Data Analytics for Decision Making: German Credit Risk Analysis

Isa Castro De Sousa
Anaïs Zodeougan-Quist

June 5, 2019

Contents

1 Introduction

Credit rating is defined as the evaluation of the creditworthiness of a borrower with respect to a set of criteria in order to predict their capability of debt repayment(Myers et al., 1991). Therefore, credit rating may be seen as a quantitative asset which helps credit providers to grant credits. Given the German Credit dataset provided by our Professor Zuber. Note however, that the dataset in its original format comes from the Professor Hans Hofmann (Univsersität Hamburg)[1].

The objective of this project is to build a predictive model to determine whether a new applicant will represent a credit hazard for the bank or not. This, by means of different statistical learning models for decision making. Thereby, after depicting the dataset throughout an exploratory analysis of the data, six methods will be detailed and compared to select the most effective in detecting credit risk.

2 Exploratory Data Analysis

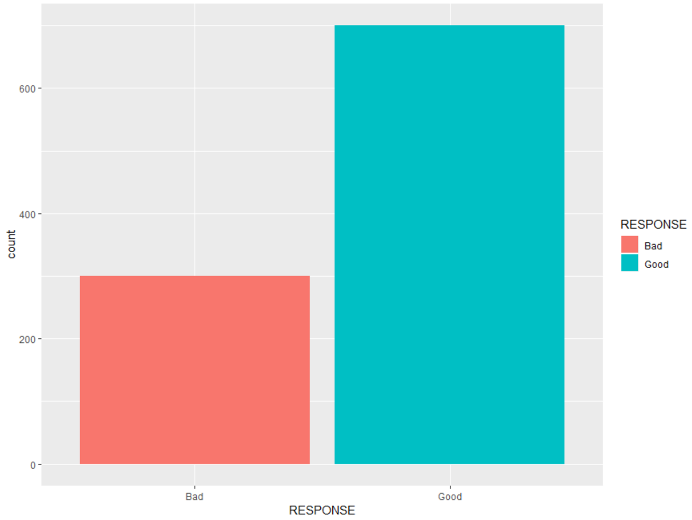
Prior to building the best model using different machine learning methods, the most important part in exploring the data is to clean the dataset accordingly to its description. Indeed, this will avoid to have inaccurate models. In our dataset, we have detected one typo error within the variable EDUCATION. More precisely we have found a '-1' level, whereas the description provided mentioned only two levels '0' or '1'.

Further, the levels for the variable PRESENT_RESIDENT have been corrected in order to have appropriate level starting from '0' to '3' instead of '1' to '4'. We have also identified and removed a useless level within the variable GUARANTOR. Eventually, the variable AGE contained an outlier, following the dataset provider, this has been modified to the value 75 years old (instead of 125 years).

2.1 Response variable description

Thereby the dataset 30 attributes after removing the first column which referred to the individual identification number, and 1000 past credit applicants. Amongst which we find 6 continuous variables, 18 binary variables plus the response variable, and 6 categorical variables.

The variable response corresponds to the risk label, coded as 1 if the past credit applicant has a good credit rating and 0 otherwise. The following barplot, depicts the related repartition of the applicants.



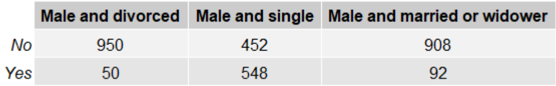
[1] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Figure 1: Repartion of the credit rating

The proportion of applicants classified as risky is of 30% and 70% for the applicants classified as good contractors (cf. AppendixA1). This gives us an unbalanced response variable.
As mentioned *supra*, the dataset contains 30 attributes to classify the past credit applicants. Firstly, we have personal records precising the gender, the age and the civil status.

2.2 Explanatory variables description

Let us specify that this dataset contains information only on male applicants ans their personal status. Indeed, after screening the original dataset, we have denoted both indication on male and female.



	Male and divorced	Male and single	Male and married or widower
No	950	452	908
Yes	50	548	92

Table 1: Repartition of male applicants by civil status

Therefore the repartition in Table1 is that, we have 50 divorced male applicants, 548 single male applicants (which is the largest case), and 92 married/widower male applicants. In the original dataset, these proportions obviously remain, however it does not exclude females repartition (cf. TableA4). Regarding the age of applicants, the distribution goes from 19 years old for the youngest, to 75 for the elderest. Although the table below depicts the 10 most cases, the majority of the applicants are between 22 and 50 (cf. AppendixA2): applicants from 22 to 36 years old represent the most cases, then after we observe a slow decrease of the number of applicants until 50 from where the drop is more obvious (cf. AppendixA6 ,for the whole results).

```
| AGE| count|proportion |
|---:|-----:|:----------|
|  27|    51|5.1 %      |
|  26|    50|5 %        |
|  23|    48|4.8 %      |
|  24|    44|4.4 %      |
|  28|    43|4.3 %      |
|  25|    41|4.1 %      |
|  35|    40|4 %        |
|  30|    40|4 %        |
|  36|    39|3.9 %      |
|  31|    38|3.8 %      |
```

Table 2: Age distribution

Despite the fact that the categories are not very precise, information concerning the past applicants' occupation and its nature are also important factors in the credit rating. Therefore, each time we have computed those with respect to the response variable.

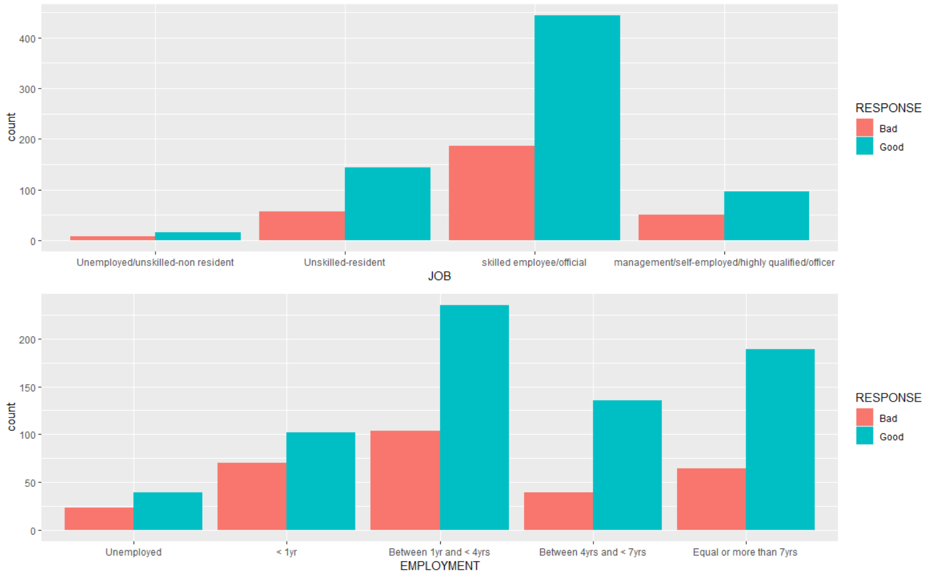


Figure 2: Years of service and occupation tenure

Those two histograms show that we have a large proportion of the past applicants that hold a skilled occupation with years of services comprised between 1 to 4 years and more. Then, still related to the personal records of each past borrower, the dataset contains information on the checking account status, the average balance in savings account and the credit history. As displayed, in the graphic below, it is interesting to notice that individuals with less than 100 Deutsche Mark (DM), represent the majority of past borrowers (and within this attribute a large proportion is rated as good contractors).

Further, applicants that do not hold a checking account in this bank are the most numerous (39.4%, cf. AppendixA5). Category from which, we also observe a greater contingent of good contractors. Last, applicants with higher amount of liquidity (more than 200DM) are the fewest category, representing less than 10% of the total applicants (cf. AppendixA5).

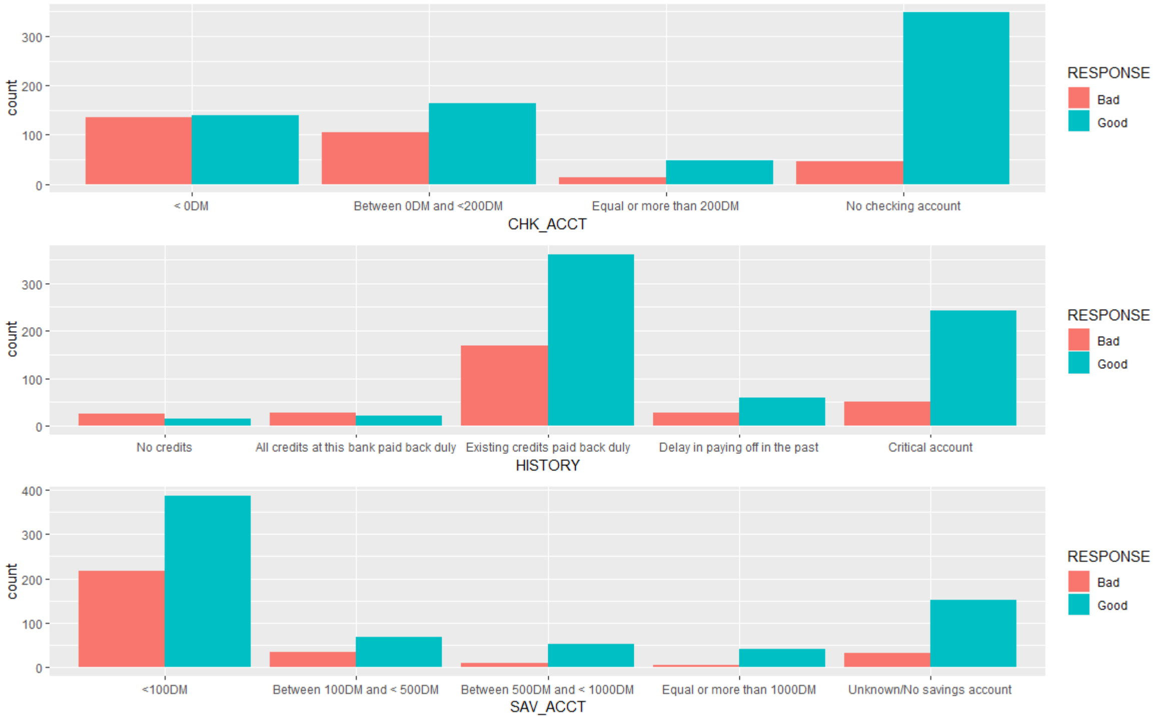


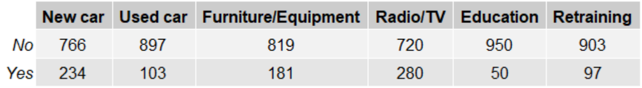
Figure 3: Accounts and credit history information

