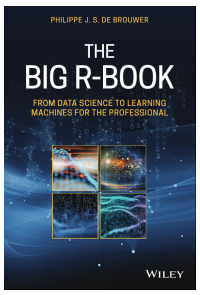

MRM Academy

Project for the course Data Science

The Selection of a Model for German Credit Applications

Philippe De Brouwer

26 March, 2021

Abstract

We use the data of German Credit applications to build a credit model. We use two approaches: neural networks and logistic regression and conclude that the neural network is not preferable, because it over-fits too much and is black box. For the logistic regression we also build a challenger model that has one variable more and conclude that this model is the best so far.

Contents

1 Introduction

This is a template that can be used for the report of the project that goes with the course *Data Science* of **Philippe De Brouwer** for AGH. It is obviously not perfect, nor really finished. We invite you to use this as a starting point, and rethink all decisions yourself. Eventually we hope that you can build a better model.

Best is to take the models that is selected here ("logistic 2") as a reference and take it as a challenge to build a better model. The quality of the model is both the performance and the robustness ("not over-fit"). Ideally it is better in at least one of those two areas while at least as good in the other criterion.

You are free to take another view anything. In particular we suggest you to consider:

1. the selection of variables,
2. interaction variables (matrix variables),
3. the choice of model (logistic, probit, decission tree, neural network, SVM, etc.), or eventually improve the models chosen (e.g. how could you make the neural network to be less over-fitted?),
4. of course you can also choose to consider other performance criteria and even use other cross validation methods, such as *k-fold cross validation*, and
5. the documentation, are the better ways of presenting it, what to present, etc.

Good luck!

The data is believed to be public domain.

The main reference is De Brouwer (2020), and more in particular (De Brouwer 2020, pt. 1 to 6 with elements of part 8), the slides presented in the classroom, and the code that we walked through together.

2 The Data

2.1 Introduction

The data consists of 1000 rows and 22 columns. In Table 1 we list all column names.

Table 1: The column names of the data.

Column Names
default
account_check_status
duration_in_month
credit_history
purpose
credit_amount
savings
present_emp_since
installment_as_income_perc
personal_status_sex
other_debtors
present_res_since
property
age

Column Names
other_installment_plans
housing
credits_this_bank
job
people_under_maintenance
telephone
foreign_worker
isGood

The breakdown of the dependent variable "default'' is as presented in Table 2.

Table 2: The Data consists of a high number of defaults. We assume that this is because the data has been prepared by dropping randomly "good" customers.

Var1	Freq
0	700
1	300

2.2 The Data Quality

From the 1000 observations, there are 0 rows with missing values. Some of the values have one of the categorical variables that indicates the lack of information. We refer to the following sections where we will show how these "NA" values will be treated.

No suspicious outliers were found.

3 Data Wrangling

3.1 Categorical Variables

3.1.1 The information Value

The first step is selecting the variables that we ultimately want to use in our model. First we will consider the categorical variables and study the information value.

Table 3: The table of all information values for each categorical variable ordered in decreasing order. We will work with the ones that have an information value above 0.1.

	varName	IV
1	account_check_status	0.6660115
2	credit_history	0.2932335
4	savings	0.1960096
3	purpose	0.1691951
8	property	0.1126383

	varName	IV
5	present_emp_since	0.0864336
10	housing	0.0832934
9	other_installment_plans	0.0576145
6	personal_status_sex	0.0446707
13	foreign_worker	0.0438774
7	other_debtors	0.0320193
11	job	0.0087628
12	telephone	0.0063776

3.2 The Continous Variables

The continuous variables are presented in Table 4. Also their content follows from the naming convention.

Table 4: List of the continuous scale variables in the data.

Continuous Scale Variables
duration_in_month
credit_amount
installment_as_income_perc
present_res_since
age
credits_this_bank
people_under_maintenance

3.2.1 Decide which Continuous Variable to Use

Note that we have introduced a variable *isGood* which is one for good credits and zero for bad credits.

In order to select continuous scale variables we will investigate the correlation and study the plot of the average behaviour in function of the relevant varialbes in order to detect non-linear dependencies. Later (in next section) we will consider interactions between the different variables (relevant for the logistic regression for example).

Based on the correlation we can certainly consider *duration_in_month*, *credit_amount*, *installment_as_income_perc* (which is usually a strong candidate, however in this data the dependency is not remarkable: -0.0724039), and *age*.

We also note the high correlation between *installment_as_income_perc*, *duration_in_month* and *credit_amount* and make note that we might need to combined them in order to avoid co-linearity if we would decide to keep both variables in a regression model.

We used a loess model and histogram to assess what binning is optimal for these variables. Here we present the loess with superimposed confidence level for the variables where the correlation with *isGood* is above 0.05%.

We already knew that the variable *installment_as_income_perc* had a reasonably low correlation with the dependent varialbe. From Figure 2 we learn also that the shape of the relationships is rather flat. Therefore we decide in a first approach not to use this variable (the models “logistic 1”

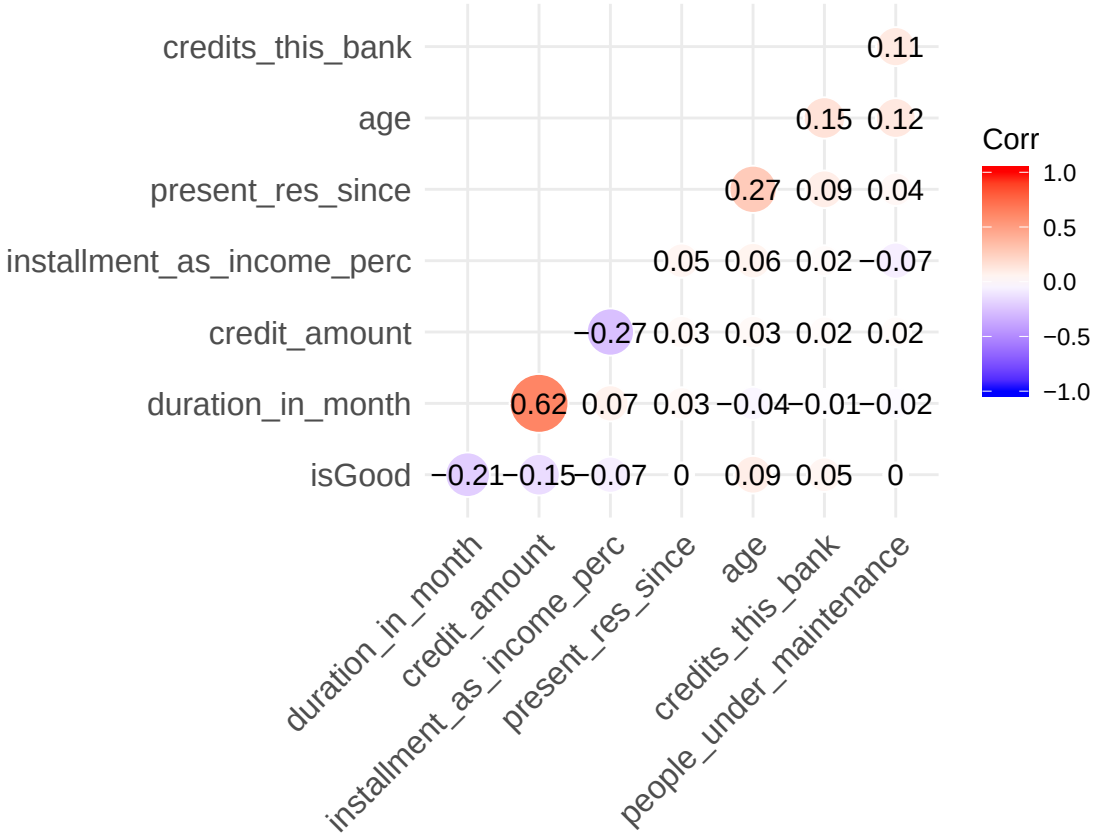

Figure 1: The correlation matrix for all continous scale vriables. Note that *isGood* is the binary variable that equals 1 for all successful credits and 0 for credits that defaulted.

