Credit Analysis

Daniel Podolecki, Matthias De Paolis, Lukas Niederhaeuser

10/6/2022

Figure 1: Symbolic picture of a bank

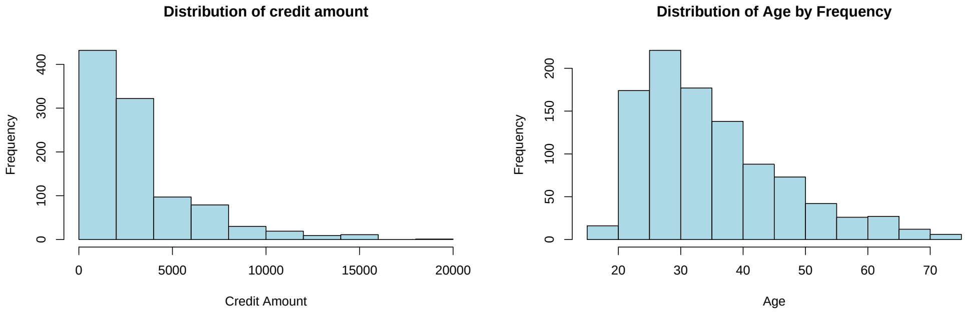
1. Introduction / Dataset Overview

For the client, which is a bank different machine learning methods are applied to a dataset. The dataset contains information on customers of this bank regarding credits. It contains 1'000 observations and 21 variables. It contains information about age, gender, credit amount, credit duration etc.

The client has following research questions:

- Is there a difference for the credit-amount and duration for particular customers?
- Can our clients be categorized?
- Can a prediction been made, whether a client has good or bad liability?

In a very first step the data is getting checked visually. First the credit amount as well as the age distribution of the clients is visualized.



When investigating the data further following conclusion can be made:

Most of the credits have an amount between 0 and 5000 Euro. This means the usual credit amount is quiet small. The duration of a credit vary, most of the observations lie between 10 - 15 years and 20 to 25 years.

Most clients have an age between 20 and 40 years. Male make up for 69% of the observations of the dataset. Thereof 54.8% are single males which have a credit line. Only 31% of the credits are from females. The purpose of the credit is different, the purposes which are most often occurring are radio / TV, new car and furniture - equipment.

Clients with the classification "bad" have a higher median credit amount than clients with class "good". This means: on average, bad customers want a higher loan amount than good customers.

When a client doesn't have a property the median of the credit is the highest. For real estate as property the credit amount is the lowest. Most of clients take out a loan to buy a new radio/TV, a new car or new furniture. Regarding the savings: 60 % of the clients have savings below 100k. That makes sense, that's why they want to have a credit.

2. Linear Model

The lead / responsibility for this model has Lukas Niederhaeuser.

In a first step the correlations within the dataset are assessed to define the response variables and the predictors, which are used for the linear model. The variables are getting assessed based on their correlation, which means that first only numerical variables are included for the assessment.

Correlogram: From the Correlation-Plot it can be seen that credit duration and credit amount are highly correlated. Before we assessed only the numerical variables. Since the dataset includes several categorical variables we will assess their relationship with a pair-plot to check the dataset for further relationships.

Pair-Plots: The variable "existing credits" seems also to have an influence on credit duration and amount. One would assume that if a client already has existing credits, the credit amount we would give is decreasing. This seems logical. Other than that no variables seem to have a clear relationship.

2.1. Linear Model Assessment

Since the above analysis showed some indication on which variables to chooses. We first perform a variable selection procedure based on stepwise regression. With this we try to find the the explanatory variables that describe credit amount best.

When computing the linear model with all variables included in the dataset, it can be seen that first of all too many variables are selected and secondly we have a few variables / categories which are stat. significant. These are:

Numerical: Duration and installment_commitment

Categorical: Checking_status = '0<=X<200', credit_history = 'no credits/all paid', purpose = 'used car' and 'other', savings_status = 'no known savings', personal_status = 'male single', property_magnitude = 'no known property', job = 'unemp/unskilled non res', job = 'unskilled resident', job = skilled, own_telephone = yes and class = good.

2.2. Stepwise Regression - backwards selection

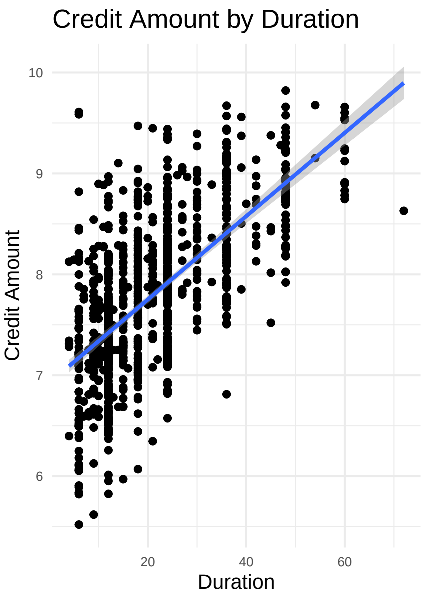
Applying backwards selection as well as forward selection to choose the best most "important" variables, we are left with following explanatory variables for credit amount as response variable:

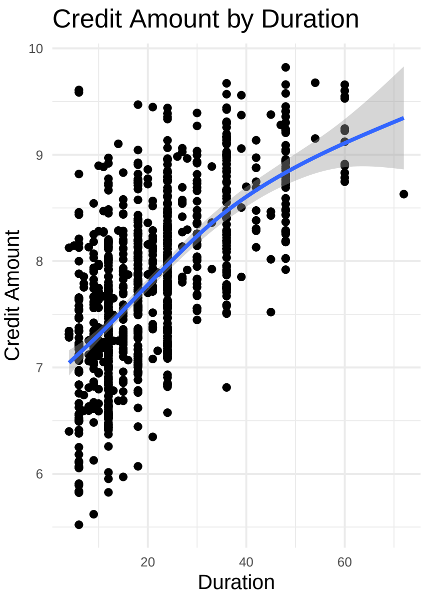
- checking_status
- duration
- credit_history
- purpose
- savings_status
- installment_commitment
- personal_status
- other_parties
- property_magnitude
- job
- own_telephone
- class

Now we start to create a linear model. The approach is to first calculate a linear model with only one variable, then add explanatory variables to the model to compare the explanatory power.