This fact is also found, in the average balance of saving accounts. Applicants with less than 100DM represent 60.3% of the sample. Notwithstanding the fewest borrowers holding the largest balance on their savings account (cf. AppendixA7), the proportions of good credit scoring is the highest with $(42/(6+42) = 87.5\%)$.
Repayment history is also an important feature of credit scoring since it draws the ability of borrowers of duly repaying their credits. In Figure3 (middle position), the majority of applicants are good payers (53% of the latters having currently credits are paying back duly, cf.AppendixA6). And within this category, 68.1% (361/(361+169))of them are classified as a good credit contractors.
The duration of the credits is expressed in months, and below we have shown the first 7 credit durations frequently asked for, by the past applicants (for the whole results, cf. AppendixA.8). So it is that a majority of borrowers seeks for mid-terms credit, since we have 55.9% of the total credit duration between 12 months and 36 months (the first 4 entries).

```
| DURATION| count| proportion|
|--------:|-----:|----------:|
|       24|   184|       18.4|
|       12|   179|       17.9|
|       18|   113|       11.3|
|       36|    83|        8.3|
|        6|    75|        7.5|
|       15|    64|        6.4|
|        9|    49|        4.9|
```

Table 3: Credit durations

It is also interesting to consider the reasons for opening a credit. The dataset proposes several consumer credits, as described in Table 4 below. The majority of the applicants are granting credits for the acquisition of a new car (234 applicants) and for audiovisual equipments (280 applicants).



	New car	Used car	Furniture/Equipment	Radio/TV	Education	Retraining
No	766	897	819	720	950	903
Yes	234	103	181	280	50	97

Table 4: Types of credit repartition

The motive for not displaying these variables (describing the various purposes for a credit)on a barplot or boxplot form, is because of the extreme contrast in the proportions. And this, for each of these variables. Not that it is necessarily unusual, but generally this can be the fact of information collected from multiple-responses variable coded as binary instances when creating a dataset. It is indeed the case, since all the "Yes"-reponses sum up to 945 instances. After investigating, the original dataset, we find 1000 instances because some of types of credit were not present in the dataset we are analysing (cf.Appendix A4). Which confirmed our prior thinking.
Finally, the amounts borrowed varies almost from an applicant to another (from 18'424DM to 250DM for the leastest amount). Inasmuch as for each amount we count at most 3 applicants, or else 1 or 2 applicants. We obtained these results using the following code,

```
credit %>%
  group_by(AMOUNT) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100,
    proportion = paste(proportion,"%")
  ) %>%
  arrange(desc(AMOUNT)) %>%
  #head(10) %>%
  kable
```

On the Figure4 below, we observe a very skwed distribution with an important concentration of instances between 250DM and less than 5'000DM.

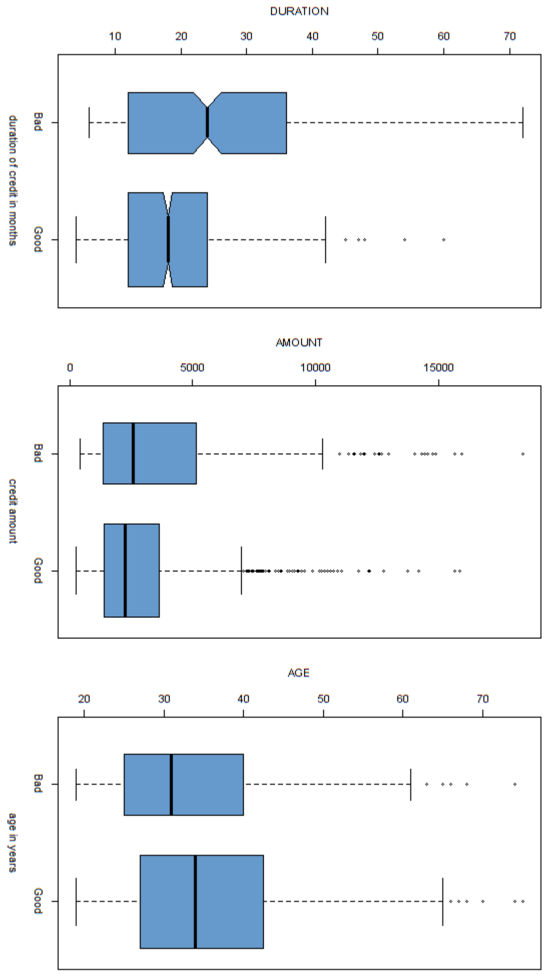


Figure 4: Boxplots of the credit duration and amount, and age applicants

We also notice few outliers within the class of bad contractors with respect to the credit's duration. Meaning that most cases of bad contractors are borrowing between and 40 months, and 5 cases for a significant longer period.The overall distribution within the AMOUNT variable (cf. Appendix A3) shows also skewed distribution. However given what has been said previously regarding the fact that each case has or 1,2 or 3 observation, it is not so surprising to see the numerous dots for the outliers (for both classes). At last, we computed a correlation matrix (cf. AppendixA3)between the 6 continuous variables in order to see if we could detect particular relationships or path, prior to the data modelling. It simply stressed out a moderate uphill relationship between the duration of the credit and its amount. Which is indeed visible when drawing the scatterplot matrix (cf. Appendix A3), where we see that as the amount augments, the duration tends to increase also. Further, we observe that although the variables for the installment rate, the number of existing credits, and the number or people the applicant provides financial support are having sort of categorical variables path. This is due to the fact that we do not have in-between values. For example, the number of existing credits variable (NUM_CREDITS) denotes only integer values from 0 to 4.

3 Data Modelling

In order to build an appropriate model prediction to detect risk credit, we will use 6 commonly used machine learning methods such as logistic regression, classification tree, random forest, neural network, SVM-radial and k-NN. To do so, we have first divided our dataset into 70% and 30% for the training sample and the validation sample in order to test the model. As our dataset has 1000 tuples, 700 random tuples from the dataset has been assigned to the training set and 300 for the testing set.
For each technique, a full model and two different reduced models have been built. The first reduced model has been determined, based on the most important variables of the full model.Therefore, the variables which do not have a significant level to build the model have been eliminated.
Then we have applied the same idea a selecting important feature by the use of the `step()` . Combining the logistic model for the entire predictors, and the function step(), a reduced model has been found, by adding and removing predictors to find the best AIC (Akaike information criterion). For these three distinct models, we applied each machine learning technique using the default parameters in way to have the best predictions. And finally, to improve our models, some tuning methods have been applied to determine the optimal parameters for each method.

3.1 Methodology and approach

In this section we will present, the different methods used to obtain the results for each predictive models. The `caret()` package has served widely in the development of the methods provided, consequently a short description of this package is included.

Caret The `caret()`(short for Classification And REgression Training) is a package which contains plethora of classification and regression and useful functions to create predictive models. Thanks to the possibility to standardized tuning and parameters, this allows to create similar conditions, in other words, the same environment for each learning method. The comparison between models tend then, to be easier.

K-fold cross validation This resampling method requires a split of the dataset training into k-fold (James et al., 2013). The resampling will be performed k times, and each time a unique fold will be conserve as a validation sample and the others k-1 fold will be select to learn. Finally, the mean squared error will be computed for each turn, in the way to average the MSE and obtain the k-fold CV estimate. For this dataset analysis, we have chosen a 10-fold, which is the optimal cross validation according to the literature (Zuber, 2019). K-fold cross validation is an important tool because it can reduce over fitting when examining the models.

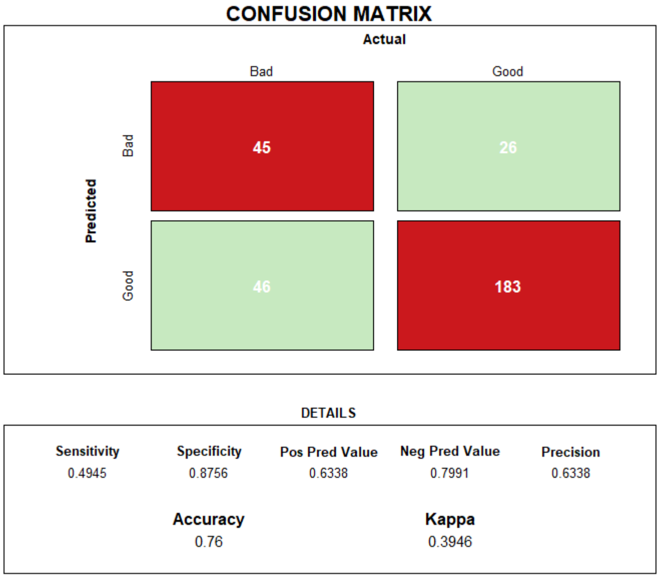
Variable selection The selection of variables is a crucial concept in machine learning, which allows to reduce the variance of models, prevents over fitting issues and ease the model interpretation. This elimination of irrelevant variables can be done in R using the `step ()` function. Which can be apply jointly with the logistic regression algorithm or the function `varImp()`, to select the most relevant predictors.