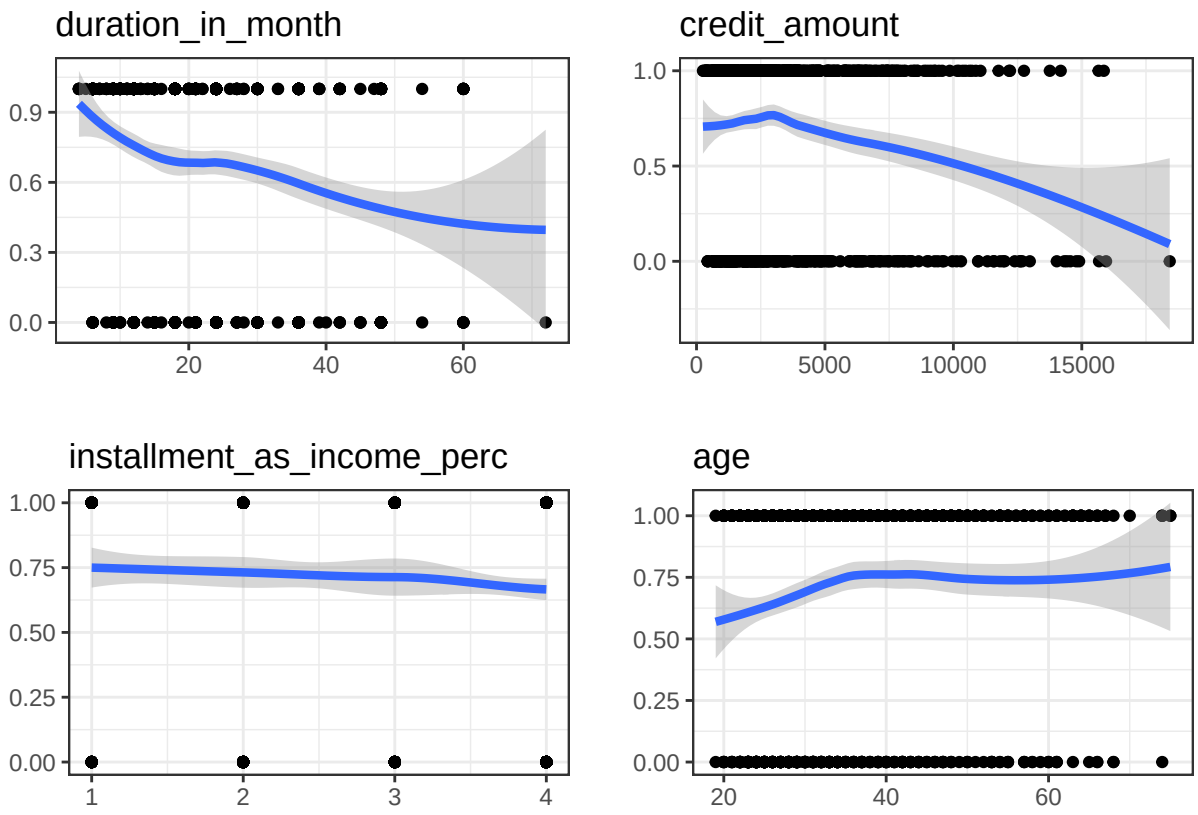

Figure 2: The loess estimations and its confidence level for the variables with correlation above.

and "neuralnet" will be built without this parameter) and build a challenger models that includes this variable ("logistic 2").

3.3 Data Binning

3.3.1 The Categorical Variables

The existing binning is probably directly taken from the production system. This means that we should at least check if it is optimal for a logistic regression and its cross validation. Ideally we want to avoid bins that are too small and we also want to make sure that the relationships observed are realistic, logical and consistent.

We decided to make sure all bins have at least 200 observations in them and will take bins that have similar WOE together.

3.3.1.1 Account_check_status To illustrate the thought process, we display the information for *account_check_status*. First we study the WOE of the existing bins (see Figure 3) and their detailed information about histogram, weight of evidence and information value (see Table 5).

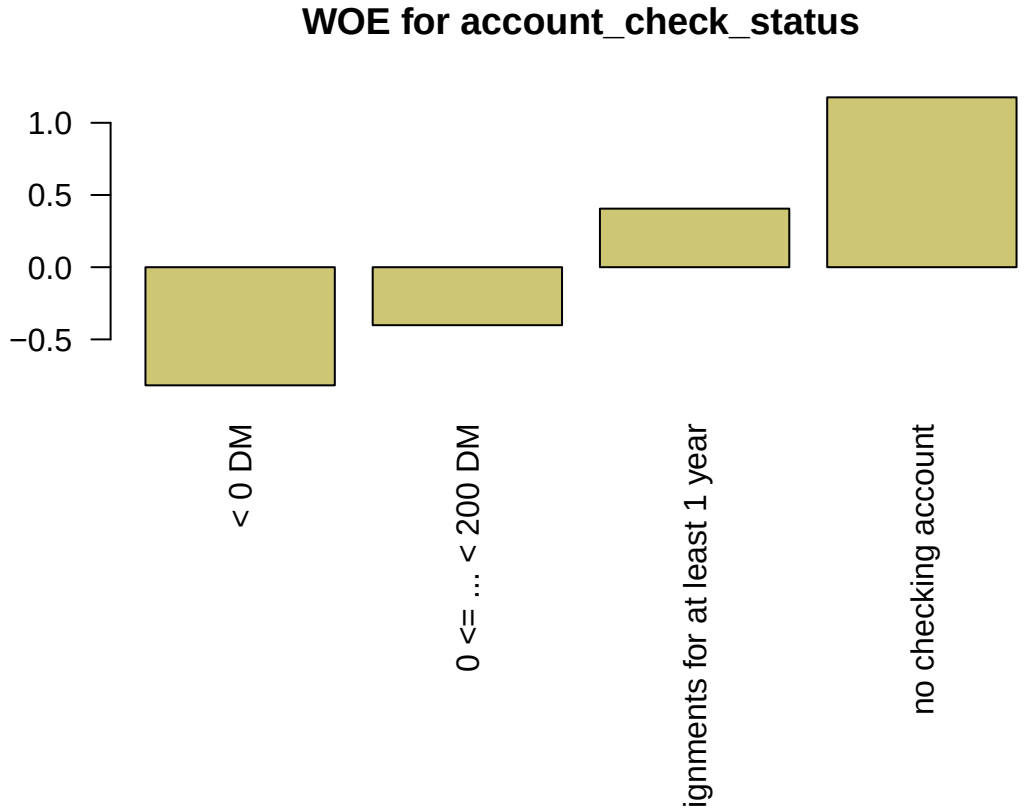

Figure 3: The WOE for the variable *account_check_status*.