2.2. Linear Model with one predictor

As a response variable the credit amount is defined. As explanatory variable credit duration is taken, since this variable seems to have the most explanatory power for credit amount. Since credit amount can not be negative and is furthermore right-skewed, we apply log-transformation on the variable. First we plot the linear model visually. On the left hand side without smoothed abline and on the right hand side the smoothed abline:





It seems that credit amount and credit duration have a positive linear relationship. This makes sense: As the duration of a credit increases the credit amount increases as well.

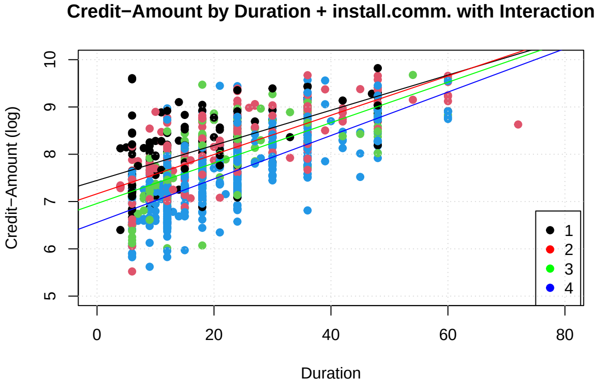
Interpretation: The explanatory variable duration has a p-value, which is less than the significance level of 0.05, this indicated a strong effect on the response variable. The adjusted R-squared is 0.41. This means that 41% of the variance in credit amount can be explained with duration. Furthermore, an interpretation about the change in duration can be made: For a one unit increase credit amount, the duration increases by 4.21 percentage.

When comparing the linear model with intercept only (the most basic model) and the linear model including credit duration as independent variable via an F-Test, it can clearly be stated that there is strong evidence that the model with duration better fits the data.

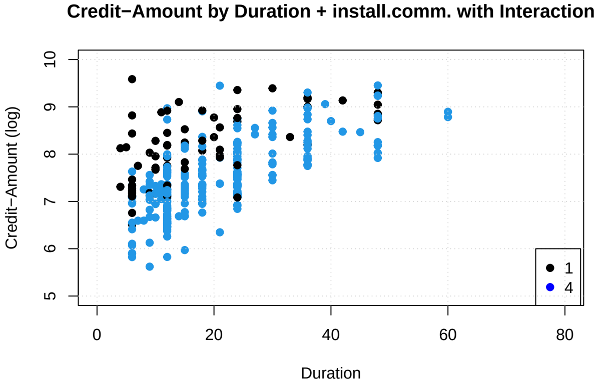
2.3. Multiple Linear Regression (two predictors)

As defined above there are further variables which are increasing the explanatory power of the credit amount. For the second "installment commitment" is added to the model, since the pair-plots above indicated that this variable seems to have strong impact on credit amount. First the boxplot is computed to check whether there are differences between the median-value of credit amount between the different categories of installment commitments.

The amount of credit given seems to differ in terms of the mean-value as well as the median-value for credit-amount by the level of installment-commitment. Now a check of the interceptions is performed checking for different intercepts between the categories of installment-commitments. By default R uses "installment-commitment1" as reference level.



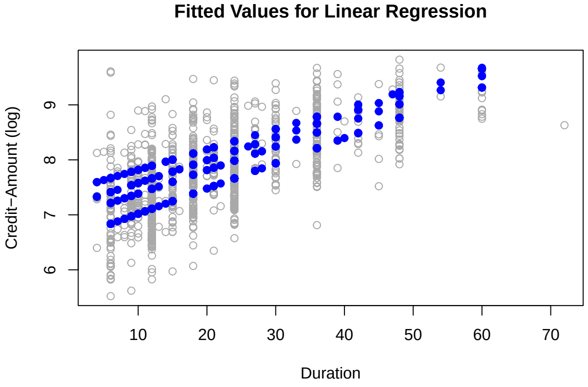
It can be seen that there is a difference between the intercepts for the different categories of installment-commitment. Clients with one installment commitment (eine Ratenzahlung) seem to receive in general higher credit amounts. On the other side clients with four (4) installment commitments seem to receive lower credit amounts in general. This can also be observed visually when comparing the black and the blue dots (see below):



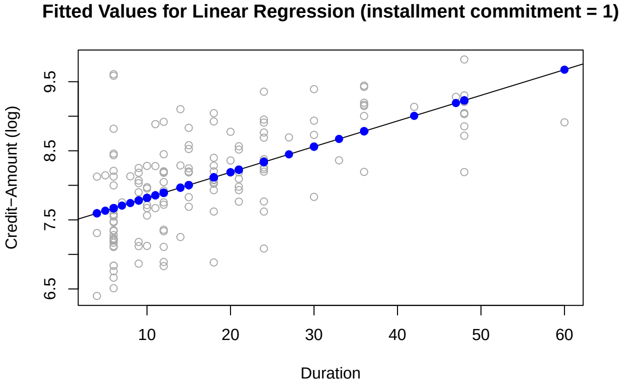
Since we computed a multiple linear regression by adding installment-commitment as an explanatory variable, a comparison by Adjusted R-Squared as well as an F-Test is performed. Comparing the first model with duration as the only explanatory variable with the model having both duration and installment-commitment as explanatory variable.

In the simple linear regression model we obtained an adjusted R-squared of 41%. With the multiple linear regression we could improve the adjusted R-squared to 53.5%. When comparing the models with an F-Test, we obtain a stat. significant result, indicating that the multiple linear regression model has a better performance and installment-commitment plays a relevant role in explaining the variance in credit amount.

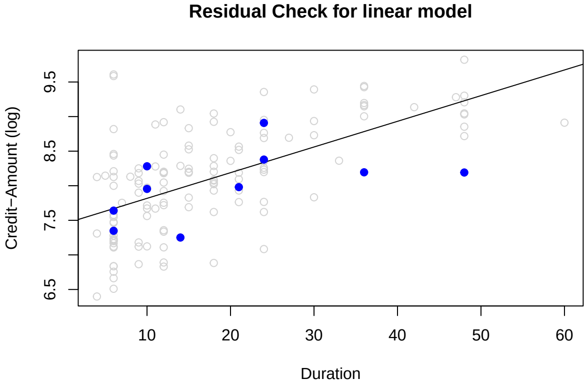
Let's check the modelling we have done. For this we are fitting the values according to our linear model and take a sample out of the data to see whether our predictions seems plausible.



It can be observed in above plot that fitted values are calculated for each category of installment commitment level. There seems to be a deviation between the fitted values and the actual values. To go into more detail we fit values for only one category of installment commitment, which is in below case 1. The first plot is showing the fitted values for the newly computed linear model only including the filtered data with installment commitment = 1.