The Upsampling algorithm This subsampling method provides a good solution for unbalanced classes (which is the case in our response variable). Indeed, this algorithm allows to duplicate the observation of the class with the lowest number of observations, thus improving its signal and its detection power.

The receiver operator characteristic (ROC) and the area under the curve AUC The ROC curve is measured by plotting the sensitivity(true positive rate) against the specificity(false positive rate). By generating this curve, the area under the curve (AUC) can be derived from, and will be used to evaluate the performance of the binary classifier. The AUC is interpreted in terms of a probability, and an AUC equal to 1, means a perfect classification accuracy. Consequently, the higher the AUC value, the better the model explains the data.

3.2 Logistic regression model

In presence of binary variable, we have seen it is best practice to use non-parametric method in order to calculate the probability that each new observation belongs to a certain group. In our instance, we are interested in computing the probability that a new applicant is of a good rating or not. The training set will be used to run the logistic regression model. From the regression outputs, we get that 15 predictors are significant (amongst which we find, predictors describing the checking account the duration, the credit history, the credit purposes for new car and retraining, the amount of credit asked, the saving account balance, the years of employment, the installment rate).



CONFUSION MATRIX

Predicted \ Actual	Bad	Good
Bad	45	26
Good	46	183

DETAILS

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
0.4945	0.8756	0.6338	0.7991	0.6338

Accuracy	Kappa
0.76	0.3946

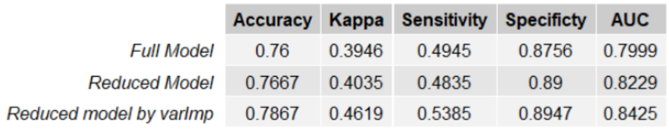
Figure 5: Confusion matrix for the Logistic regression model

`caret()` library has been used to build the classification tree The confusion matrix drawn using the `confusionMatrix()` function computes a set of information. First, it is important to identify the 'positive' class according to the model. Here it is the bad credit rating. To be precise the

- true positive = bad rating applicants identify as such by the model,
- true negative = good rating applicants identify as such by the model,
- false positive = good rating applicants identify as bad by the model,
- false negative = bad rating applicants identify as good by the model.

The model has an accuracy of 76% in classifying new applicants, from which we can derive the error rate for instance ((46+26)/300=24%). It denotes the overall rate of error when classifying new applicants. The precision rate indicates when the model predicts an applicant as a good credit contractor how often this classification is correct. Its computation is based on taking

the true positive predicted over its total (45/(45+26)=0.6338). The specificity which is the true negative rate indicates to what extent the model predicts effectively the good contractors (true negative/reference_good = 183/209) which is 87.56%. However the model is not very effective to identify the bad rating applicants, the true positive rate (the sensitivity) is at 50% (45/(45+46)=0.4945).
Finally the Cohen's Kappa score is not very high (0.3946) and denotes how well the bank would identify its applicants for a credit without the model knowledge.
Further, the (ROC) is used to select the optimal model using the largest value. The table below summarize all the outputs obtained for the full model, the reduced model (based on predictors significance, and the reduced model using the `varImp()` function.



	Accuracy	Kappa	Sensitivity	Specificty	AUC
Full Model	0.76	0.3946	0.4945	0.8756	0.7999
Reduced Model	0.7667	0.4035	0.4835	0.89	0.8229
Reduced model by varImp	0.7867	0.4619	0.5385	0.8947	0.8425

Table 5: Summary of the Logisitic models precision

And regarding the value of the area under the curve (AUC) the best model would be the third model (reduced model). It also provides the highest Kappa, Accuracy and sensitivity.

3.3 Classification tree

The decision tree is a non-parametric supervised method employed for both classification and regression problems. Graphically, this method is handy to interpret and understand the model results(James et al., 2013). The algorithm works by stratifying the predictor space, in various parts. These parts are represented by a tree and the order of the tree branch is determine according to the importance level of the independent variables. Although this method is superior to others in terms of model representation, it also has a major disadvantage: overfitting issues. Figure 6 below, shows the classification tree, according to the whole model. the response levels ('Bad'/'Good') are displayed beneath each terminal node. We can see that the predictors describing the checking account, the duration of the credit, the saving account balance of the applicant and the existence of other installment plan credit have the longest branches (see also AppendixA8), indicating that the discrimination between the good and bad contractors are the most accurate. The method to generate the tree was rather simple compare to the visual quality of the output provided although it is commonly said that decision-tree based are not robust. However it seems to be very suitable to manage many categorical variables as it is the case in our dataset (it is also one of the advantages cited by (James et al., 2013) and (Zuber, 2019).

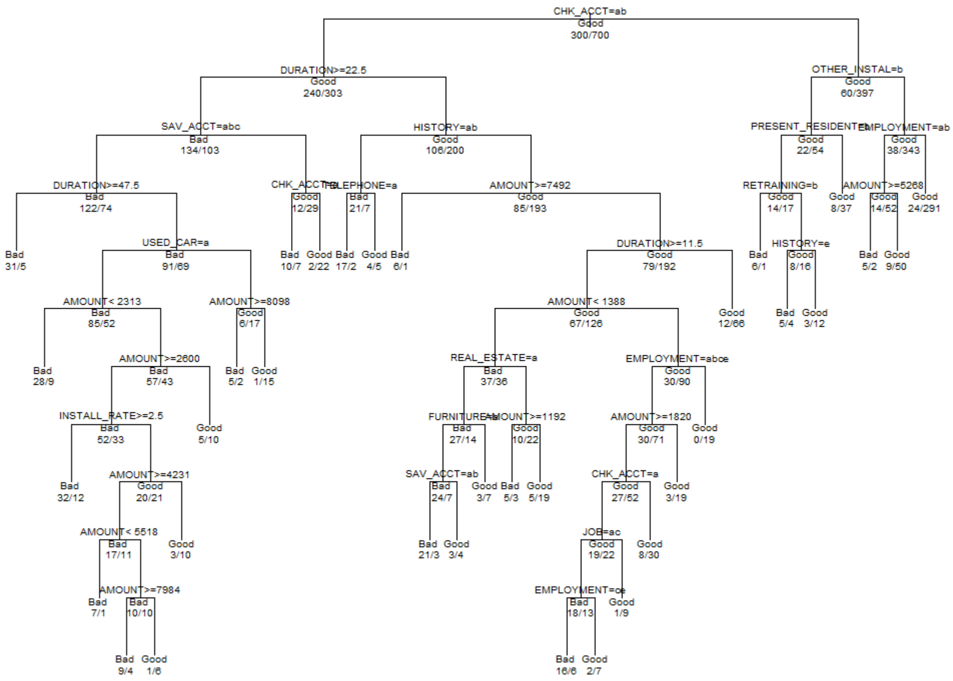
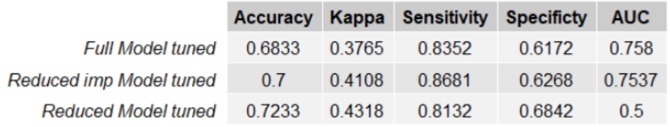


Figure 6: Classification tree produced on German credit data

Finally, Table6 shows the different measurements obtain while running the full model, and both reduced model. These results shows that the full model has a better probability to predict new applicants credit rating with accuracy compared to the others.



	Accuracy	Kappa	Sensitivity	Specificty	AUC
Full Model tuned	0.6833	0.3765	0.8352	0.6172	0.758
Reduced imp Model tuned	0.7	0.4108	0.8681	0.6268	0.7537
Reduced Model tuned	0.7233	0.4318	0.8132	0.6842	0.5

Table 6: Summary of the Classification tree models precision

Further, comparing the ROC curves from the logistic full model and the classification tree full model, we see clearly the lack from the latter model. Indeed, the area under the cureve is the smallest for the decision-tree based model (0.7601 for the logistic model versus 0.6577 for the latter model).

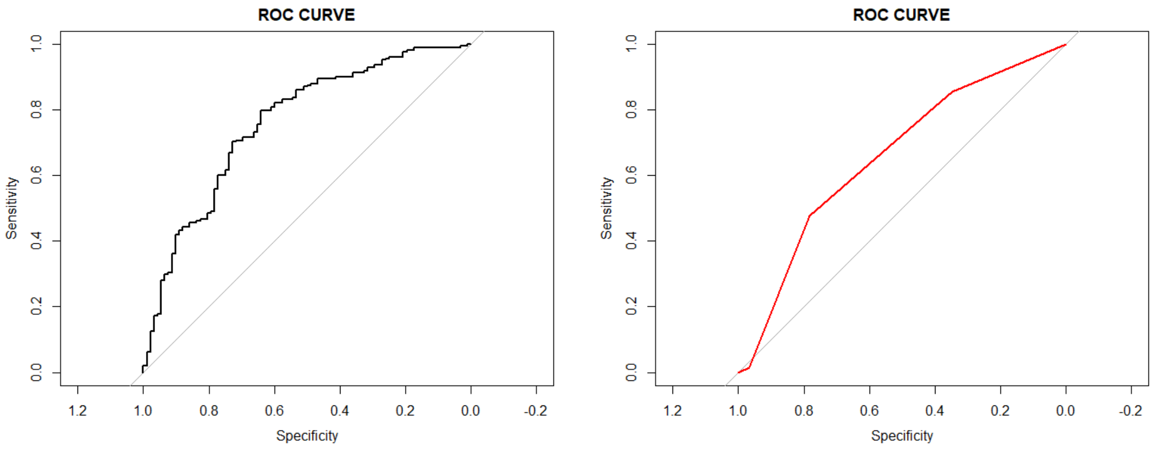


Table 7: Summary of the Classification tree models precision

3.4 Random forest

Random forest gathers a large number of decision trees each independent from another, but operating as a whole. So, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. And then, the class with the most votes becomes our model prediction. After running the appropriate fit on the training dataset for all full model and reduced models, we have come to the following confusion matrix.