Table 5: The WOE and IV for the variable account_check_status.

account_check_status	n	nbrBad	nbrGood	pctGood	WOE	IV
< 0 DM	274	135	139	0.51	-0.82	0.21
0 <= ... < 200 DM	269	105	164	0.61	-0.40	0.05
>= 200 DM / salary assignments for	63	14	49	0.78	0.41	0.01

account_check_status	n	nbrBad	nbrGood	pctGood	WOE	IV
no checking account	394	46	348	0.88	1.18	0.40

Then we decide on which bins to combine. In this case, we decide to take the second and third bin together. This is done by the following code:

```
df$acc_chk_sts   <- if_else(d$account_check_status == '< 0 DM', 'neg',
  if_else((d$account_check_status == '0 <= ... < 200 DM') |
  (d$account_check_status == '>= 200 DM / salary assignments for at least 1 year'),
  'pos',
  if_else(d$account_check_status == 'no checking account', 'none', 'ERROR')))

# Check for mistakes (they should appear as ERROR or an undefined category):
df$acc_chk_sts %>% table
```

```
## .
##  neg none  pos
##  274  394  332
```

3.3.1.2 Other variables We work with similar logic through all other variables and make the following transformations:

```
### credit_history
df$credit_hist <- if_else((d$credit_history == 'no credits taken/ all credits paid back du
                           (d$credit_history == 'all credits at this bank paid back duly'
                          'clear',
                  if_else((d$credit_history == 'existing credits paid back duly till now')
                           (d$credit_history == 'delay in paying off in the past'), 'paid
                  if_else(d$credit_history == 'critical account/ other credits existing (n
                          'crit', 'ERROR')))

### savings
df$savings <- if_else((d$savings == '... < 100 DM') | (d$savings == '100 <= ... < 500 DM')
              if_else((d$savings == 'unknown/ no savings account') | (d$savings == '500 <=
              if_else(d$savings == '.. >= 1000 DM ', 'large', 'ERROR')))

### Propoerty
df$property <- if_else(d$property == 'unknown / no property', 'none',
               if_else((d$property == 'if not A121/A122 : car or other, not in attribute 6
                  (d$property == 'if not A121 : building society savings agreement/ life i
                  'movbl',
               if_else(d$property == 'real estate', 'real', 'ERROR')))

### purpose
df$purpose <- if_else((d$purpose == '(vacation - does not exist?)') |
                           (d$purpose == 'furniture/equipment') |
                           (d$purpose == 'car (new)')
                         , 'friv',
              if_else((d$purpose == 'education') |
```

```
                                (d$purpose == 'business') |
                                (d$purpose == 'repairs') |
                                (d$purpose == 'radio/television')
                                  , 'invest',
                    if_else((d$purpose == 'domestic appliances') |
                                (d$purpose == 'car (used)') |
                                (d$purpose == 'retraining')
                                , 'useful', 'ERROR')))
```

Finally we check the information value of the new binning:

variable	InformationValue	Predictability
acc_chk_sts	0.6345968	Highly Predictive
credit_hist	0.2918291	Highly Predictive
savings	0.1944263	Highly Predictive
property	0.1126339	Highly Predictive
purpose	0.1524439	Highly Predictive

3.3.2 The Continuous variables

Based on a similar reasoning[1], but also making sure that we capture any non-linear behaviour, we split the continuous variables as follows:

```
#### duration_in_month
df$duration <- if_else(d$duration_in_month <= 12, 'L',
               if_else(d$duration_in_month <= 24 & d$duration_in_month > 12, 'M',
               if_else(d$duration_in_month > 24, 'H', 'ERROR')))

### credit_amount
df$credit_amount <- if_else(d$credit_amount <= 2500, 'L',
                    if_else(d$credit_amount <= 5000 & d$credit_amount > 2500, 'M',
                    if_else(d$credit_amount > 5000, 'H', 'ERROR')))

### age
df$age <- if_else(d$age <= 30, 'L',
          if_else(d$age > 30, 'H', 'ERROR'))
```

4 The Logisic Regresssion

The first model that we will study is a logistic regression. We name it for furture reference "logistic 1."

```
# Define the formula:
frm <- isGood ~ acc_chk_sts + credit_hist + savings + property +
                purpose + duration + credit_amount + age +
                savings * age + purpose * credit_amount
```

[1] We want to end up with bins that are not too small and we want behaviour in that one bin to be as similar as possible.

```r
# Fit the model:
m <- glm(formula = frm, data = df, family = "binomial")

# Investigate the model:
summary(m)
```

```
##
## Call:
## glm(formula = frm, family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8710  -0.8178   0.4417   0.7537   1.8456
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.06353    0.71752  -1.482  0.13828
## acc_chk_stsnone               1.60271    0.21354   7.505 6.13e-14 ***
## acc_chk_stspos                0.48746    0.18411   2.648  0.00811 **
## credit_histcrit               1.59916    0.30318   5.275 1.33e-07 ***
## credit_histpaid               0.85061    0.26677   3.189  0.00143 **
## savingsother                  0.13769    0.62905   0.219  0.82674
## savingssmall                 -0.67843    0.58460  -1.161  0.24585
## propertynone                 -0.43598    0.23092  -1.888  0.05902 .
## propertyreal                  0.26545    0.19686   1.348  0.17751
## purposeinvest                 0.16447    0.44126   0.373  0.70935
## purposeuseful                 0.56367    0.40853   1.380  0.16767
## durationL                     1.18307    0.26444   4.474 7.68e-06 ***
## durationM                     0.60467    0.22422   2.697  0.00700 **
## credit_amountL               -0.36478    0.37708  -0.967  0.33335
## credit_amountM                0.01628    0.44834   0.036  0.97103
## ageL                          0.31462    0.97975   0.321  0.74811
## savingsother:ageL            -0.89994    1.05378  -0.854  0.39310
## savingssmall:ageL            -0.73550    0.99716  -0.738  0.46076
## purposeinvest:credit_amountL  0.39365    0.51650   0.762  0.44597
## purposeuseful:credit_amountL  0.44690    0.48837   0.915  0.36015
## purposeinvest:credit_amountM  0.50853    0.58928   0.863  0.38816
## purposeuseful:credit_amountM  0.65443    0.59472   1.100  0.27116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  969.29  on 978  degrees of freedom
## AIC: 1013.3
##
## Number of Fisher Scoring iterations: 5
```