The fitted values on the abline (blue) are showing a strong deviation to the actually observed datapoints indicated in grey. Now also a random sample is taken out of the dataset and checking visually.



Interpretation: In the above plot it can be seen that some values are fitting the regression line quite well are are therefore predicting the credit amount in a useful manner. However there are some points which are strongly deviating from the regression line are the linear model as computed above would not give a good prediction for credit amount and duration.

2.4. Multiple Linear Regression (three predictors)

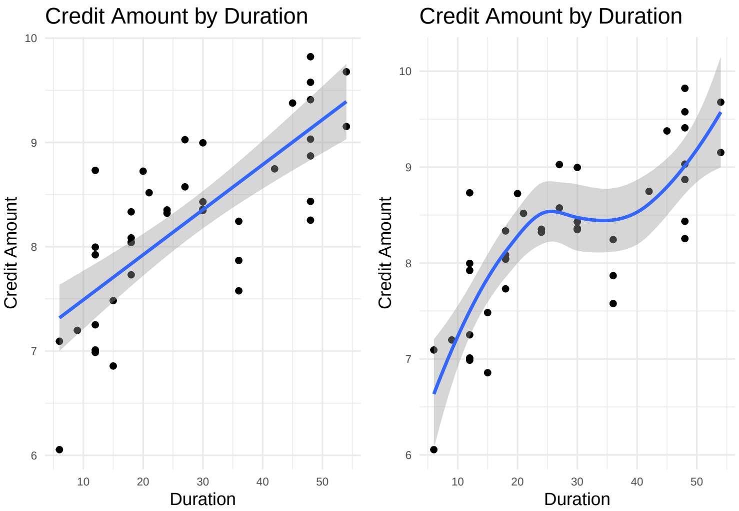
Since the optimal linear model, obtained with backwards selection achieved an adjusted R-squared of 0.639, we try to obtain a similar result with an additional variable. Still, the goal is to predict credit-amount. As in the previous steps we keep following explanatory variables: duration, installment-commitment. We add another explanatory variable which is "job" and compute the linear model:

```
##
## Call:
## lm(formula = log(credit_amount) ~ duration + installment_commitment +
##     job, data = credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74768 -0.34499  0.02212  0.32366  2.20782
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.825718   0.065986 118.596  < 2e-16 ***
## duration                     0.040165   0.001353  29.678  < 2e-16 ***
## installment_commitment2     -0.230143   0.054570  -4.217 2.70e-05 ***
## installment_commitment3     -0.418998   0.059009  -7.101 2.37e-12 ***
## installment_commitment4     -0.753209   0.049288 -15.282  < 2e-16 ***
## job'unemp/unskilled non res' -0.688843   0.115535  -5.962 3.46e-09 ***
## job'unskilled resident'     -0.567427   0.055615 -10.203  < 2e-16 ***
## jobskilled                  -0.429351   0.046039  -9.326  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5009 on 992 degrees of freedom
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.5839
## F-statistic: 201.3 on 7 and 992 DF,  p-value: < 2.2e-16
```

We are able to achieve a similar result in terms of adjusted r-squared by only using three explanatory variables instead of 12 as the stepwise regression suggested. We were therfore able to reduce the complexity drastically.

3. Non - Linearity

For the model which was computed above a linear relationship makes most sense. But when investigating the dataset and different relationships, there are also some were a non-linear relationship can be assumed. To visualise this the relationship between credit-amount, duration and credit-history is assessed. In the below plot, the data has been filtered for clients with a credit-history "no credits / all paid". This means clients who did not have a previous credit or have repaid all there previous credits. On the plot on the left we assume a linear relationship, on the plot on the right hand side, non-linearity is assumed.

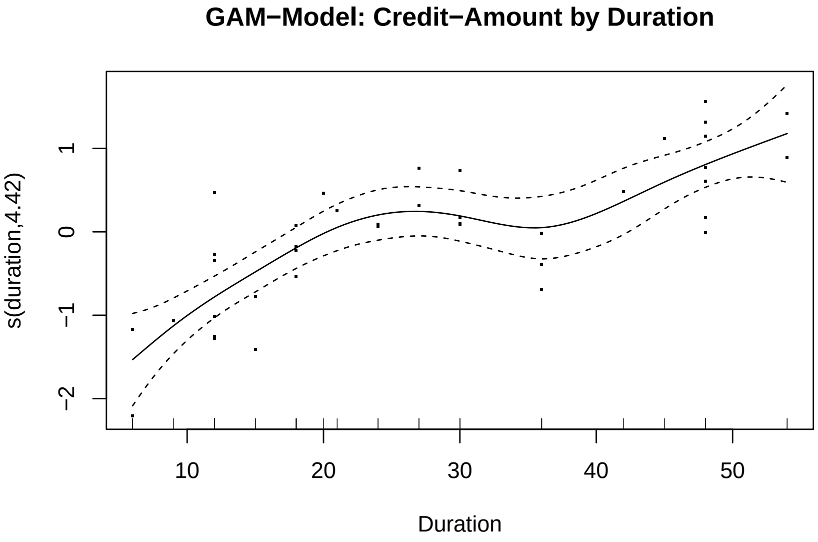


Since a non linear relationship can be stated for the above constellation. A comparison between two model is performed. One model which is assuming a linear relationship and another model assuming a non-linear relationship. For the non-linear assumption a higher order polynomial (cubic polynomial) is used to express the relationship.

The comparison of the two model in terms of adjusted R-squared gives the following results: Linear relationship has an adjusted R-squared of 0.57, explaining 57% of the variance in credit-amount. The non linear model has an adjusted R-squared of 0.63, explaining 63% of the variance in credit-amount. Comparing these measures together, it seems that a non-linear relationship does explain the variance better. When comparing the models with an F-Test the results indicated as well, that the non-linear model seems to fit the data better.

3.1 Fitting a Generalized Additive Model (GAM)

Since a non-linear relationship can be assumed a Generalized Additive Model (GAM) model is computed.



```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(credit_amount) ~ s(duration)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.2628     0.0778   106.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(duration) 4.418  5.431 14.43  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.662   Deviance explained =   70%
## GCV = 0.28007  Scale est. = 0.24214   n = 40
```

The output for the GAM-Model indicates strong evidence that duration has a non-linear effect on credit-amount. The edf for duration is with 4.41 above 1, which indicates non-linearity. Furthermore the p-Value for duration is statistically significant. The adjusted R-squared is with 0.66 higher than with the above computed cubic polynomial.