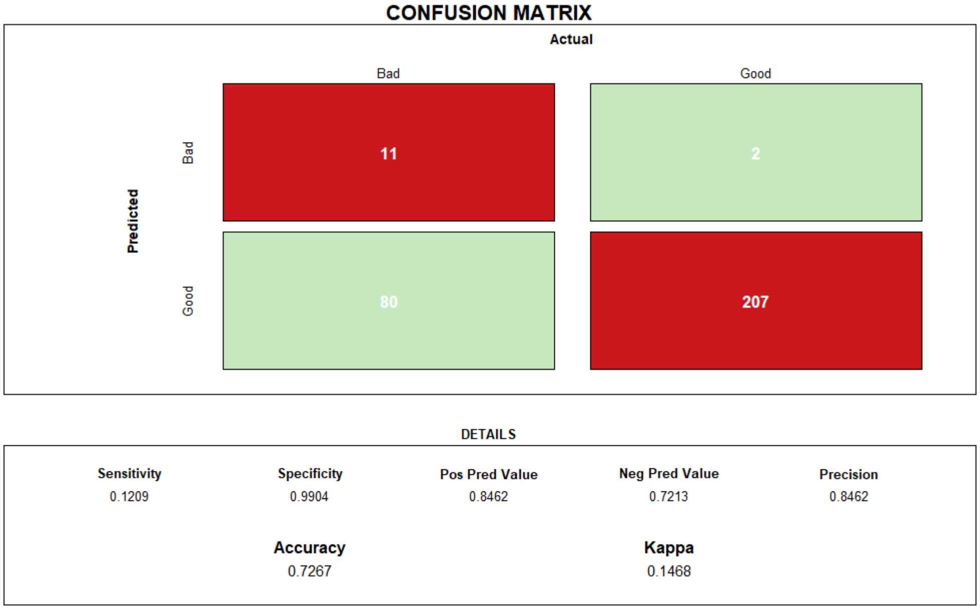
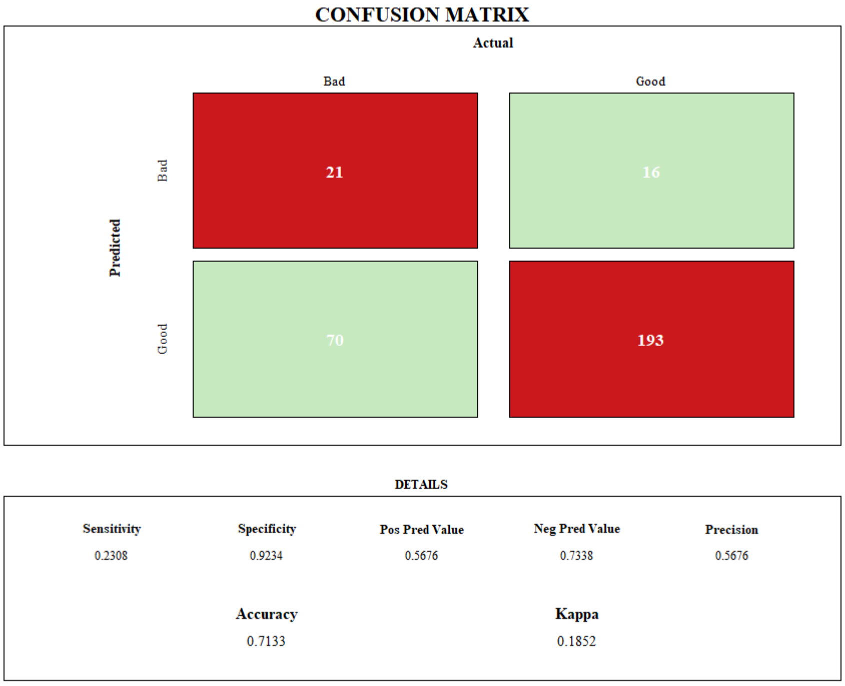


Figure 7: Confusion matrix

The rate of true negative (which is also the specificity and the recall) is significantly higher than in the classification tree model. It is of 0.9904. Meaning that this model is effective to identify the good contractors (for all our confusion matrix the Positive class is the 'Bad' category of the response variable). Although the accuracy and the Kappa, are lesser than in the classification tree, we believe that this model can show reliability in identifying new credit applicants. Further, the Kappa is very sensitive to the size of the categories from the confusion matrix. Finally the value of the area under the curve is 0.82 which is also greater than the AUC from the classification tree model, and denotes of a better accuracy of the random forest model.

3.5 k-Nearest Neighbour

As we have seen in class, k-Nearest Neighbour classification (KNN) is based on Euclidean distance computation between a set of observations. KNN can be used for both classification and regression predictive problems. However, in practice it is more used in classification problems in the industry (James et al., 2013). The idea is to class an observation based on the most represented class of its 'neighbours' around this given observation.



CONFUSION MATRIX

Predicted	Actual: Bad	Actual: Good
Bad	21	16
Good	70	193

DETAILS

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
0.2308	0.9234	0.5676	0.7338	0.5676

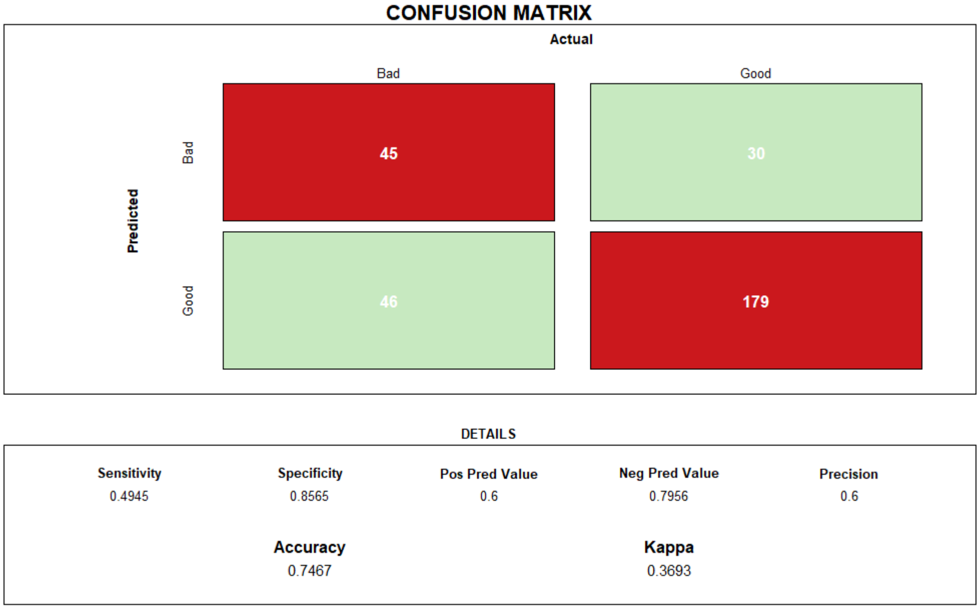
Accuracy	Kappa
0.7133	0.1852

Figure 8: Confusion matrix of the full model

The model has an accuracy of 71.33% in classifying new applicants, from which we can derive the error rate ((16+70)/300=28.67%). It denotes the overall rate of error when classifying new applicants. The precision rate indicates when the model predicts an applicant as a good credit contractor how often this classification is correct. Its computation is based on taking the true positive predicted over its total (21/(21+16)=0.6338). The specificity which is the true negative rate indicates to what extent the model predicts effectively the good contractors (true negative/reference_good = 193/263) which is 73.38%. However the model is not effective to identify the bad rating applicants, the true positive rate (the sensitivity) is at 23.08%.
Finally the Cohen's Kappa score (0.1852)is even lower than the previous model and denotes how well the bank would identify its applicants for a credit without the model knowledge. Based on this model, the lack of information to predict the profile of new borrowers will lead to a bad decision in more than 8 times out of ten.

3.6 Neural networks

Intended to simulate the behaviour of human brain, neural networks are designed to recognize patterns. They are applicable to both classification and regression problems (Zuber, 2019). Neural networks are organised in layers interconnected by nodes. Each node in a layer has a connection to each nod of the subsequent layer.On AppendixA8 we can see the importance variables selected by the model. Giving the checking accountm the installment rate, the saving accounts balance, the years of residence and the credit history as the most critical criteria to take into consideration to predict the credit rating.



