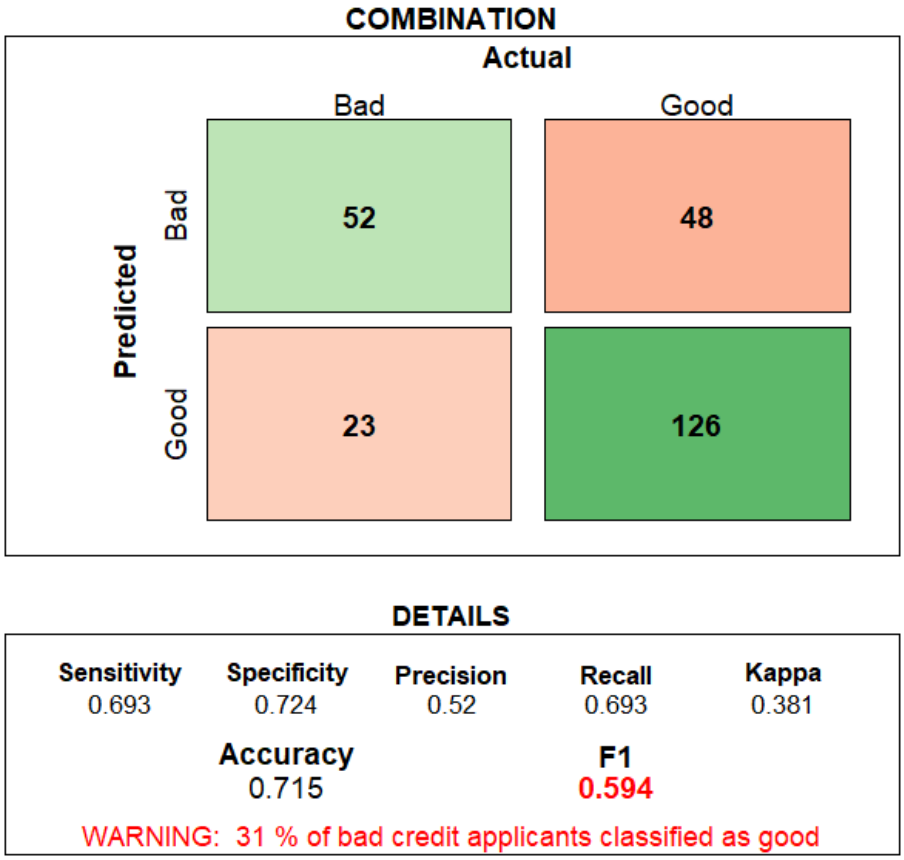
Figure 14. Confusion matrix of the Naive-Bayes model.



2.13 Combination

Lastly, a simple ensemble method is tested in an attempt to improve results by combining all previous models. The method consists of classifying an applicant as good only if all models predict a good credit risk; as long as one of the models predicts a bad credit risk, the application will be denied. Nonetheless, the previous models were already restricted to focus on bad credit applicants, meaning that the combination of these conservative models leads to the denial of almost all applications. Thus, the ensemble method is applied on non-restricted - or conventional - models, with a threshold of 0.5. The performance of the ensemble method can be assessed from its corresponding confusion matrix (see Figure 15).

Figure 15. Confusion matrix of the ensemble method, which combines all previous models.



3 Discussion

The models' performance is assessed from their corresponding confusion matrices, particularly the F1 and false negative rate values. The F1 value, harmonic mean of precision and recall, is the suggested metric to evaluate imbalanced datasets. The false negative rate denotes the percentage of bad credit applications that are approved. Thus, a high F1 and low false negative rate are sought. The metrics of each model are summarized in Table 1.