Interpretation: The QQ-Plot, which compares the model residuals to a normal distribution indicates that the model's residuals are close to a straight line. However in the upper part of the line, the model residuals deviate sightly from a normal distribution. This is also indicated in the histogram. It is not perfectly symmetrically bell shaped, indicating slight right-skewness. It can be concluded that the GAM-Model which is fitted to the data seems to be indicative, but does not indicate a perfect fit.

4. Generalised Linear Models (GLM)

In the dataset there are also binary data, represented by 1 and 0 (or success and failure), as well as count data. In a first step the variables are assessed to categorize them into count data and binary data.

Count Data:

- duration
- existing_credits
- installment_commitment

Binary / Binomial Data:

- Own telephone
- Foreign worker
- Class
- gender (extracted from personal status)

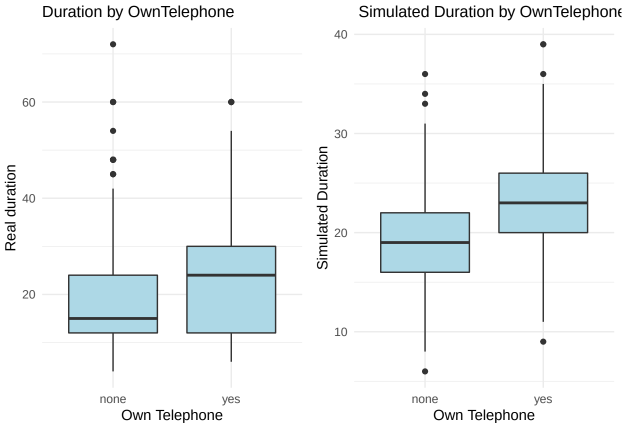
4.1 Count Data

In order to analyze the available count data to the age attribute we will divide the variable age into different groups.

Count data has the property that the variance is often not normally distributed. Also the values are discrete and strict positive. Therefore, a linear model may not give us realistic results. We have to change our approach and use a generalized linear model with a poisson distribution to simulate reality more accurate.

Let's model duration vs owning a telephone:

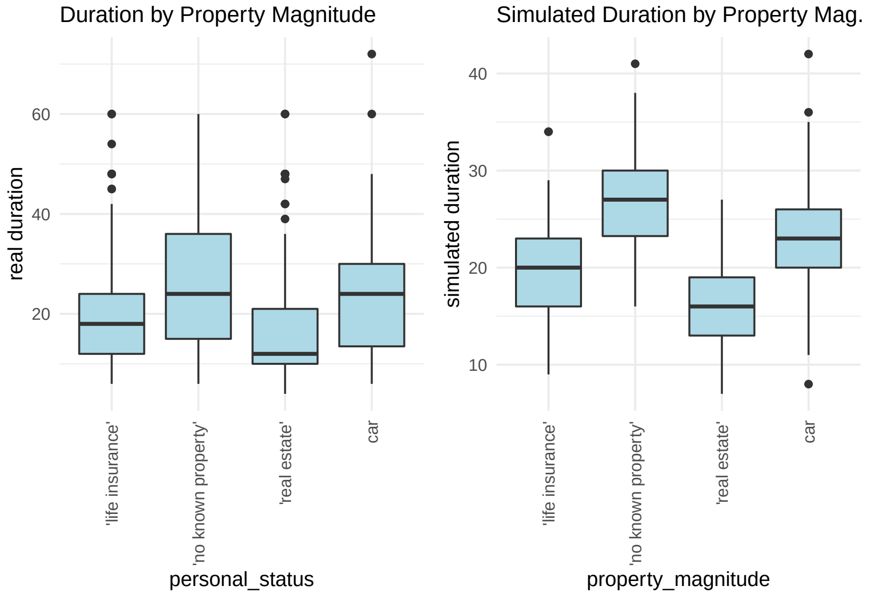
```
##
## Call:
## glm(formula = duration ~ own_telephone, family = "poisson", data = credit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2826  -1.8434  -0.2922   1.0378   9.1846
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.958469   0.009332  317.04   <2e-16 ***
## own_telephoneyes 0.190600   0.013901   13.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6457.2  on 999  degrees of freedom
## Residual deviance: 6270.7  on 998  degrees of freedom
## AIC: 11001
##
## Number of Fisher Scoring iterations: 5
```



Interpretation: It can be seen in the summary output that owning a telephone is highly significant to the response variable duration. This indicates that there is a difference on the duration when a client is owning a telephone or not. In the boxplot it is also indicated visually. Clients who own a telephone have a higher median duration than clients who don't own a telephone. The model seems a good fit.

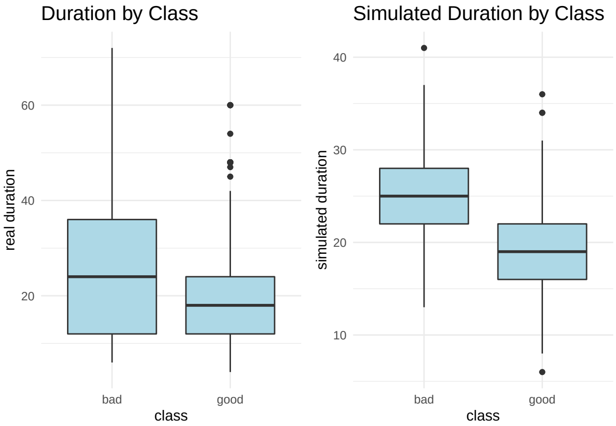
Further Analysis: Further binary data and their impact on the response variable duration are performed. The variable “property_magnitude”, which indicates what material properties the client owns is also significant to the defined response variable. The attribute ‘class’ has also a statistically significant influence on the response variable duration, indicating that bad debtors have a higher median duration than good debtors. The boxplots below shown the results graphically

```
##
## Call:
## glm(formula = duration ~ property_magnitude, family = "poisson",
##     data = credit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8806  -1.8631  -0.5726   1.4055   8.3257
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            2.97856    0.01481 201.162   <2e-16 ***
## property_magnitude’no known property’  0.31415    0.02146  14.639   <2e-16 ***
## property_magnitude’real estate’       -0.19123    0.02092  -9.141   <2e-16 ***
## property_magnitudecar                  0.15457    0.01872   8.256   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6457.2  on 999  degrees of freedom
## Residual deviance: 5813.3  on 996  degrees of freedom
## AIC: 10547
##
## Number of Fisher Scoring iterations: 4
```