CONFUSION MATRIX

		Actual	
		Bad	Good
Predicted	Bad	45	30
	Good	46	179

DETAILS

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
0.4945	0.8565	0.6	0.7956	0.6

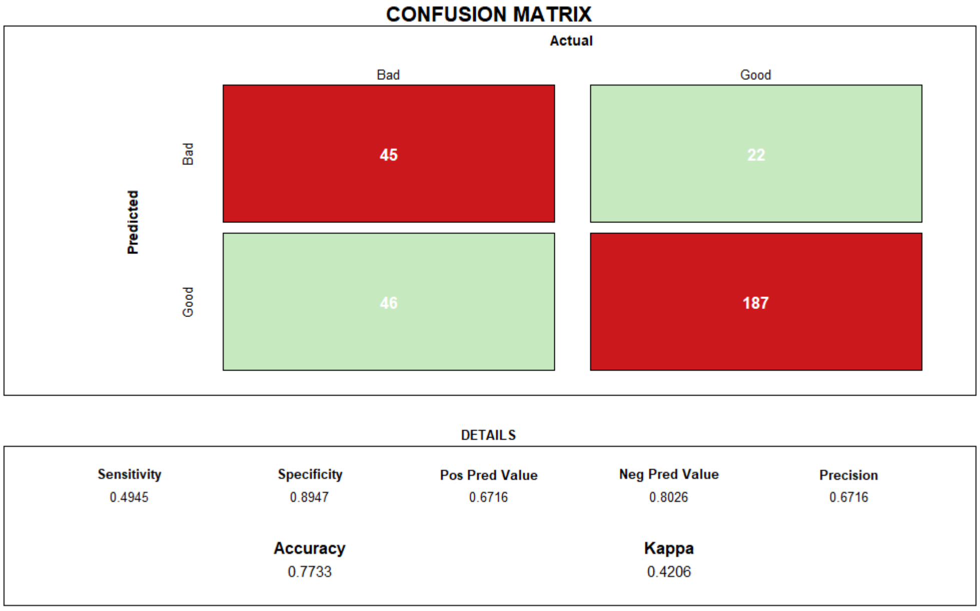
Accuracy	Kappa
0.7467	0.3693

Figure 9: Confusion matrix of the full model

Based on the confusion matrix results, this model is not much less accurate than the classification tree, nor the random forest. Further it is important to notice the sensitivity rate is quite low (49.45%) and this should be taking into account although it is only an average. Because it is the ability of the model to identify a potential bad applicant effectively (when it is indeed the case).

3.7 SVM-radial

Support vector machines are statistical models considered to be very flexible and effective machine learning tools. Unlike, random forest method SVM are supervised learning algorithm which can be used in classification models. The idea is to generate linear boundaries between the observation regarding their class. If the algorithm can find this hyperplane, it stops here. Else, the SVM will use a 'non-linear mapping to transform the training dataset'(James et al., 2013). And this non-linear transformation of the predictors is done by using the 'Kernel trick'(James et al., 2013).



CONFUSION MATRIX

		Actual	
		Bad	Good
Predicted	Bad	45	22
	Good	46	187

DETAILS

Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
0.4945	0.8947	0.6716	0.8026	0.6716

Accuracy	Kappa
0.7733	0.4206

Figure 10: Confusion matrix for the Full model

Although the AUC is the slightly the highest regarding the 5 other models (0.7795), the confusion matrix does not show definite better rates regarding the accuracy, nor the specificity or the sensitivity rates. Here again, we see that the important attributes are overall the same as the previous models (cf. AppendixA8 and A9). So indeed, the most important criteria are the checking account, the duration of the credit the balance on the saving account and the age of the male applicants.

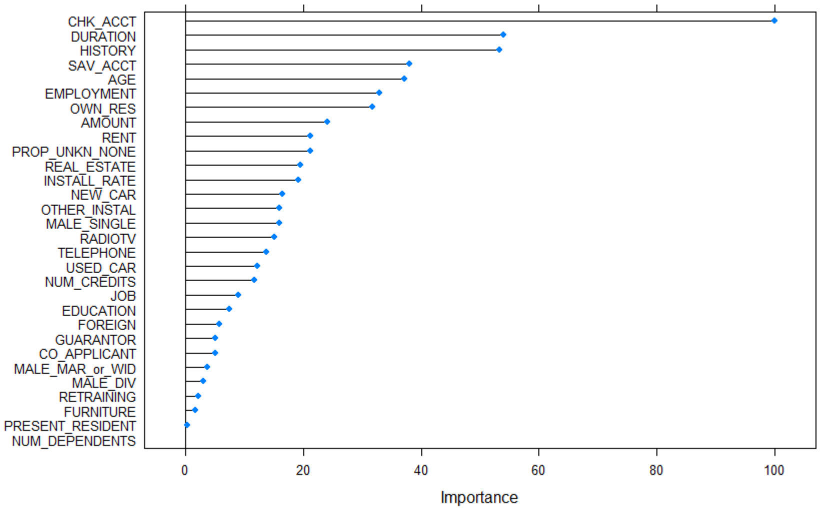


Figure 11: Important variable in the SVM-radial model

4 Discussion

Classification accuracy provides a good information about the model but it is not enough to take a decision among all models that we built. Of course the accuracy is an important metric, which evaluates how good our model is, but simple accuracy is not appropriate when we have unbalanced data(Chawla et al., 2002). Therefore, since we work with a dataset where the risk is taken into account, we need to consider other possible metrics as the sensitivity and specificity of the model aside the accuracy. Indeed, the question that we want to solve with this dataset is and we are trying to focus on the errors of the model for one specific class. Because of this, the metric ROC and AUC seemed to be the most reliable metrics for the evaluation our models. Although all the methods are rather effective and tend to give similar results(regarding the important variables for instance), the method with the best AUC value is the model SVM-radial and the classification tree (called decision tree on Figure12) is the less accurate in predicting potential borrowers.

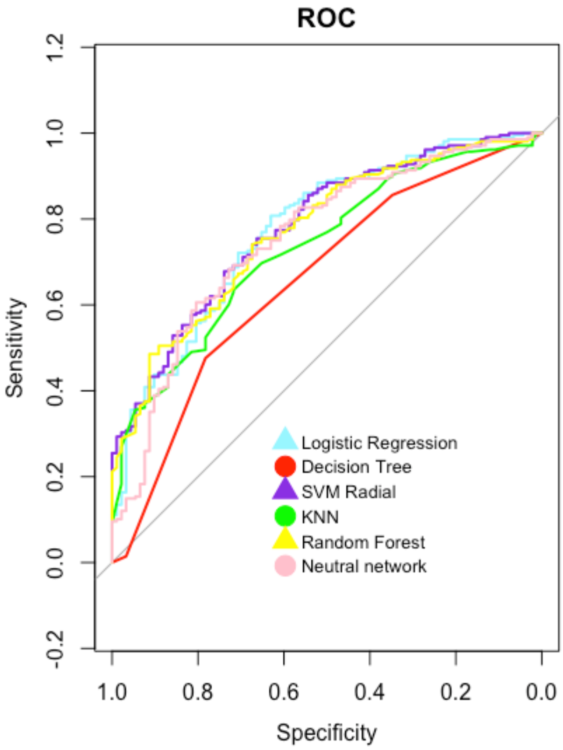


Figure 12: Models comparison

Also we would like to add that the κ-measure has been confusing. Indeed, if the κ is low is it necessarily a bad signal , in the sense that it could denote that the model is not precise enough. Rather than a lack of consensus (Cohen, 1960)between two different raters? Also, we have to note that the dataset does not contain information on female applicants. So we do not to what extent it is sufficient to have information only on one gender and to infer it to the other.

5 Conclusion

The analysis of the German credit dataset allowed us to understand the importance for a bank to have a model to base its decision on. Particularly when decision making can lead to profit loss. This is the case when attributing consumer credits. Going through this dataset, we have seen that the theoretical important criteria (Myers et al., 1991) meets with the empirical criteria, such as the checking and saving account amount of applicants, their credit history at this bank, the amount and the duration of the credit, and the age. These attributes where the most chosen amongst important variable by the different predictive models fitted. Regarding, the gender/sex attribute we have denoted that the dataset provided information only on male applicants, though it would have been interesting to analyse the credit risk for both gender. Furthermore the analysis of the German credit data, we have been able to applied few amongst many data mining methods in order to predict credit rating for potential borrowers. The SVM-radial method as shown the most reliable results for the bank to based its decision on. What we can conclude, is that data mining when used appropriately is an essential step in knowledge discovery involving both theories and statistical tools for revealing patterns in data. Because we have seen the importance of exploiting different sources of attributes (from demographic to socio-economic profiles), althought it seems that internal information (accounts balances, credit history) were more reliable.

Bibliography

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

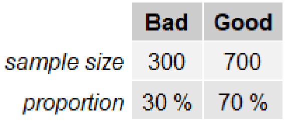
Myers, S. C., Allen, F., et al. (1991). Principles of corporate finance.

Zuber, J. (2019). Lecture notes in the seminar of applied statistics.

A List of appendices

A.1 Proportions of the credit rating

Table A1: Proportions of the credit rating



	Bad	Good
sample size	300	700
proportion	30 %	70 %

A.2 Repartition of male applicants and civil status

Table A2: Original repartition of applicants gender and civil status

```
# A91 : male   : divorced/separated
# A92 : female : divorced/separated/married
# A93 : male   : single
# A94 : male   : married/widowed
|V9  | count|proportion |
|:---|-----:|:----------|
|A91 |    50|5 %        |
|A92 |   310|31 %       |
|A93 |   548|54.8 %     |
|A94 |    92|9.2 %      |
```