We notice that none of the interactions is statistically significant and decide to leave them out in

order to make the model more robust. Then we fit the model agai in the following code:

```
frm <- isGood ~ acc_chk_sts + credit_hist + savings + property +
  purpose + duration + credit_amount + age
m <- glm(formula = frm, data = df, family = "binomial")
summary(m)
```

```
##
## Call:
## glm(formula = frm, family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8307  -0.7925   0.4375   0.7505   1.8630
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.05599    0.58977  -1.791  0.07337 .
## acc_chk_stsnone  1.62357    0.21238   7.645 2.10e-14 ***
## acc_chk_stspos   0.51114    0.18286   2.795  0.00519 **
## credit_histcrit  1.58161    0.29876   5.294 1.20e-07 ***
## credit_histpaid  0.85618    0.26276   3.258  0.00112 **
## savingsother    -0.20481    0.50248  -0.408  0.68356
## savingssmall    -0.94107    0.47261  -1.991  0.04645 *
## propertynone    -0.40059    0.22911  -1.748  0.08039 .
## propertyreal     0.26633    0.19639   1.356  0.17505
## purposeinvest    0.49070    0.20042   2.448  0.01435 *
## purposeuseful    0.95140    0.19908   4.779 1.76e-06 ***
## durationL        1.17874    0.26203   4.498 6.85e-06 ***
## durationM        0.60451    0.22207   2.722  0.00649 **
## credit_amountL  -0.08644    0.25595  -0.338  0.73557
## credit_amountM   0.41477    0.24717   1.678  0.09334 .
## ageL            -0.41982    0.16484  -2.547  0.01087 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  971.41  on 984  degrees of freedom
## AIC: 1003.4
##
## Number of Fisher Scoring iterations: 5
```

5 The performance of the Model

We will check the performance of the model with the aid of the ROCurve.

The model has a an AUC of 0.796981, and a KS of 0.4590476. These findings are not bad and we will proceed now to calculate the optimal cutoff. This cutoff is as follows:

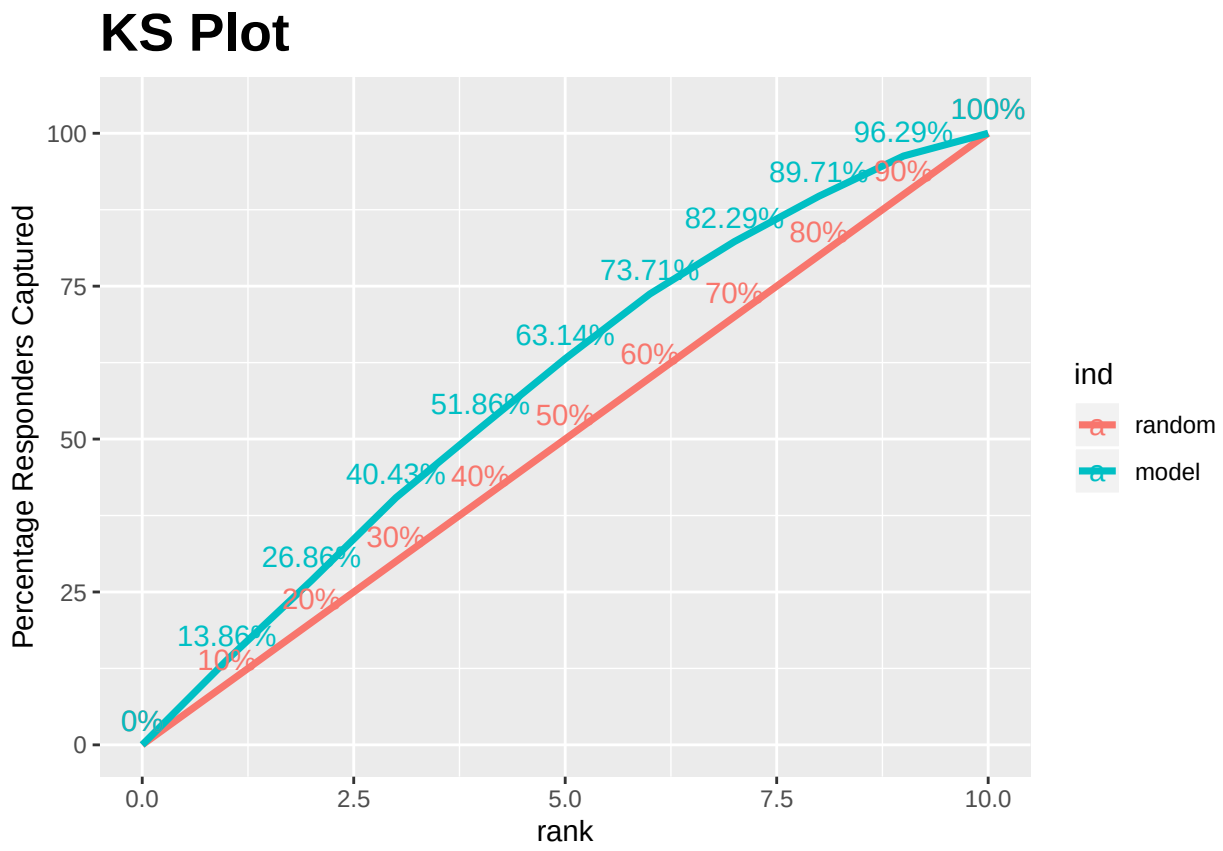

Figure 4: The ROC (receiver operating curve) for our model

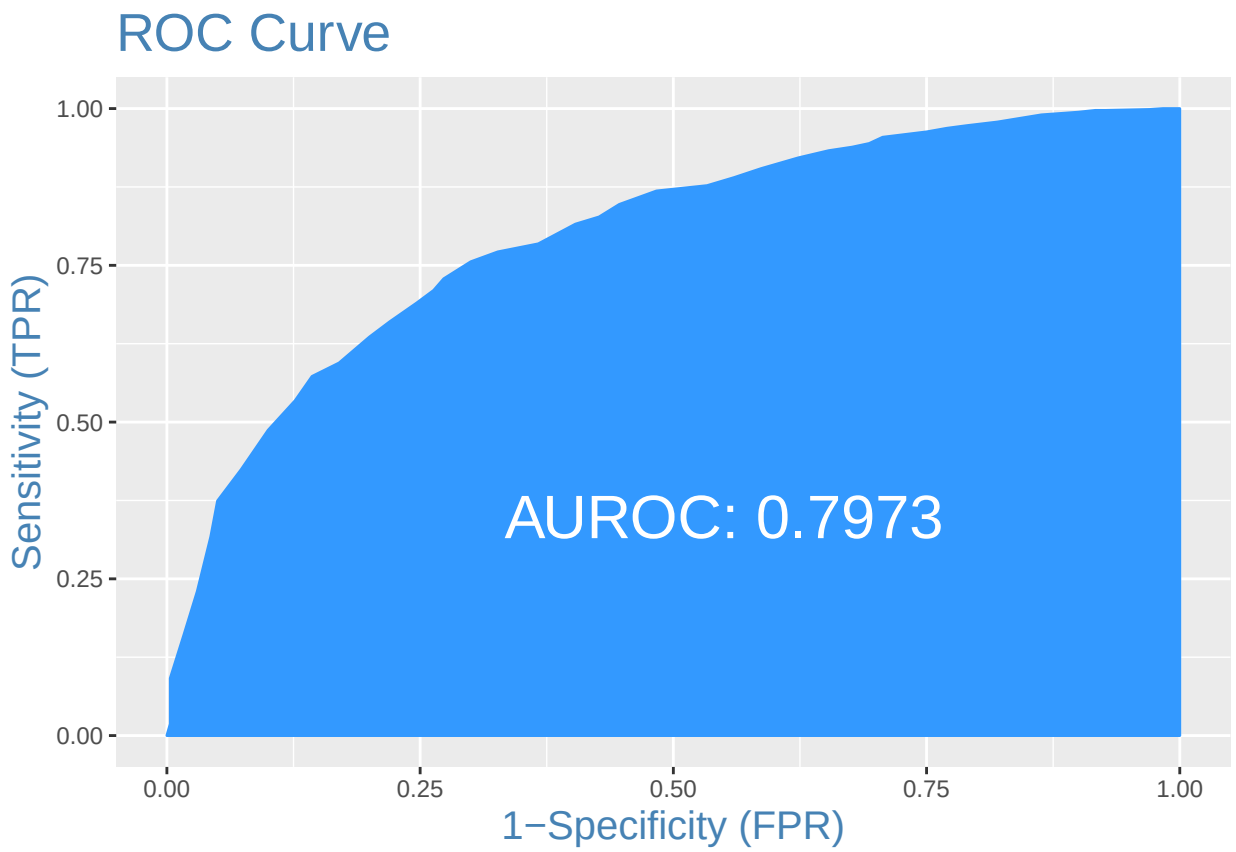

Figure 5: The lift of the model (bottom): the cumulative percentage of responders (ones) captured by the model

```
## The optimal cutoff if the false positives cost 10 times more than the false negatives:
##                   [,1]
## sensitivity 0.3500000
## specificity 0.9566667
## cutoff      0.8923711
```

We can also use the package $InformationValue$ to calculate the optimal cutoff:

```
optimalCutoff(actuals = df$isGood, predictedScores = predScores, optimiseFor = "Zeros")
```

```
## [1] 0.9914701
```

```
optimalCutoff(actuals = df$isGood, predictedScores = predScores, optimiseFor = "Both")
```

```
## [1] 0.6514701
```

```
optimalCutoff(actuals = df$isGood, predictedScores = predScores, optimiseFor = "Ones")
```

```
## [1] 0.1714701
```

```
optimalCutoff(actuals = df$isGood, predictedScores = predScores, optimiseFor = "misclasserr
```

```
## [1] 0.5514701
```

6 Validation of the Model

To validate the model we will use the cross validation method, and opt for the Monte Carlo Cross Validation. [^While we could opt for the *k-fold* validation (with k not too high), we choose the Monte Carlo methods because it allows to draw more random selections.]

6.1 Monte Carlo Cross Validation

We use the Monte Carlo Cross Validation with a test dataset that spans 30% of our observations and 70% in the training data-set. We will draw 200 times a training data-set of 0.7 and study the AUC on the testing data-set.

The results are in Figure 6. We notice that the model is over-fit, but even for the test data-set the performance is acceptable. All variables retained have at least one of the categories with a $p-value$ that is smaller than 0.01. We could consider to leave out the categories that have a higher $p-value$ to make the model more robust. We choose, however, not to do that because all coefficients are somehow logical.

The median of the observed values for the AUC of the test data is 0.7760046, the average is 0.7744981 with a standard deviation of 0.0248717.

7 The Challenger Models

We will build two challenger models.

7.1 Neural Network

As a challenger model, we us a neural network on the same binned data and with three hidden layers with resp 16, 8, and 4 hidden neurons.

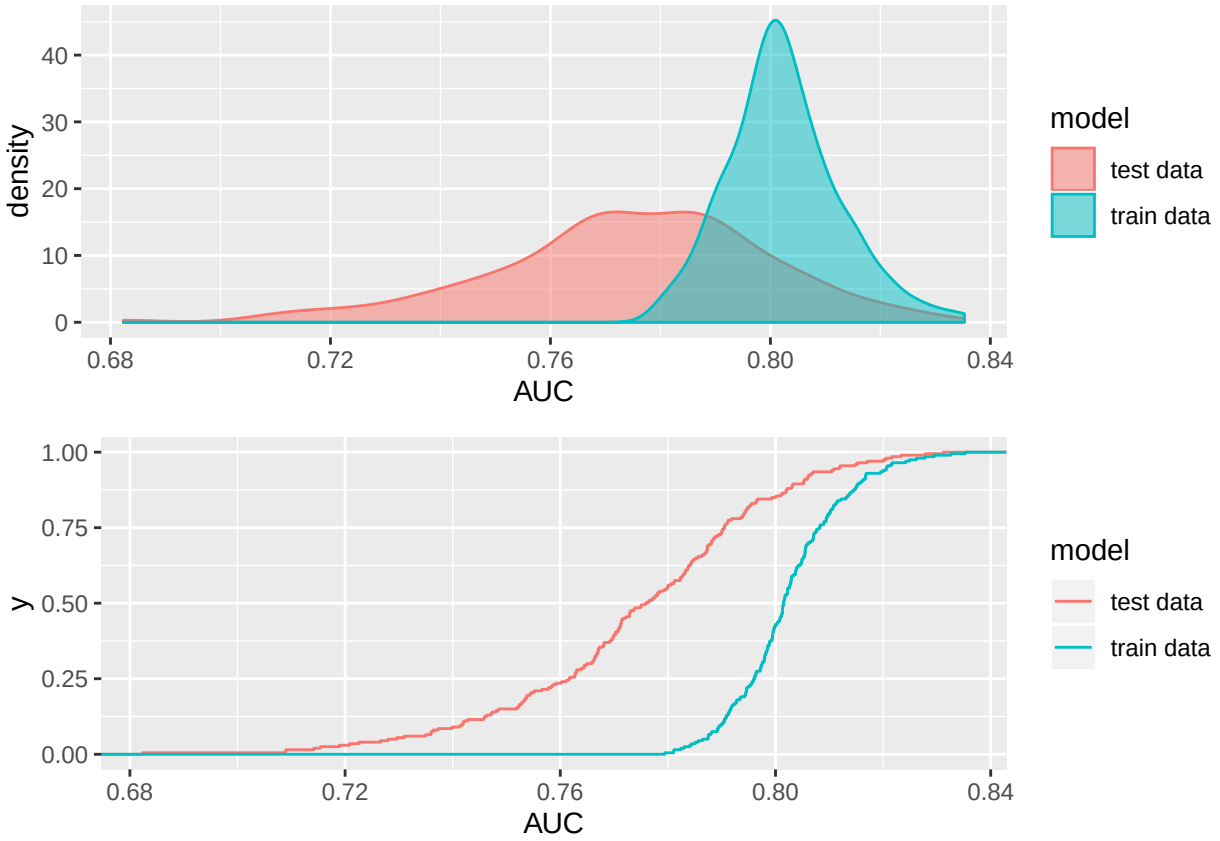

Figure 6: The histogram for the AUC of the ramdomised data-sets with the Monte Carlo cross vailidation for the first logistic regression model.