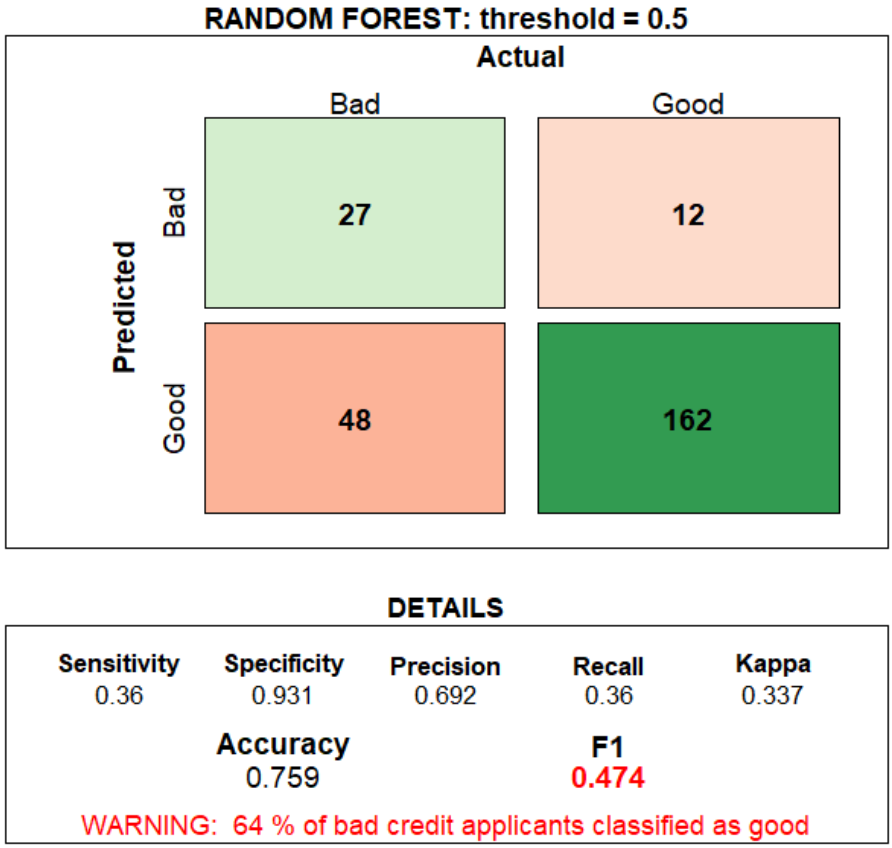
Table 1. Summary of the metrics of each model.

Model	F1	False negative rate
Logistic regression	0.520	40
Neural network	0.550	37
Decision tree	0.532	28
Random forest	0.603	24
LDA	0.545	36
KNN	0.475	16
SVM	0.559	31
Naive-Bayes	0.577	20
Combination	0.594	31

Based on this criteria, the random forest model performs the best; it has the highest F1, and the third lowest false negative rate. The lowest false negative rate is obtained with the KNN model. However, the model seems to be 'too focused' on bad credit applicants, and its F1 score is significantly the lowest. The second lowest false negative rate is obtained with the Naive-Bayes model. In this case, the model's F1 score is slightly lower than the one of random forest. Therefore, Naive-Bayes is concluded to be a trustworthy model too.

As far as the threshold value is concerned, 0.75 yields acceptable results. In fact, the performance of the models, based on the aforementioned metrics, improves compared to the predictions obtained with a threshold of 0.5. The random forest model performance with a threshold of 0.5 is shown in Figure 16. While the accuracy is higher in this case, the F1 and false negative rate values are worse than the ones of the model with a threshold of 0.75 (see Figure 10). These results corroborate the choice of F1, instead of accuracy, for instance, as evaluation criteria.

Figure 16. Confusion matrix of the random forest model, with a threshold of 0.5.



Additionally, the random forest model outlines the influence of each variable on the final prediction. The most important variables are:

1. Checking account status (CHK_ACCT)
2. Credit duration ($DURATION$)
3. Credit history ($HISTORY$)
4. Average balance in savings account (SAV_ACCT)
5. Credit amount ($AMOUNT$)

These results agree with the hypotheses made on the exploratory data analysis.

3.1 Future work

There are several aspects that could be studied and potentially lead to an improvement on the models' performance:

- Hyperparameter tunning: although cross validation is occasionally employed, there are numerous hyperparameters that could be optimized.
- More research could be done regarding the imposed threshold; different threshold values could be tested for each model.
- Model assumptions should be verified; e.g. LDA assumes the variables to be normally distributed and the categories to have identical covariance matrices.
- Further models and model extensions could be tested; e.g. Quadratic Discriminant Analysis (QDA).

4 Conclusion

Nine models are considered to classify credit applicants based on their credit risk. In an attempt to deal with the imbalanced nature of the data, and with the aim to focus on bad credit detection, a threshold of 0.75 is set for good credit predictions. The analysis reveals that random forest is the most appropriate classification method, achieving a 0.603 F1 value and 24% false negative rate.