A.3 Histograms of the continuous variables and correlation matrices

Figure A2: Histograms of the continuous variables

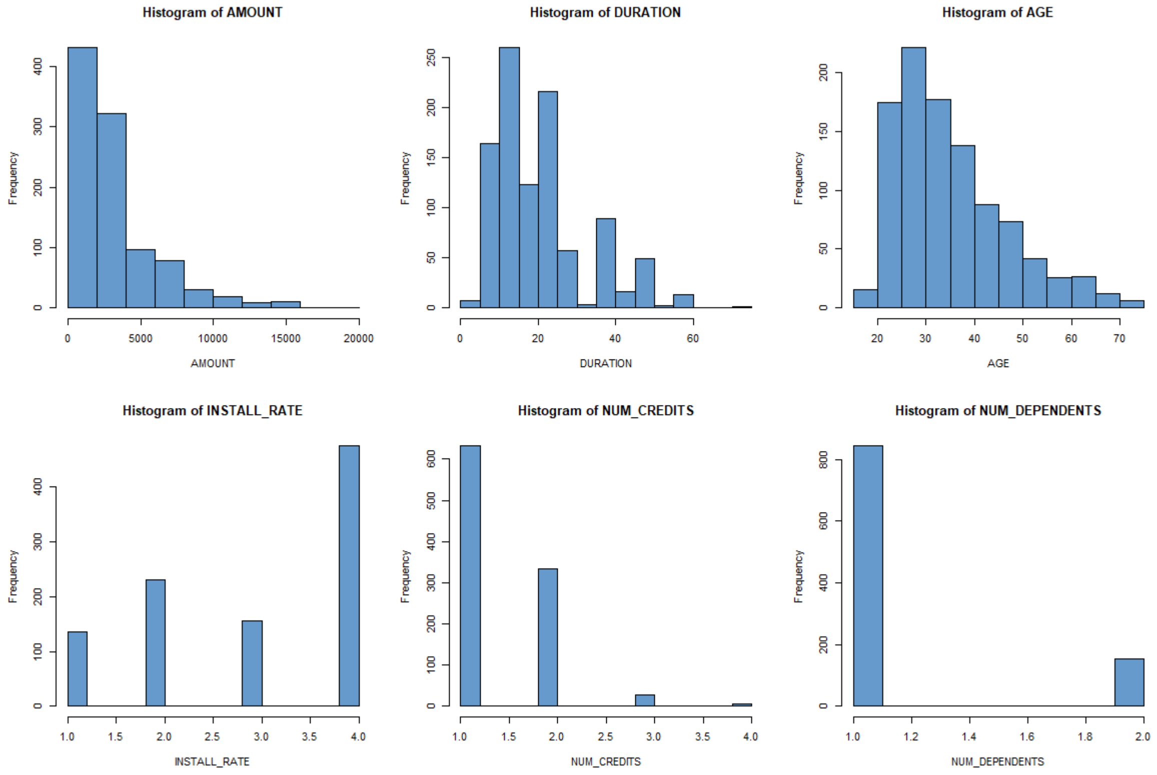
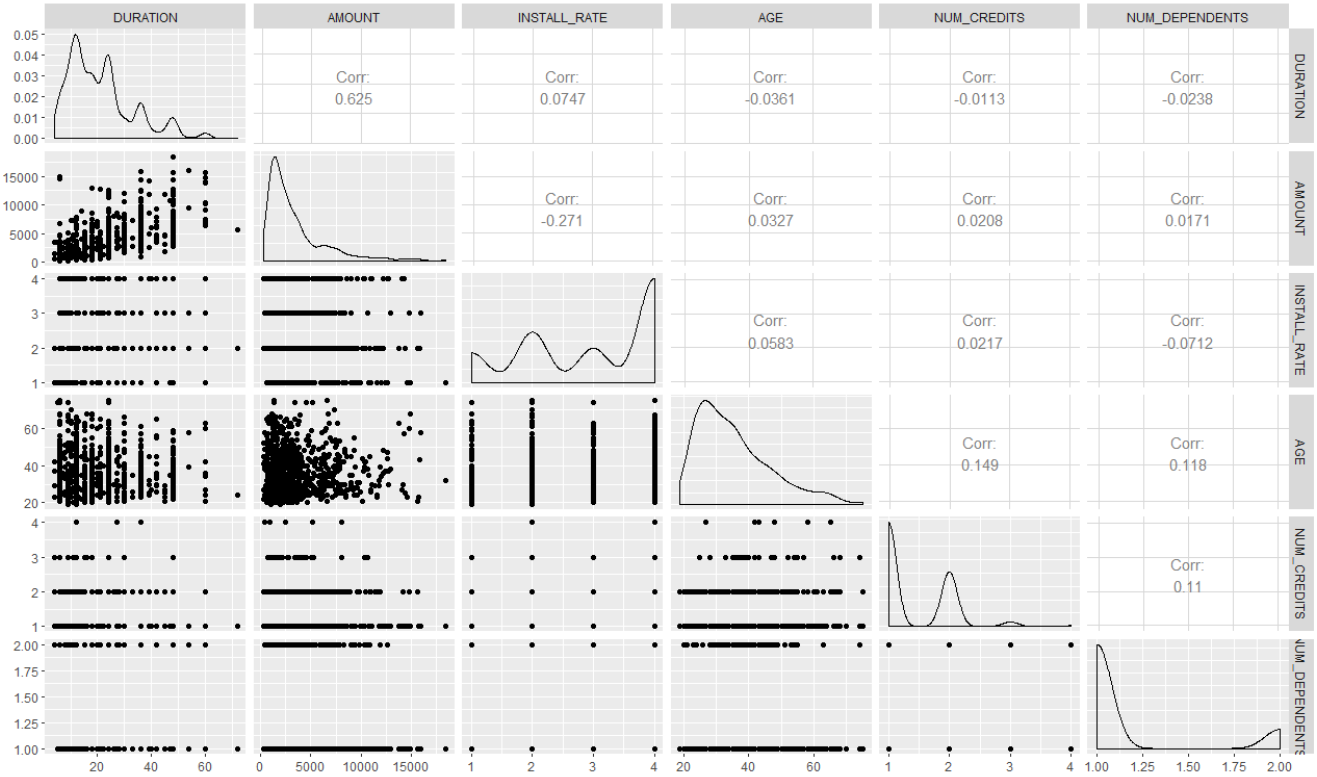


Table A3: Correlation matrix

	DURATION	AMOUNT	INSTALL_RATE	AGE	NUM_CREDITS	NUM_DEPENDENTS
DURATION	1.000	0.625	0.075	-0.036	-0.011	-0.024
AMOUNT	0.625	1.000	-0.271	0.033	0.021	0.017
INSTALL_RATE	0.075	-0.271	1.000	0.058	0.022	-0.071
AGE	-0.036	0.033	0.058	1.000	0.149	0.118
NUM_CREDITS	-0.011	0.021	0.022	0.149	1.000	0.110
NUM_DEPENDENTS	-0.024	0.017	-0.071	0.118	0.110	1.000

Figure A3: Scatterplot matrix

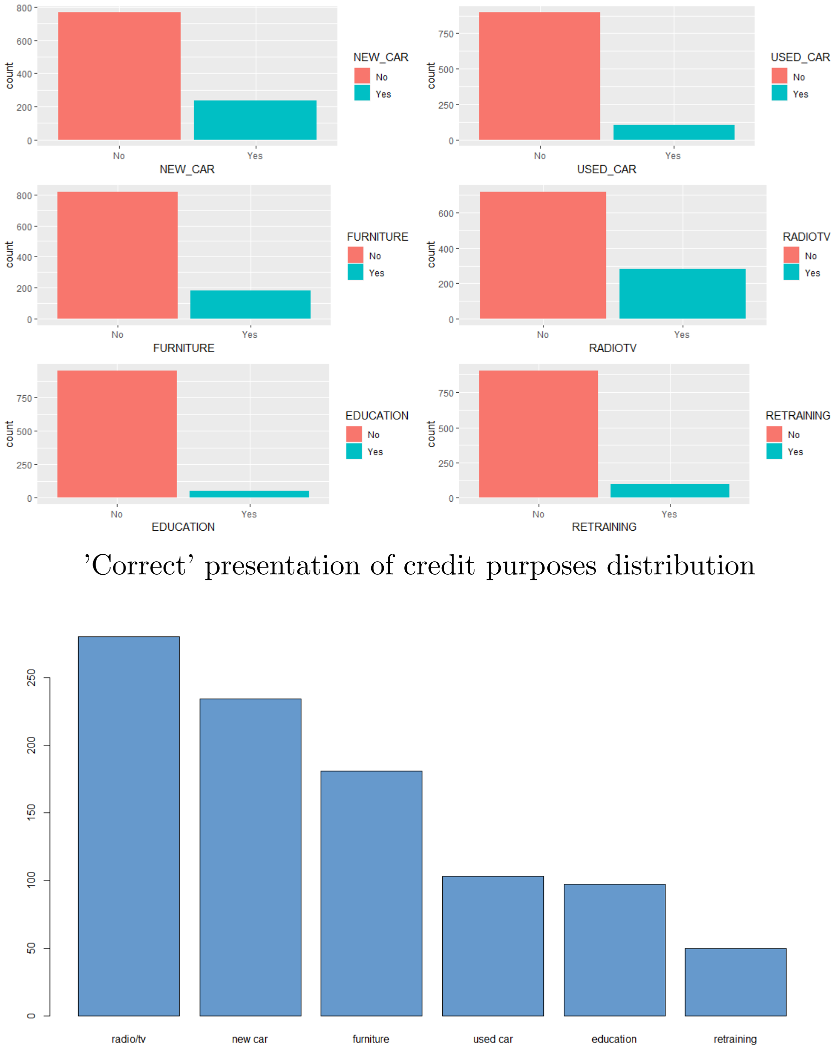


A.4 Types of credit

Table A4: Original repartition of credit purposes

```
> table(V4)
V4
A40  A41 A410  A42  A43  A44  A45  A46  A48  A49
234  103   12  181  280   12   22   50    9   97
# A40 : car (new)
# A41 : car (used)
# A42 : furniture/equipment
# A43 : radio/television
# A44 : domestic appliances
# A45 : repairs
# A46 : education
# A47 : (vacation - does not exist?)
# A48 : retraining
# A49 : business
# A410 : others
```