The attribute ‘class’ has also a statistically significant influence on the response variable duration based on the p-values, indicating that bad debtors have a higher median duration than good debtors. The boxplots below shown the results graphically

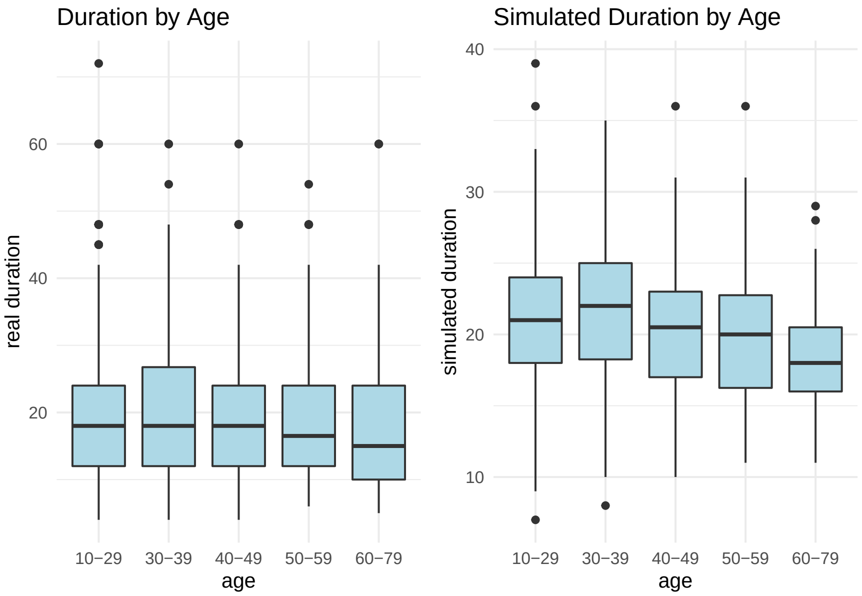
```
##
## Call:
## glm(formula = duration ~ class, family = "poisson", data = credit_age_interval)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5455  -1.7678  -0.2784   1.0523   7.6714
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.21326    0.01158  277.50   <2e-16 ***
## classgood   -0.25798    0.01444  -17.87   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6457.2  on 999  degrees of freedom
## Residual deviance: 6146.5  on 998  degrees of freedom
## AIC: 10876
##
## Number of Fisher Scoring iterations: 4
```



The attribute ‘age’ has no statistically significant influence on the response variable duration. The boxplots below shown the results graphically.

```
##
## Call:
## glm(formula = duration ~ age, family = "poisson", data = credit_age_interval)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6598  -2.1013  -0.6327   0.7982   8.7356
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.03589    0.01138 266.813  < 2e-16 ***
## age30-39     0.03695    0.01642   2.250  0.02447 *
## age40-49    -0.02532    0.02031  -1.247  0.21255
## age50-59     0.01121    0.02777   0.404  0.68655
## age60-79    -0.10078    0.03422  -2.945  0.00323 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6457.2  on 999  degrees of freedom
## Residual deviance: 6435.4  on 995  degrees of freedom
## AIC: 11171
##
## Number of Fisher Scoring iterations: 5
```

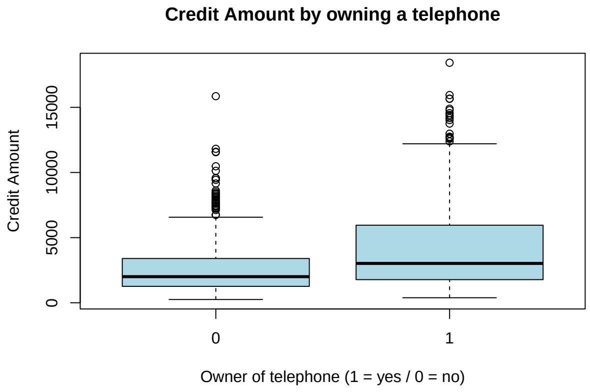

Duration by Age

Simulated Duration by Age

4.2 Binary Data

In order to analyze the binary data we will transform the variables of interest such as: owning a telephone, classification, foreign worker and gender. In a first step the binary data as a predictor of credit amount is computed.

It can be seen in the summary output that owning a telephone is highly significant to the repsonse variable credit amount. This indicates that there is a difference on the credit amount when a clinet is owning a telephone or not. In the below boxplot it is also indicated visually. Clients who own a telephone have a higher median credit amount and also a higher variance than clients who don’t own a telephone.

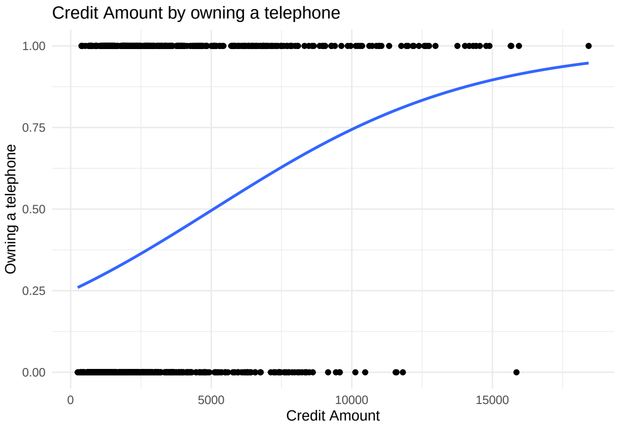


Further Analysis: Further binary data and their impact on the response variable *credit amount* are performed using 'GLM' simulating the distribution with 'Quasipoisson'. The following insights have emerged:

- The variable "class", which indicates if a customer is rated as as good or bad debtor, is significant for the defined response variable.
- The variable "foreign worker", which indicates whether the client is a domestic or a foreign worker is not significant for the defined response variable.
- Gender on the other hand has a statistically significant influence on the response - variable credit amount, indicating that male clients have a higher median credit amount than females.

4.3 Binary Data: Logistic Regression

Since above analysis showed a statistically significant influence whether a client is owning a telephone or not on the response variable credit amount, the relationship between the two variables is being plotted below:



Interpretation:In the above graph we can see that the blue line, which is moving upwards. This indicates that with increasing credit amount the probability that the client is owning a telephone is increasing as well. Therefore owning a telephone is weakly associated with the credit amount, as also the summary in an above example stated.