First we fit the model on all the data:

```
## [1] "AUC: 0.837145238095239"
```

```
## [1] "KS: 0.56952380952381"
```

Now we perform the Monte Carlo Cross Validation for the neural network. The results are in Figure 8.

7.2 Another logistic regression: logistic 2

We use also the installments as percentage of income, the correlation was small but it is usual a good variable to use for consumer lending. We also decide the parameter as it is. The variable is a number between 1 and 4.

```
summary(d$installment_as_income_perc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   2.973   4.000   4.000
```

The variable is encoded as a number between 1 and 4. We believe this is not too different from the other binary variables, and will fit the model with the variable unchanged.

```
d2 <- cbind(df, installment_as_income_perc = d$installment_as_income_perc)

frm2 <- isGood ~ acc_chk_sts + credit_hist + savings + property +
  purpose + duration + credit_amount + age + installment_as_income_perc
```

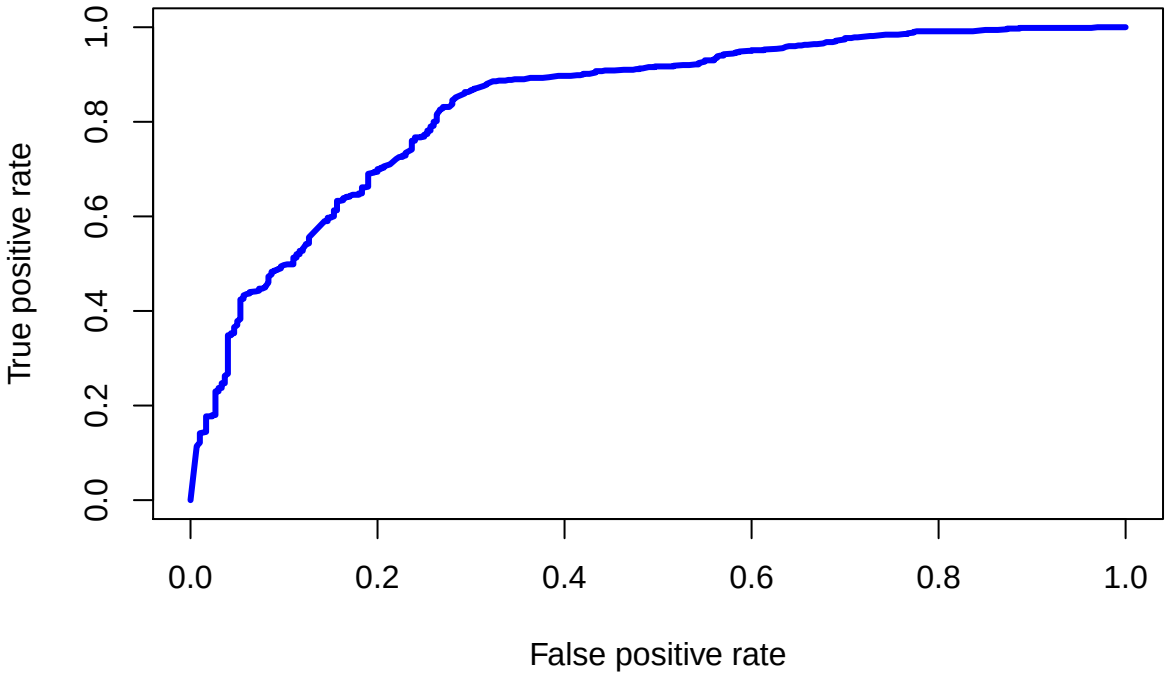

Figure 7: The ROC for the neural network with three hidden layers of 14, and 7 neurons respectively.