Figure A4: Repartition of credit purposes

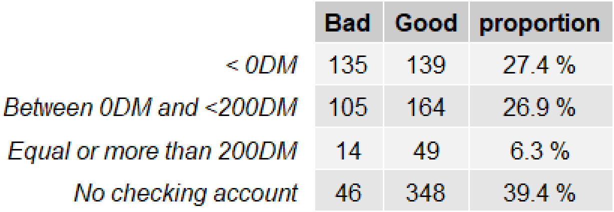


'Misleading' presentation of credit purposes distribution

'Correct' presentation of credit purposes distribution

A.5 Proportion of the credit rating among checking account status

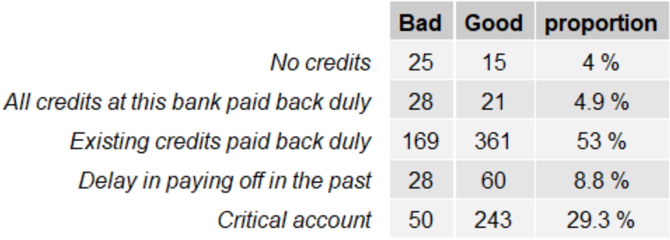
Figure A5: Proportion of the credit rating among checking account status



	Bad	Good	proportion
< 0DM	135	139	27.4 %
Between 0DM and <200DM	105	164	26.9 %
Equal or more than 200DM	14	49	6.3 %
No checking account	46	348	39.4 %

A.6 Payment history repartition grouped by credit ratings

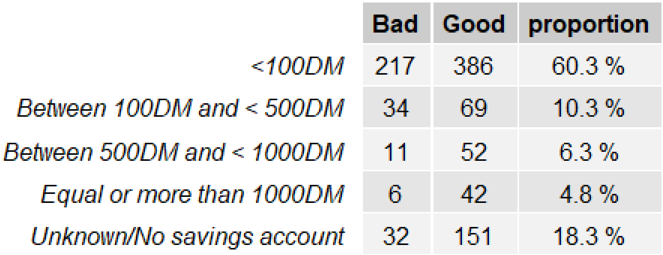
Figure A6: Payment history repartition



	Bad	Good	proportion
No credits	25	15	4 %
All credits at this bank paid back duly	28	21	4.9 %
Existing credits paid back duly	169	361	53 %
Delay in paying off in the past	28	60	8.8 %
Critical account	50	243	29.3 %

A.7 Repartition of average balance in savings account with respect to credit ratings

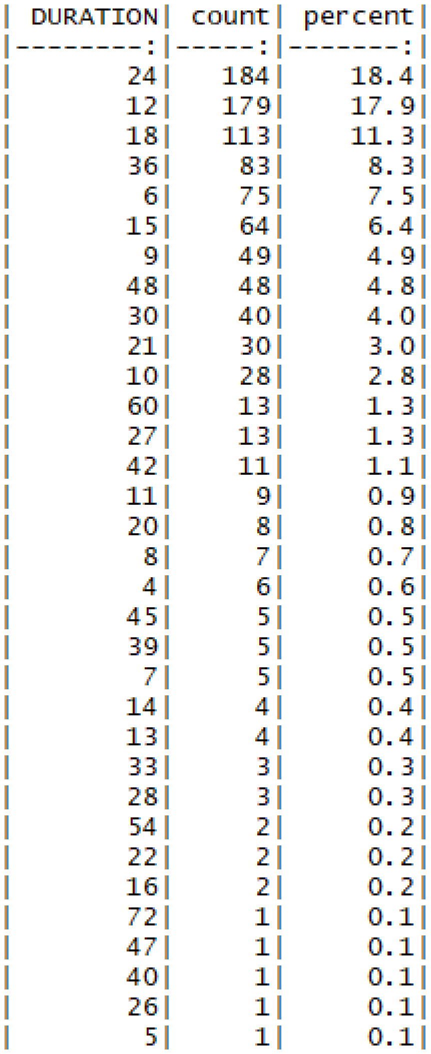
Figure A7: Repartition of average balance in savings account with respect to credit ratings



	Bad	Good	proportion
<100DM	217	386	60.3 %
Between 100DM and < 500DM	34	69	10.3 %
Between 500DM and < 1000DM	11	52	6.3 %
Equal or more than 1000DM	6	42	4.8 %
Unknown/No savings account	32	151	18.3 %

A.8 Frequency of monthly credit duration

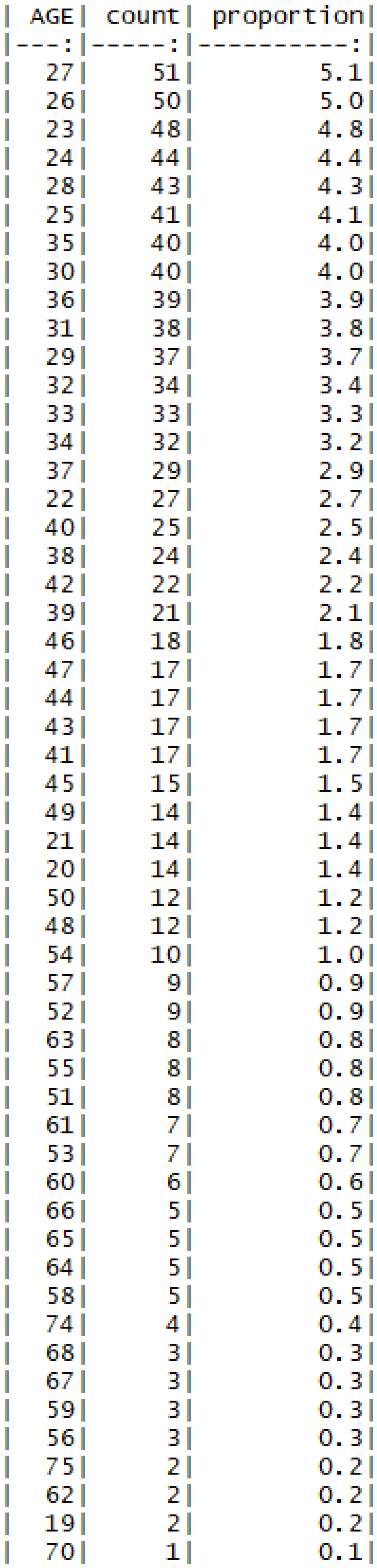
Table A5: Repartition of monthly credit duration



DURATION	count	percent
24	184	18.4
12	179	17.9
18	113	11.3
36	83	8.3
6	75	7.5
15	64	6.4
9	49	4.9
48	48	4.8
30	40	4.0
21	30	3.0
10	28	2.8
60	13	1.3
27	13	1.3
42	11	1.1
11	9	0.9
20	8	0.8
8	7	0.7
4	6	0.6
45	5	0.5
39	5	0.5
7	5	0.5
14	4	0.4
13	4	0.4
33	3	0.3
28	3	0.3
54	2	0.2
22	2	0.2
16	2	0.2
72	1	0.1
47	1	0.1
40	1	0.1
26	1	0.1
5	1	0.1

A.9 Distribution of applicants' age

Table A6: Distribution of applicants' age



AGE	count	proportion
27	51	5.1
26	50	5.0
23	48	4.8
24	44	4.4
28	43	4.3
25	41	4.1
35	40	4.0
30	40	4.0
36	39	3.9
31	38	3.8
29	37	3.7
32	34	3.4
33	33	3.3
34	32	3.2
37	29	2.9
22	27	2.7
40	25	2.5
38	24	2.4
42	22	2.2
39	21	2.1
46	18	1.8
47	17	1.7
44	17	1.7
43	17	1.7
41	17	1.7
45	15	1.5
49	14	1.4
21	14	1.4
20	14	1.4
50	12	1.2
48	12	1.2
54	10	1.0
57	9	0.9
52	9	0.9
63	8	0.8
55	8	0.8
51	8	0.8
61	7	0.7
53	7	0.7
60	6	0.6
66	5	0.5
65	5	0.5
64	5	0.5
58	5	0.5
74	4	0.4
68	3	0.3
67	3	0.3
59	3	0.3
56	3	0.3
75	2	0.2
62	2	0.2
19	2	0.2
70	1	0.1