Now a logistic regression is fitted on the data.

```
##
## Call:
## glm(formula = own_telephoneBi ~ log(credit_amount), family = "binomial",
##     data = credit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6208  -0.9948  -0.7722   1.2002   1.9766
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.28536    0.71926  -8.739   <2e-16 ***
## log(credit_amount)  0.75332    0.09112   8.268   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1349.2  on 999  degrees of freedom
## Residual deviance: 1274.1  on 998  degrees of freedom
## AIC: 1278.1
##
## Number of Fisher Scoring iterations: 4
```

Interpretation: The coefficient for credit amount is positive, which is indicating that higher credit amount has a positive / increasing effect on the response variable.

Now we want to estimate the performance of the above binary model and compute fitted values / predicted values in order to compare them with the underlying, real data.

```
##    fit
## obs   0   1
##   0 505  91
##   1 261 143
```

Interpretation: When fitting the values according to the above defined model and comparing them with the confusion matrix we draw following conclusion: Predicting telephone owner according to the model results in 143 matches. This is expressed in relative terms 35.4%. Predicting clients without a telephone results in 505 matches or correctly predicted values. This is in relative terms 84.7%. It seems that the model is predicting clients without a telephone quiet well, but is performing badly in predicting telephone owners. It could be that outliers are influencing the result, even after the process of logarithmization. We can see in the above plot that there are at least two outliers, which have a high credit amount for clients owning a telephone and not owning one.

5. Support Vector Machine (SVM)

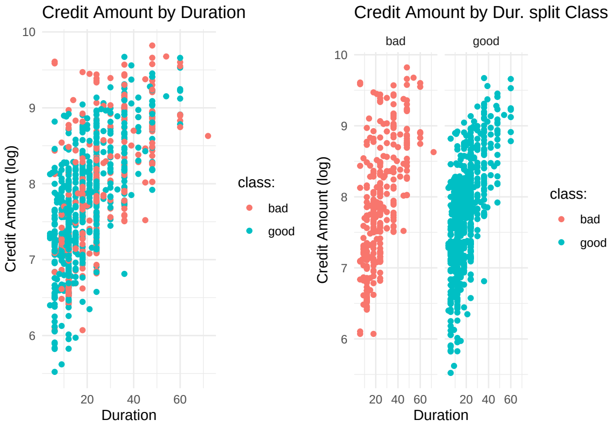
The lead / responsibility for this model has Daniel Podolecki.

Disclaimer: The results vary slightly each time the model is run, especially in terms of the calculated cost factor with the tune function. That means that the associated text does not fit. Therefore, the script was run through once and the corresponding results including graphics were inserted afterwards. This procedure was chosen only for presentation purposes and does not falsify the results in any way.

In this section we are building a support vector model. With the support vector we try to categorise and predict the clients in a class good or bad. The prediction of other categorical variables like credit history and savings status were also tested. But the prediction of class in terms of Accuracy, Sensitivity and Specificity was the most successful, which is why it is focused on here.

In a first step we are taking the category “class” and plotting the clients in good and bad to see the possible categorization visually. Additionally the installment commitment will be checked but not included in the output because it serves as slicer for later svm plots.

Visual Inspection of Class :



The following analysis will be structured in three subchapters: 5.1 Taking all variables into account / 5.2 Taking a reduced dataset, a subset containing the following variables: Credit Amount, Duration and Installment_Commitment / 5.3 Taking all variables, but each factor level of a categorical variable will represented as new column (binary variable 0 or 1). 5.3 is Omitted in the PDF - Only visible in code.

5.1 SVM - Taking all variables into account

Step 1: Data Preparation

```
## Prepare the Data for Training
set.seed(123)
indices <- createDataPartition(credit$class, p=.80, list = F)
```

In a first step we select 80% of the datapoints at random. The function takes “class” from the original dataset as an argument and is then taking an 80% share and returns an indices. The data partition is created in terms of classes. Then the data is splitted in train-data, test-data and validation-data.

```
### Create some easy Variables to access Data
train <- credit %>%
  slice(indices)
test_in <- credit %>%
  slice(-indices) %>%
  dplyr::select(-class)
test_truth <- credit %>%
  slice(-indices) %>%
  pull(class)
```

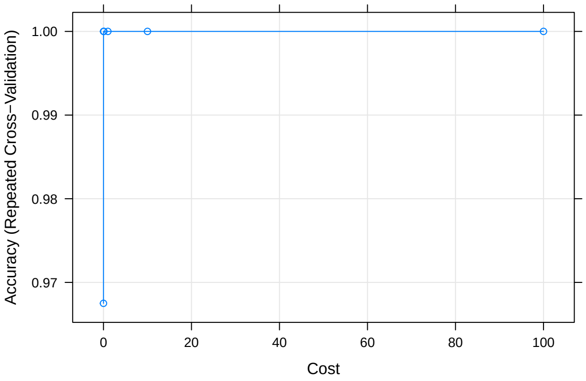
Step 2a: Tune the cost hyperparameter

```
#Radial
set.seed(123)
tuned <- tune(svm, class~., data = train,
              kernel = "radial",
              ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100)))


#(bestmod1 <- tuned$best.model)
#Result: cost= 10

#Linear
set.seed(123)
tuned_lin <- tune(svm, class~., data = train,
              kernel = "linear",
              ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100)))

#(bestmod1 <- tuned_lin$best.model)
#Result: cost = 1
```

In terms of error the tune-function is identifying 10 as the best cost-factor radial and 1 for linear kernel.

Step 2b: Tune cost using caret

```
set.seed(123)
grid <- expand.grid( C = c(0.001, 0.01, 0.1, 1, 10, 100))
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

#Linear only
svm_linear_grid_lin <- train( class ~., data = train, method = "svmLinear",
                           trControl=trctrl,
                           preProcess = c("center", "scale"),
                           tuneGrid = grid,
                           tuneLength = 10
)
plot(svm_linear_grid_lin)
```



By using the train function in the carnet library the cost factor of 1 is suggested as optimal for the linear kernel.

Step 3: Classification using e1071 - Linear or radial

Now the SVM-Model is created with the svm()-function. All the other variables besides class are taken as “predictors”, indicated with ~. The kernel leaves two choices: linear or radial. Whereas a linear kernel is classifying the data linearly, meaning with a line, the radial kernel is also taking into account non linear classification.