Furthermore, the model provides insights regarding the influence of variables on the final prediction; checking account status (CHK_ACCT), credit duration ($DURATION$), credit history ($HISTORY$), average balance in savings account (SAV_ACCT), and credit amount ($AMOUNT$), are found to be the most important variables.

Appendix A

Figure 17. Variable information of the data.

Var.#	Variable Name	Description	Variable Type	Description
1.	OBS#	Observation No.	Categorical	
2.	CHK_ACCT	Checking account status	Categorical	0 : < 0 DM 1 : 0 < ⋯ < 200 DM 2 : ≥ 200 DM 3 : no checking account
3.	DURATION	Duration of credit in months	Numerical	
4.	HISTORY	Credit history	Categorical	0 : no credits taken 1 : all credits at this bank paid back duly 2 : existing credits paid back duly till now 3 : delay in paying off in the past 4 : critical account
5.	NEW_CAR	Purpose of credit	Binary	car (new) 0 : No, 1 : Yes
6.	USED_CAR	Purpose of credit	Binary	car (used) 0 : No, 1 : Yes
7.	FURNITURE	Purpose of credit	Binary	furniture/equipment 0 : No, 1 : Yes
8.	RADIO/TV	Purpose of credit	Binary	radio/television 0 : No, 1 : Yes
9.	EDUCATION	Purpose of credit	Binary	education 0 : No, 1 : Yes
10.	RETRAINING	Purpose of credit	Binary	retraining 0 : No, 1 : Yes
11.	AMOUNT	Credit amount	Numerical	
12.	SAV_ACCT	Average balance in savings account	Categorical	0 : < 100 DM 1 : 100 ≤ ⋯ < 500 DM 2 : 500 ≤ ⋯ < 1000 DM 3 : ≥ 1000 DM 4 : unknown/no savings account
13.	EMPLOYMENT	Present employment since	Categorical	0 : unemployed 1 : < 1 year 2 : 1 ≤ ⋯ < 4 years 3 : 4 ≤ ⋯ < 7 years 4 : ≥ 7 years
14.	INSTALL_RATE	Installment rate as % of disposable income	Numerical	
15.	MALE_DIV	Applicant is male and divorced	Binary	0 : No, 1 : Yes
16.	MALE_SINGLE	Applicant is male and single	Binary	0 : No, 1 : Yes
17.	MALE_MAR_WID	Applicant is male and married or a widower	Binary	0 : No, 1 : Yes
18.	CO−APPLICANT	Application has a co-applicant	Binary	0 : No, 1 : Yes
19.	GUARANTOR	Applicant has a guarantor	Binary	0 : No, 1 : Yes
20.	PRESENT_RESIDENT	Present resident since - years	Categorical	0 : ≤ 1 year 1 : 1 < ⋯ ≤ 2 years 2 : 2 < ⋯ ≤ 3 years 3 :> 4 years
21.	REAL_ESTATE	Applicant owns real estate	Binary	0 : No, 1 : Yes
22.	PROP_UNKN_NONE	Applicant owns no property (or unknown)	Binary	0 : No, 1 : Yes
23.	AGE	Age in years	Numerical	
24.	OTHER_INSTALL	Applicant has other installment plan credit	Binary	0 : No, 1 : Yes
25.	RENT	Applicant rents	Binary	0 : No, 1 : Yes
26.	OWN_RES	Applicant owns residence	Binary	0 : No, 1 : Yes
27.	NUM_CREDITS	Number of existing credits at this bank	Numerical	
28.	JOB	Nature of job	Categorical	0 : unemployed/unskilled - non-resident 1 : unskilled - resident 2 : skilled employee/official 3 : management/self-employed/ highly qualified employee/officer
29.	NUM_DEPENDENTS	Number of people for whom liable to provide maintenance	Numerical	
30.	TELEPHONE	Applicant has phone in his or her name	Binary	0 : No, 1 : Yes
31.	FOREIGN	Foreign worker	Binary	0 : No, 1 : Yes

Appendix B

Figure 18. Boxplots, histograms and density plots of applicants presenting good (blue) or bad (red) credit risk, for all numerical variables.

Figure 19. Proportion of applicants presenting good (blue) or bad (red) credit risk, for all categorical variables. The dashed lines represent the mean value of bad credit applicants, and the solid lines represent the mean value of good credit applicants.