A.10 Important variables with respect to the model fitted

Table A7: Important variable in the logistic regression model

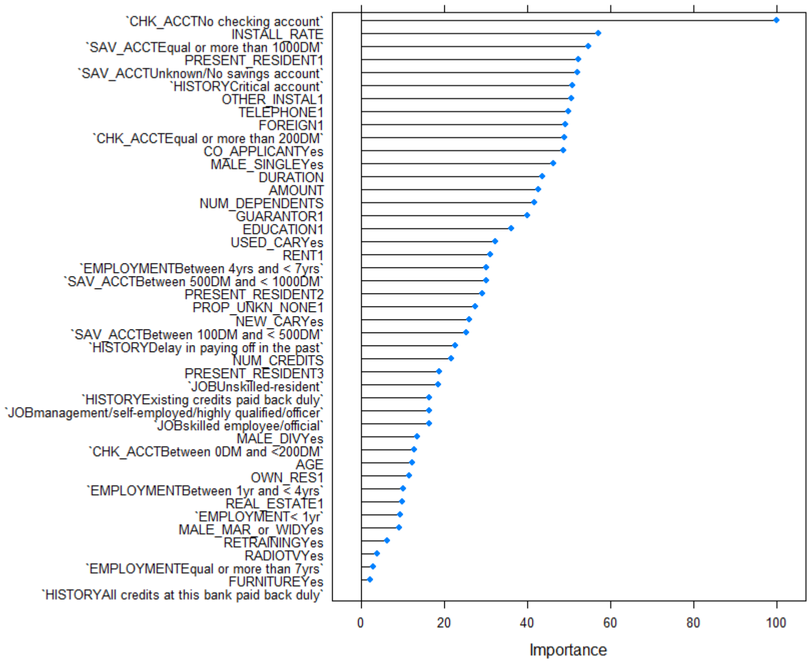


Table A8: Important variable in the classification tree full model

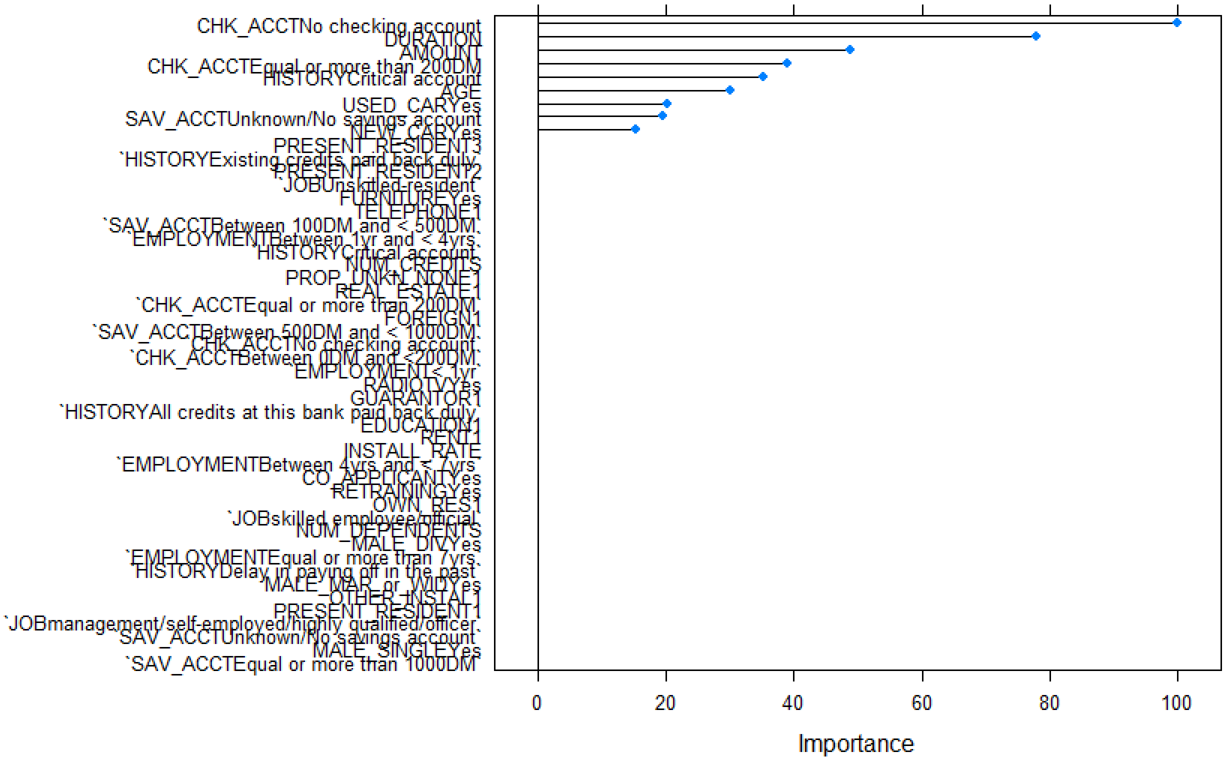
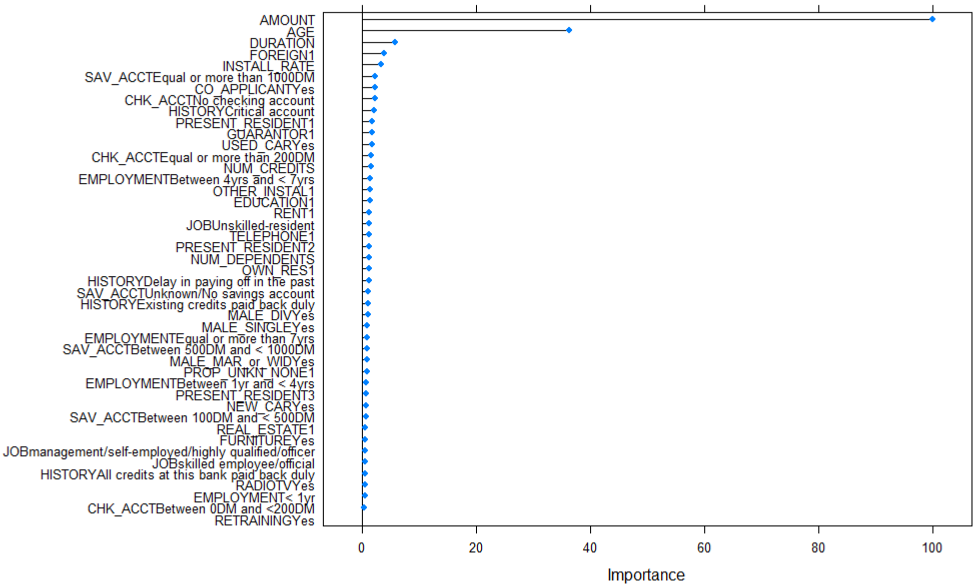


Table A9: Important variable in the Neural network model



A.11 Classification tree: Complexity table

Table A10: Complexity table

```
Classification tree:
rpart(formula = RESPONSE ~ ., data = ., method = "class", model = T,
    cp = 0.001)

Variables actually used in tree construction:
 [1] AMOUNT           CHK_ACCT         DURATION         EMPLOYMENT
 [5] FURNITURE        HISTORY          INSTALL_RATE     JOB
 [9] OTHER_INSTAL     PRESENT_RESIDENT REAL_ESTATE      RETRAINING
[13] SAV_ACCT         TELEPHONE        USED_CAR

Root node error: 300/1000 = 0.3

n= 1000
        CP nsplit rel error xerror   xstd
1  0.05167      0     1.000  1.000 0.0483
2  0.04667      3     0.840  1.010 0.0484
3  0.01833      4     0.793  0.880 0.0465
4  0.01667      6     0.757  0.877 0.0464
5  0.01444      7     0.740  0.883 0.0465
6  0.01333     10     0.697  0.870 0.0463
7  0.01000     11     0.683  0.877 0.0464
8  0.00833     13     0.663  0.880 0.0465
9  0.00667     19     0.607  0.893 0.0467
10 0.00556     25     0.567  0.920 0.0471
11 0.00500     28     0.550  0.927 0.0472
12 0.00333     30     0.540  0.930 0.0473
13 0.00100     33     0.530  0.943 0.0475
```

B Additional R computation

B.1 Tables of age,duration and installment rate

```
#additional tables for age, duration, installment rate
credit %>%
  group_by(DURATION) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100
  ) %>%
  arrange(desc(count), desc(DURATION)) %>%
  head(10) %>%
  kable

credit %>%
  group_by(AGE) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100,
    proportion = paste(proportion,"%")
  ) %>%
  arrange(desc(count), desc(AGE)) %>%
  head(10) %>%
  kable
credit %>%
  group_by(INSTALL_RATE) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100,
    proportion = paste(proportion,"%")
  ) %>%
  arrange(desc(count), desc(INSTALL_RATE)) %>%
  head(10) %>%
  kable
credit %>%
  group_by(NUM_CREDITS) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100,
    proportion = paste(proportion,"%")
  ) %>%
  arrange(desc(count), desc(NUM_CREDITS)) %>%
  head(10) %>%
  kable

credit %>%
  group_by(AMOUNT) %>%
  summarize(
    count = n(),
    proportion = count / nrow(.) * 100,
    proportion = paste(proportion,"%")
  ) %>%
  arrange(desc(AMOUNT)) %>%
  #head(10) %>%
  kable
```

B.2 Tables of credit purposes

```
table(ti)
ti=credit$NEW_CAR
ti=ti[ti=="Yes"]
ti=droplevels(ti)
tj=credit$USED_CAR
tj=tj[tj=="Yes"]
tj=droplevels(tj)

tb=credit$FURNITURE
tb=tb[tb=="Yes"]
tb=droplevels(tb)

tc=credit$RADIOTV
tc=tc[tc=="Yes"]
tc=droplevels(tc)

tf=credit$RETRAINING
tf=tf[tf=="Yes"]
tf=droplevels(tf)

tq=credit$EDUCATION
tq=tq[tq==1]
tq=droplevels(tq)

t=cbind(table(tc), table(ti), table(tb),
        table(tj), table(tf),table(tq))
names(t)=c("radio/tv", "new car","furniture", "used car", "education", "retraining")
pal <- colorRampPalette(colors = c("lightblue", "blue"))(6)
barplot(t, col=lblue,names.arg =names(t))
```

B.3 Code for the confusion matrices

```
#From stackoverflow forum
cm=confusionMatrix(lr.pred, data.test$RESPONSE )
draw_confusion_matrix <- function(cm) {

  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  rect(150, 430, 240, 370, col='#CB181D')
  text(195, 435, 'Bad', cex=1.2)
  rect(250, 430, 340, 370, col='#C7E9C0')
  text(295, 435, 'Good', cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='#C7E9C0')
  rect(250, 305, 340, 365, col='#CB181D')
  text(140, 400, 'Bad', cex=1.2, srt=90)
  text(140, 335, 'Good', cex=1.2, srt=90)

  # add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 4), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 4), cex=1.2)
  text(50, 85, names(cm$byClass[3]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[3]), 4), cex=1.2)
  text(70, 85, names(cm$byClass[4]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[4]), 4), cex=1.2)
  text(90, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[5]), 4), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 4), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 4), cex=1.4)
}
#log model
draw_confusion_matrix(cm.lr)
roc.lr <- roc(data.test$RESPONSE,lr.probs[,"Good"])
#plot the ROC curve
plot(roc.lr,col=c(1),main= "ROC CURVE")
auc(roc.lr)
#log reduced model cm
draw_confusion_matrix(cm.red.lr)

#classification tree cm
draw_confusion_matrix(cm.tree.red.grid)

#SVM
draw_confusion_matrix(cm.svm)


#random forest
draw_confusion_matrix(cm.rf)
```

C Figures and tables

List of Figures

List of Tables