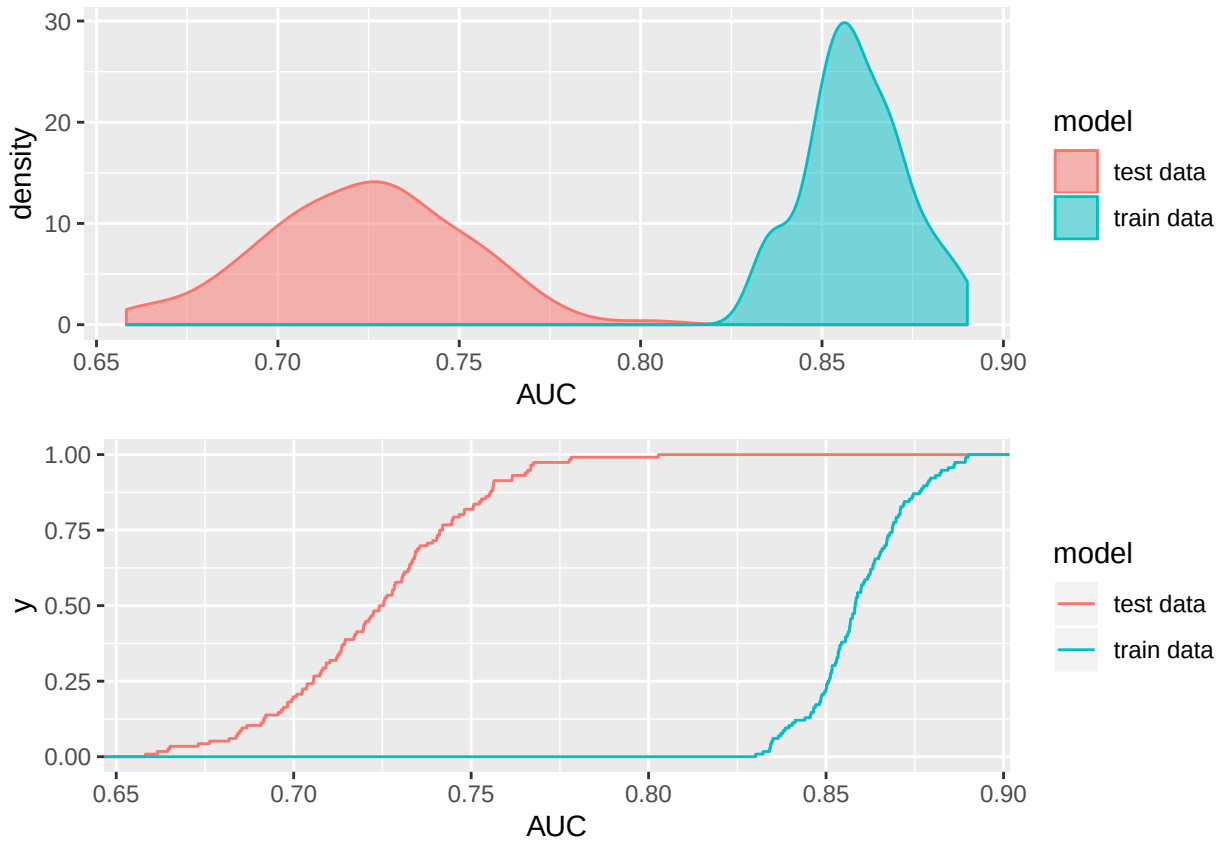

Figure 8: The results of the cross vaildation for the neural network. We notice that the neural network witho nly three layers does not significantly overfit and hence is a valid model.

```
m2 <- glm(formula = frm2, data = d2, family = "binomial")
summary(m2)
```

```
##
## Call:
## glm(formula = frm2, family = "binomial", data = d2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7935  -0.7903   0.4477   0.7375   1.9732
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -0.48358    0.63000  -0.768 0.442736
## acc_chk_stsnone              1.62035    0.21346   7.591 3.18e-14 ***
## acc_chk_stspos               0.47540    0.18407   2.583 0.009802 **
## credit_histcrit              1.59784    0.29936   5.338 9.42e-08 ***
## credit_histpaid              0.85087    0.26278   3.238 0.001204 **
## savingsother                -0.24001    0.50614  -0.474 0.635362
## savingssmall                -0.98737    0.47634  -2.073 0.038188 *
## propertynone                -0.33859    0.23090  -1.466 0.142554
## propertyreal                 0.23503    0.19768   1.189 0.234458
## purposeinvest                0.49469    0.20114   2.459 0.013919 *
## purposeuseful                0.99707    0.20075   4.967 6.81e-07 ***
## durationL                    1.02004    0.26935   3.787 0.000152 ***
## durationM                    0.51848    0.22425   2.312 0.020778 *
## credit_amountL               0.20386    0.27955   0.729 0.465840
## credit_amountM               0.52983    0.25155   2.106 0.035183 *
## ageL                        -0.42586    0.16536  -2.575 0.010012 *
## installment_as_income_perc -0.20854    0.08021  -2.600 0.009323 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  964.52  on 983  degrees of freedom
## AIC: 998.52
##
## Number of Fisher Scoring iterations: 5
```

```
## [1] "AUC: 0.800592857142857"
```

```
## [1] "KS: 0.465714285714286"
```

The coefficient of *installment_as_income_perc* is significant (the p-value is 0.009323). This is encouraging.

We will now subject this model to the same Monte Carlo cross validation and compare the results.

```
cv_mc2 <- crossv_mc(d2, n = nRuns, test = 1 - pctTrain)
mods2  <- map(cv_mc2$train, ~ glm(frm2, data = .))
```

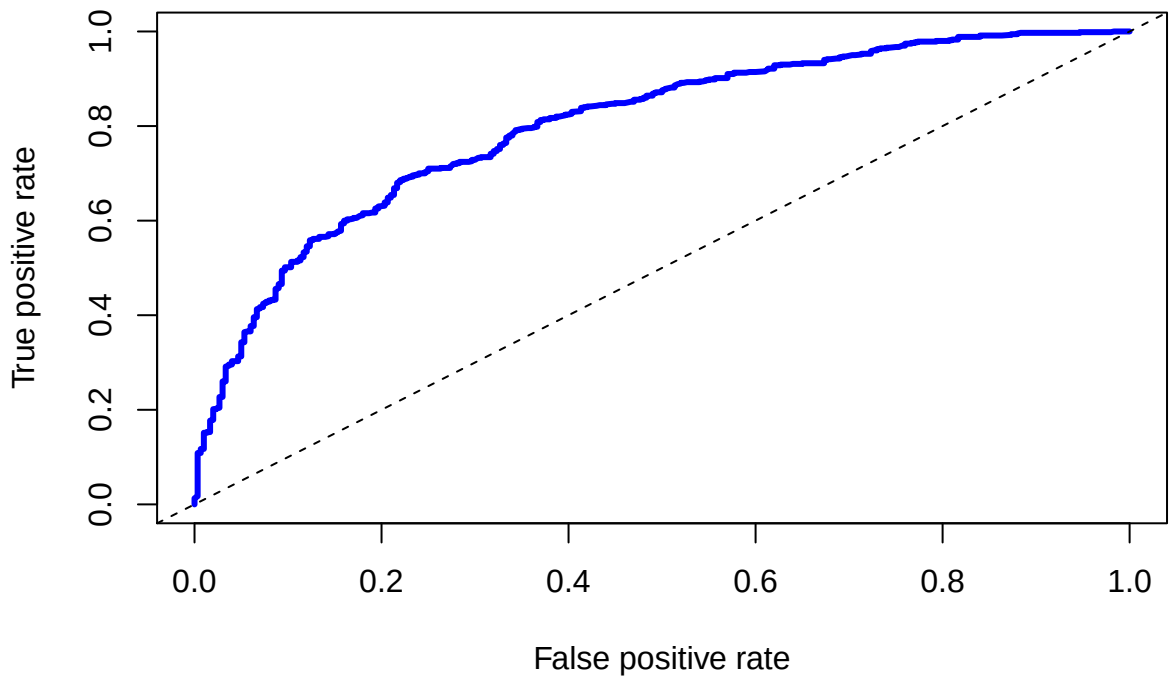

Figure 9: The ROC for the logistic challenger logistic regression

```
AUCs2_test  <- numeric(0)
AUCs2_train <- numeric(0)
for(k in 1:nRuns) {
  AUCs2_test[k]  <- fAUC(mods2[[k]], as.data.frame(cv_mc2$test[[k]]))
  AUCs2_train[k] <- fAUC(mods2[[k]], as.data.frame(cv_mc2$train[[k]]))
}

allAUCs <- rbind(tibble(model = "train data", AUC = AUCs2_train),
                 tibble(model = "test data", AUC = AUCs2_test))
p1 <- ggplot(allAUCs, aes(AUC, fill = model, colour = model)) +
      geom_density(alpha=0.5)
p2 <- ggplot(allAUCs, aes(AUC, fill = model, colour = model)) +
      stat_ecdf()
grid.arrange(p1, p2, ncol = 1)
```

The median of the observed values for the AUC of the test data is 0.779845, the average is 0.7788554 with a standard deviation of 0.0253224.

8 Conclusion

Comparing the two logistic regressions it seems that the challenger model that uses the additional variable (debt installments as a percentage of income), but also prone to over-fitting. The neural network performs significantly better on the training data, but is even more over-fit: it preforms worse on the testing data. For that reason we have to reject this particular neural network.