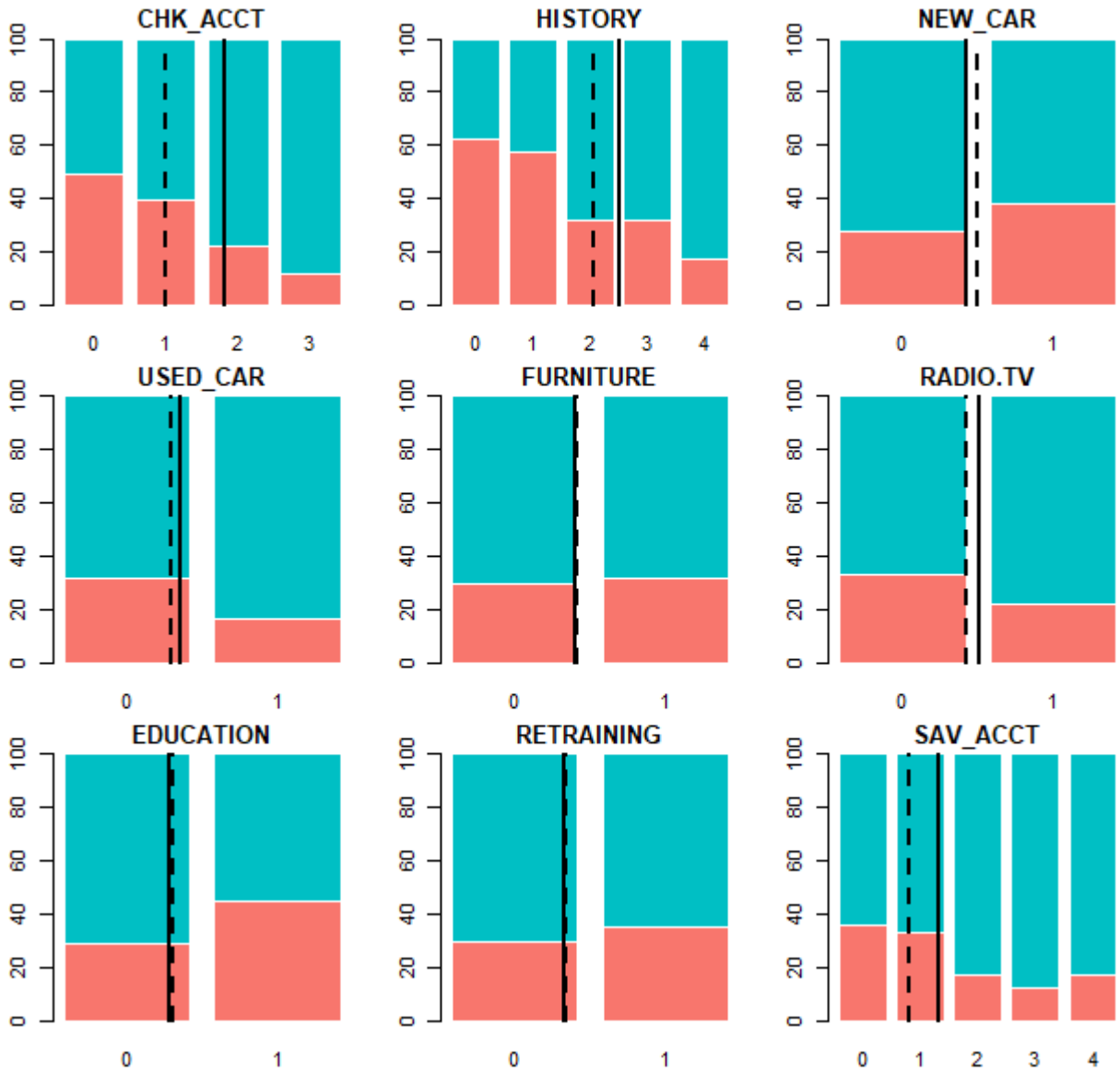


Figure 20. Proportion of applicants presenting good (blue) or bad (red) credit risk, for all categorical variables. The dashed lines represent the mean value of bad credit applicants, and the solid lines represent the mean value of good credit applicants.