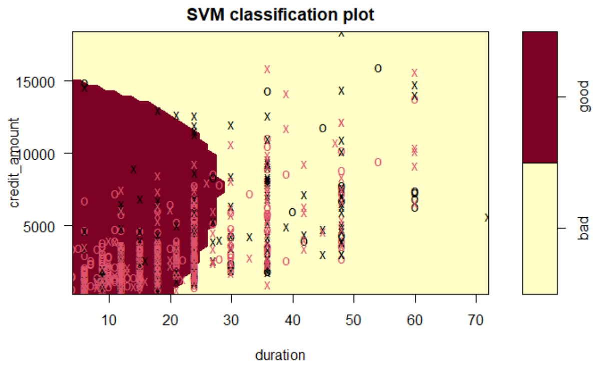
The cost factor of the tune function in step 2a will be taken into account as parameter. The higher the cost factor, the more confident we are in the separation of the classes, this means the lower the value the higher the possibility for overfitting.

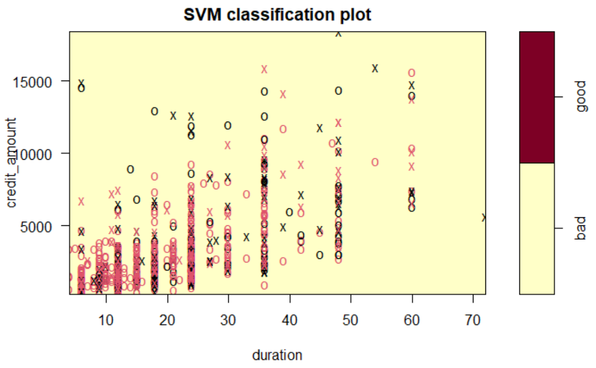
Moreover radial kernel has to be scaled, whereas the linear kernel Scale is set = False.

```r
set.seed(123)
credit_svm <- svm(class ~., train, kernel = "radial", scale = TRUE, cost =10)
set.seed(123)
credit_svm_lin <- svm(class ~., train, kernel = "linear", scale = FALSE, cost =1)
```

For the model with radial kernel 462 support vectors were identified, these are a lot of support vectors taking into account the train-data is including 800 datapoints. (more than 50 %). A large number of support vectors is often a sign of overfitting. For the linear kernel 281 are identified. This seems to be appropiate.

Step 4: Plot the Classifications - using e1071 Linear or radial





Interpretation: The interaction of duration with installment_commitment is significant as seen in previous models and therefore it is fixed on 4, because it means that the borrower has to pay an installment rate of 4 % of his disposable income. All other variables are fixed at a value where intuitively you would say it is a bad borrower for the bank.

The plot 1 shows: Some of the good clients are correctly classified as good. However, outside the red zone there are still many red data points that have been misclassified. The slicer is therefore not 100% suitable.

Analyzed on a substantive level, the plot doesn't necessarily make sense either. Intuitively, one would assume that increasing the loan amount increases the bank's risk and therefore the bank would rather consider clients who apply for a small loan amount and have a short duration to be good. Or the other alternative would be a high duration with a high loan amount. In the plot, this alternative is classified as bad, which does not necessarily make sense.

However it is not possible to slice the data on a meaningful position. Plot 2 does not show an insightful visualization. Many combinations in a loop was tested (3000 combinations). But the plots were not producing a meaningful output. That's why the idea came up to have a reduced dataset consisting of only three variables, to not have to do the slicing problem.

Step 5: Analyse model performance

Lets now predictions based on the model and pass it into test_in.

```r
test_pred <- predict(credit_svm, test_in)

test_pred_lin <- predict(credit_svm_lin, test_in)
```

The confusion matrix compares the predicted values with the data retained for testing.

```
conf_matrix <- confusionMatrix(test_pred, test_truth)
#conf_matrix

conf_matrix_lin <- confusionMatrix(test_pred_lin, test_truth)
#conf_matrix_lin
```

Interpretation: For the *radial model* we receive an Accuracy of 76,5%, a Sensitivity of 43,33 % and a Specificity of 90,7%. This model result is already quite good.

- Sensitivity means how many truth bad (60) are classified as bad in the predcition. These are 26 –> 26/ 60 = 43,3 %.
- Specificity means how many truth good (140) are classified as bad in the predcition. These are 127 –> 127 / 140= 90,7%.
- Accuracy is the overall model performance. (26+127) / 200 = 76,5%

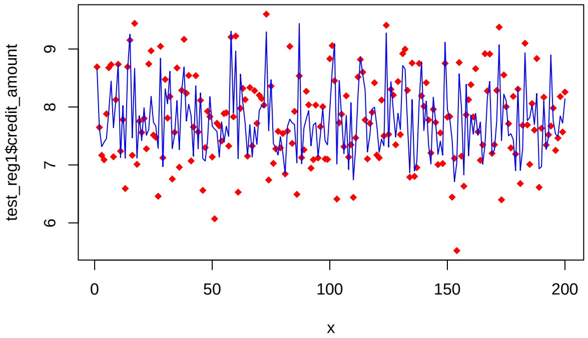
For the *linear model* we receive an Accuracy of 68,5%, a Sensitivity of 36,7 % and a Specificity of 82,1%. –> All statistical criteria are worse than in the radial model.

Step 6: Support Vector Regression

As additional steps lets now make a regression of the data and if its possible to predict the credit amount.

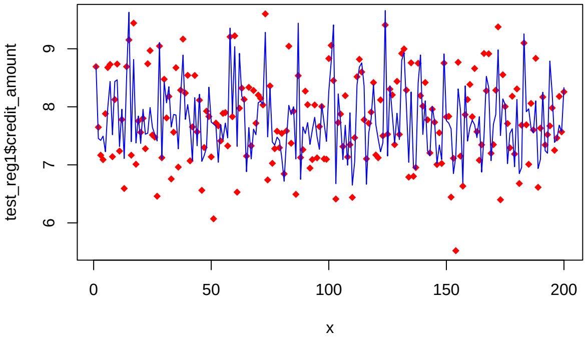
```
set.seed(123)

##Logarithmieren
credit_reg <- credit
credit_reg$credit_amount <- log(credit$credit_amount)

#SVM Regression
indexes_reg1 <- createDataPartition(credit_reg$credit_amount, p = .8, list = F)
train_reg1 <- credit_reg[indexes_reg1, ]
test_reg1 <- credit_reg[-indexes_reg1, ]
```

```
model_reg1 <- svm(credit_amount~., data=train_reg1)
#print(model_reg1)
pred <- predict(model_reg1, test_reg1)
x <- 1:length(test_reg1$credit_amount)
plot(x, test_reg1$credit_amount, pch=18, col="red")
lines(x, pred, lwd="1", col="blue")
```



```
res_svm<- sqrt(mean((test_reg1$credit_amount - pred)^2))
#exp(res_svm)
#Result = 1.58127
```

```
#Compare to OLS
model_lm1 <- lm(credit_amount~., data=train_reg1)
pred_lm1 <- predict(model_lm1, test_reg1)
plot(x, test_reg1$credit_amount, pch=18, col="red")
lines(x, pred_lm1, lwd="1", col="blue")
```



```
res_ols <- sqrt(mean((test_reg1$credit_amount - pred_lm1)^2))
#exp(res_ols)
#Result= 1.621536
```