In the following table we summarise these results:

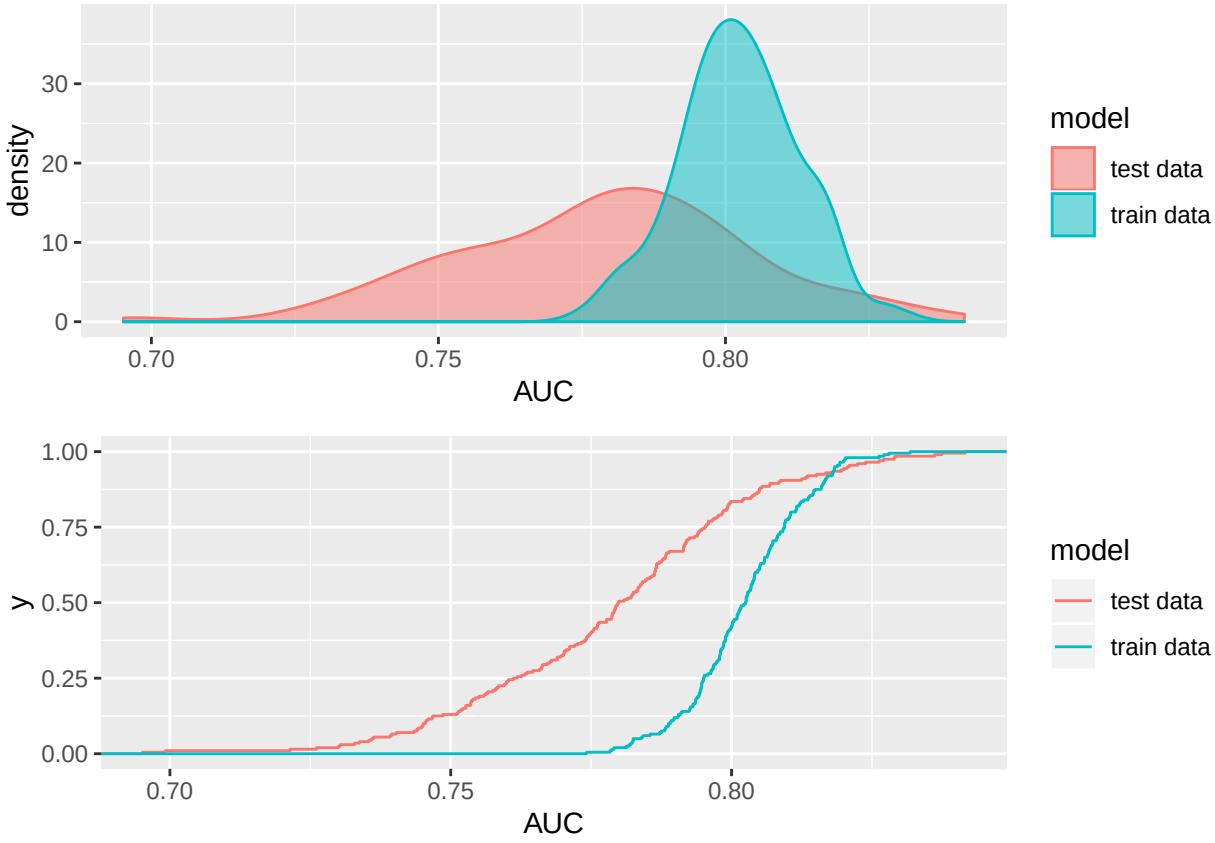

Figure 10: The distribution of the AUC for the challenger model "logistic 2."

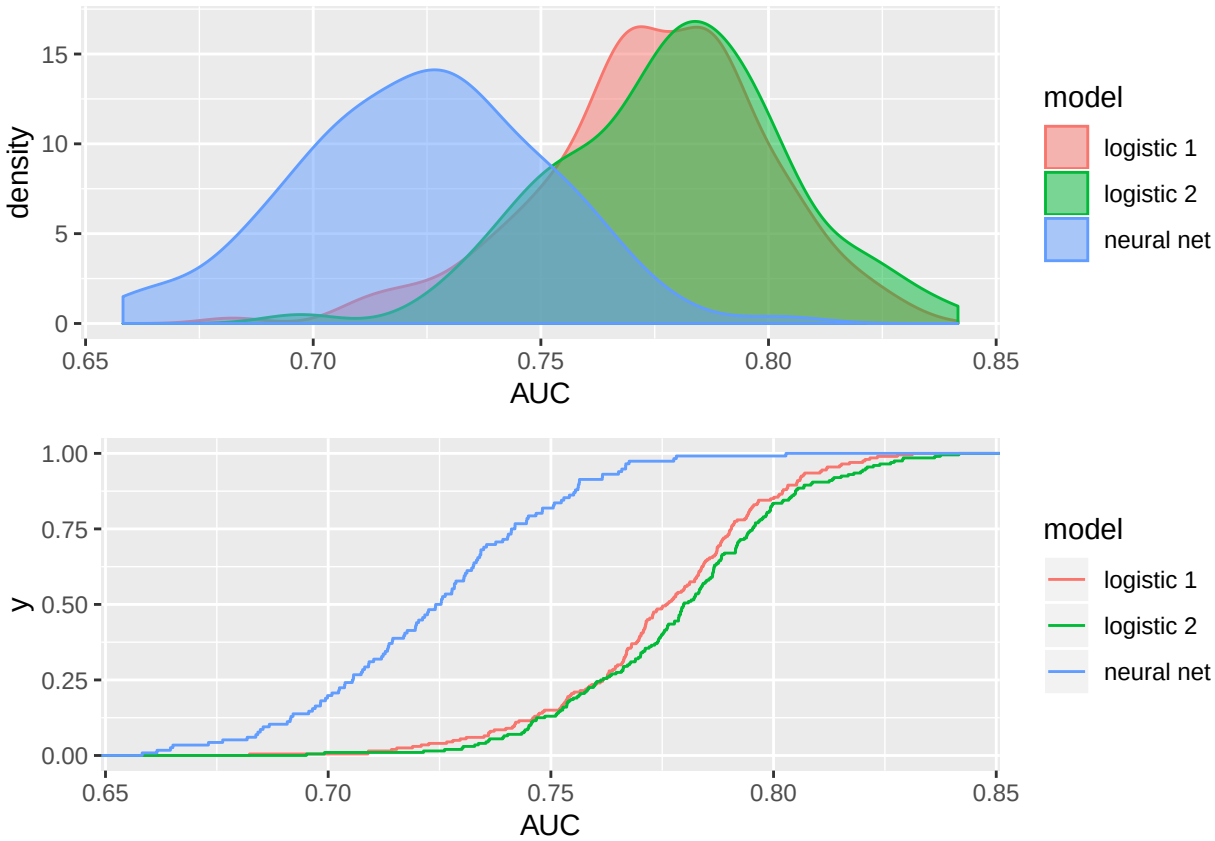

Figure 11: The kernel density for the observed areas under the curve (top) and the cumulative probability density functions (bottom) for the base and challenger models. All AUCs shown are for the test data only.

Model	Quality	Mean AUC on Test Data	AUC on All Data
logistic 1	slightly over-fit	0.7744981	0.796981
logistic 2	slightly over-fit	0.7788554	0.8005929
neural network	significantly over-fit	0.7235076	0.8371452

The decisions of the logistic regression are transparent in that sense that they can be explained to the customerˆ[The explainability is important for the bank (who seeks assurance that he or she does the right thing, the customer and eventual courts. The neural networks seems to over-fit more – so we would suggest to look into simpler neural networks with less neurons and/or layers.

The overall best models that we have is "logistic 2'' and we recommend this for implementation.

Bibliography

De Brouwer, Philippe J.S. 2020. *The Big r-Book: From Data Science to Learning Machines and Big Data*. New York: John Wiley & Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781119632757.