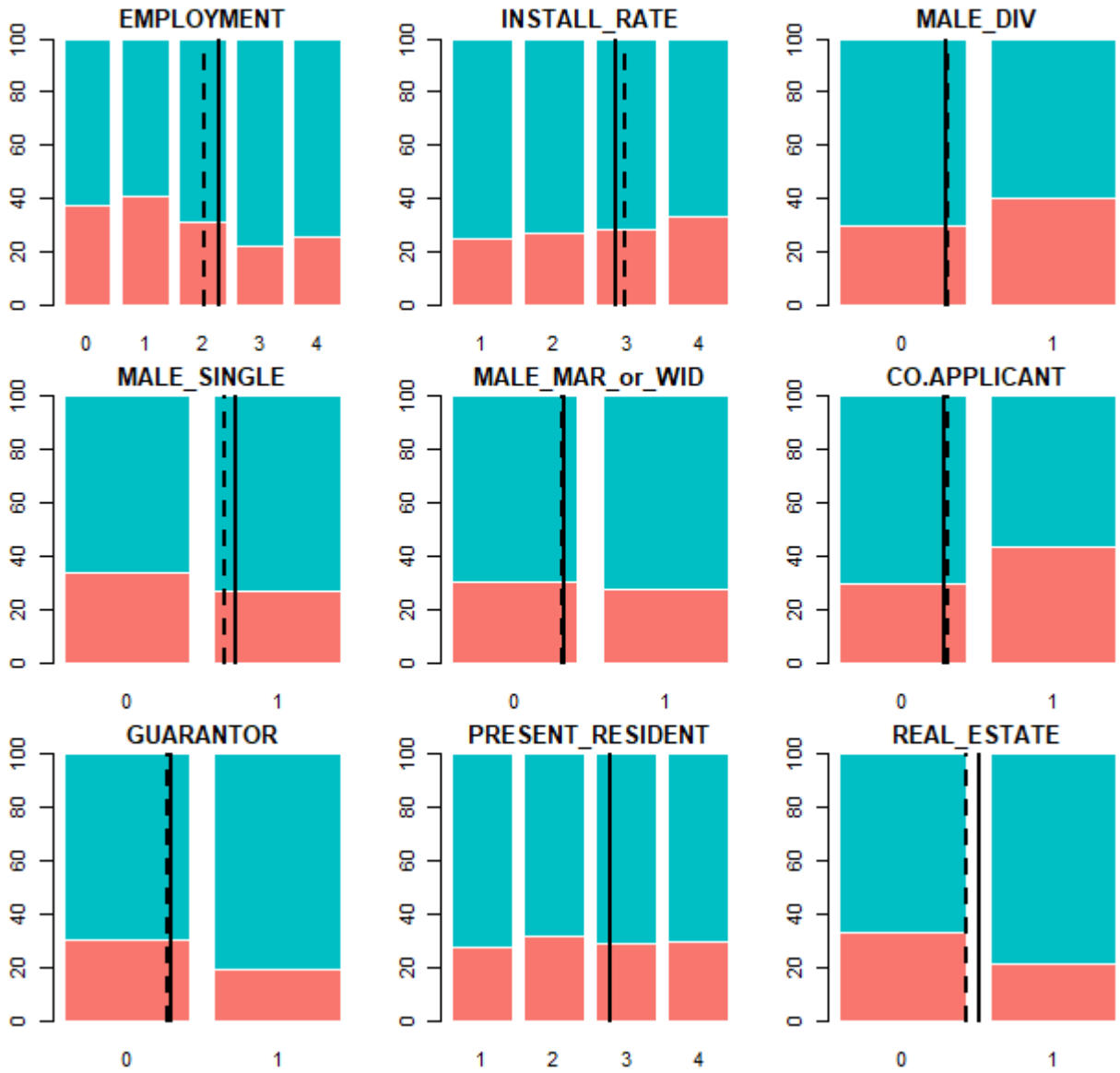
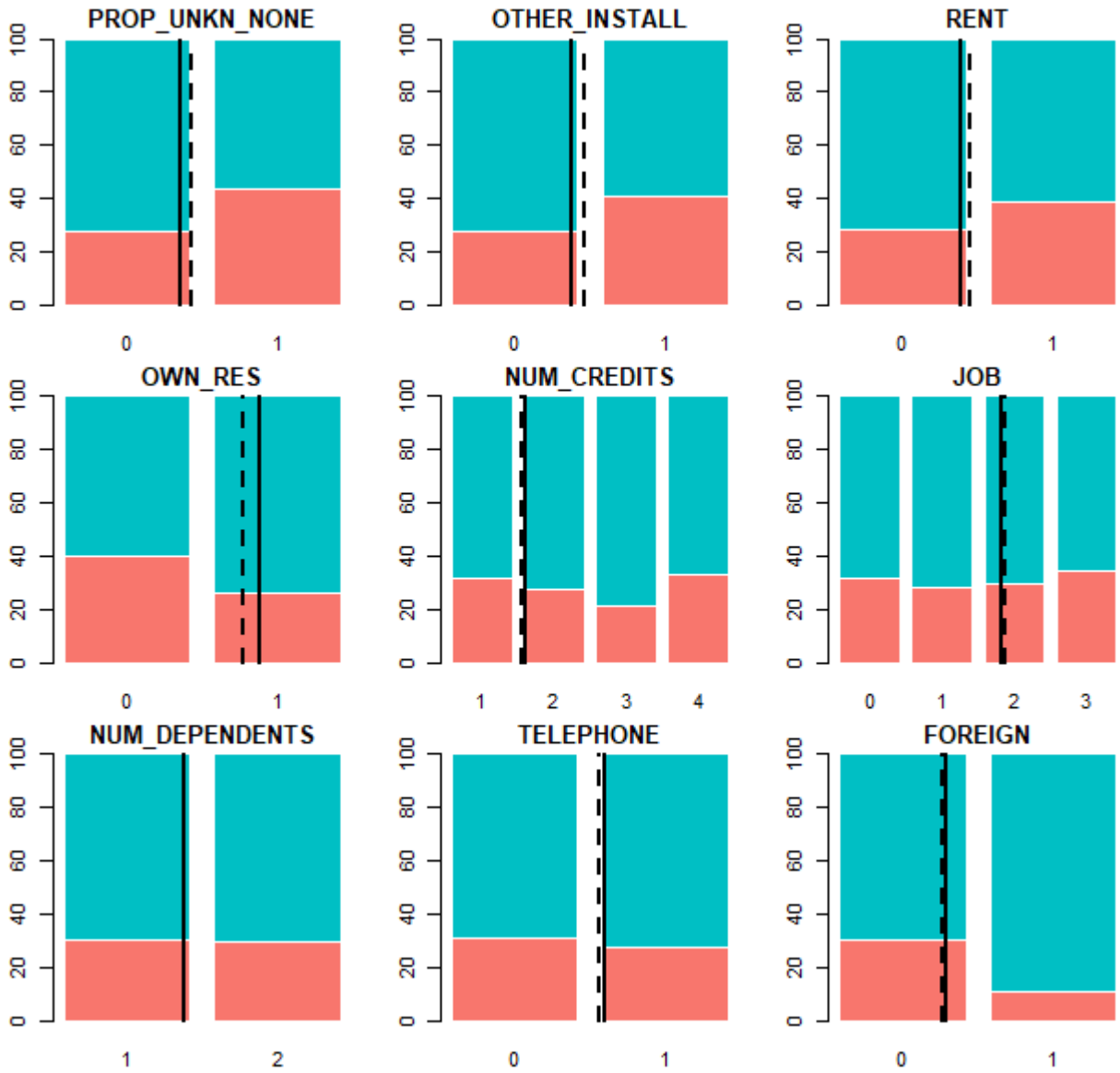


Figure 21. Proportion of applicants presenting good (blue) or bad (red) credit risk, for all categorical variables. The dashed lines represent the mean value of bad credit applicants, and the solid lines represent the mean value of good credit applicants.



References

[1] Zuber, J. *Data Analytics for Decision Making.* (2022).

[2] Wolf, R. *CRISP-DM: Ein Standard-Prozess-Modell für Data Mining.* (2012). https://statistik-dresden.de/archives/1128

[3] Breck. *R how to visualize confusion matrix using the caret package.* (2018). https://stackoverflow.com/a/53235386

[4] Kassambara, A. *Discriminant Analysis Essentials in R.* (2018). http://www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/

[5] ocram. *KNN and K-folding in R.* (2017). https://stats.stackexchange.com/a/318969

[6] Kumar, A. *Classifying data using Support Vector Machines(SVMs) in R.* (2021). https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-r/

[7] FINNSTATS. *Naive Bayes Classifier in Machine Learning.* (2021). https://finnstats.com/index.php/2021/04/08/naive-bayes-classification-in-r/