Interpretation: Comparing SVM Regression with OLS shows: OLS has a root square mean error of 1.621536 which is a little bit higher than svm with a value of 1.58127 The nearer the root square mean error is to 0 the better. Therefore, the SVM regression is slightly more accurate than the linear regression to predict the credit amount based on duration.

5.2 SVM - Reduce dataset to a simple model

Regarding the problem in 5.1 to find a suitable slicer for the svm plots, it was decided to test a reduced model to be able to interpret the results.

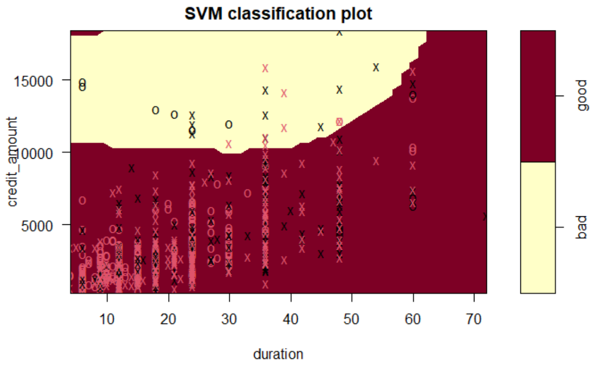
Since steps 1-3 are very similar to steps 5.1, I will refrain from further comments here and will just include the code if something specific new happens. The focus now is more on the interpretations in steps 4 and 5.

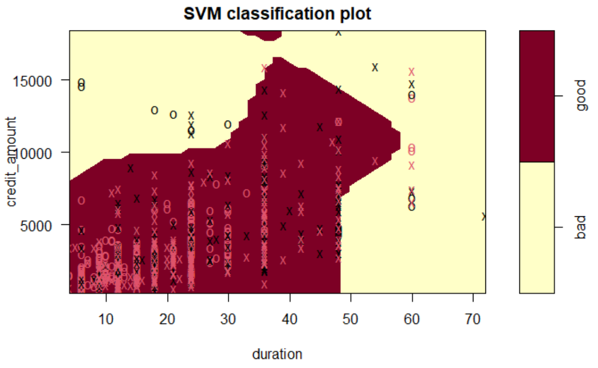
Step 1: Data Preparation

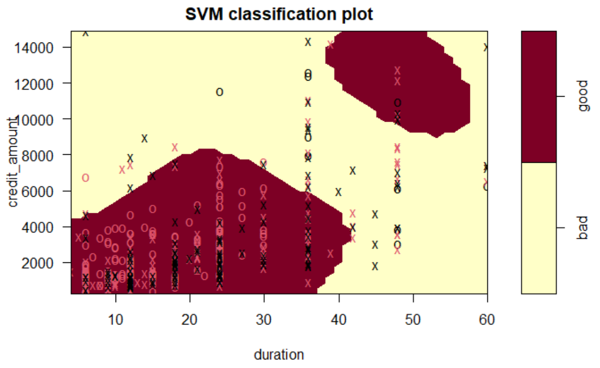
```
## Prepare the Data for Training
set.seed(123)
subset_simple<- subset(credit,select=c(credit_amount, duration,class, installment_commitment))
#All Installment Commitment values

subset_simple2 <-  subset_simple  %>%
filter(installment_commitment==1 | installment_commitment==4)
#Only Installment Commitment 1 and 4 as extreme values

indices1 <- createDataPartition(subset_simple$class, p=.80, list = F)
indices2 <- createDataPartition(subset_simple2$class, p=.80, list = F)
```

Step 4: Plot Classification using e1071 - Linear or radial

Plot 1 (top left) shows credit-amount by duration, no slicer was used and “subset_simple” as dataset is used. Plot 2 (top right) shows credit-amount by duration with slicer on variable “installment_commitment” with category 4. Plot 3 (bottom middle) shows credit-amount by duration, no slicer was used and “subset_simple2” as dataset is used.







Interpretation Plot 1, 2 and 3 fit the intuitive understanding of credits that a customer is classified as good client.

We have two alternatives: Alternative 1: The credit amount is low and the duration is short. If a loan defaults with a low amount, it does not mean a big loss for the bank. Thats why its reduces the bank’s risk and therefore the client could be classified as good.

Alternative 2: The credit amount is high and the duration is long. Since the duration is very long, the installment rate increases and thus the risk for the bank for high losses decreases. Because money has already been collected over all years via the high installment rates as compensation payments.

However there are differences if a customer with long duration and high credit amount can be classified as good or bad. Another deviation is also seen in customers, whether customers with medium duration and medium credit amount are to be classified. In order to see which model is most suitable, we will now look at the model performance.

Step 5: Analyse model performance

Interpretation - In the *radial model* with train1 we receive an Accuarcy of 70,5%, a Sensitivity of 0,33 % and a Specificity of 99,3%. –> This model is super good in predicting good customer, but extremely weak in detecting bad customers.

- In the *linear model* with train1 we receive an Accuarcy of 70 %, a Sensitivity of 0 % and a Specificity of 100%. –> This model is even in predicting a good customer, but fails completely in detecting bad customers.

- In the *radial model* with train2 we receive an Accuarcy of 69,4%, a Sensitivity of 21,0 % and a Specificity of 92%. –> Thismodel can detect a few bad customers, but the performance is still poor.
- In the *linear model* with train2 we receive an Accuarcy of 68.6 %, a Sensitivity of 0 % and a Specificity of 100%. –> This model is even in predicting a good customer, but fails completely in detecting bad customers.

Compared with the best model from 5.1 all of these models are worse. However the svm plots in 5.2 can be better interpretedbecause the slicer is not that arbitrary as in 5.1

6. Artificial Neural Network (ANN)

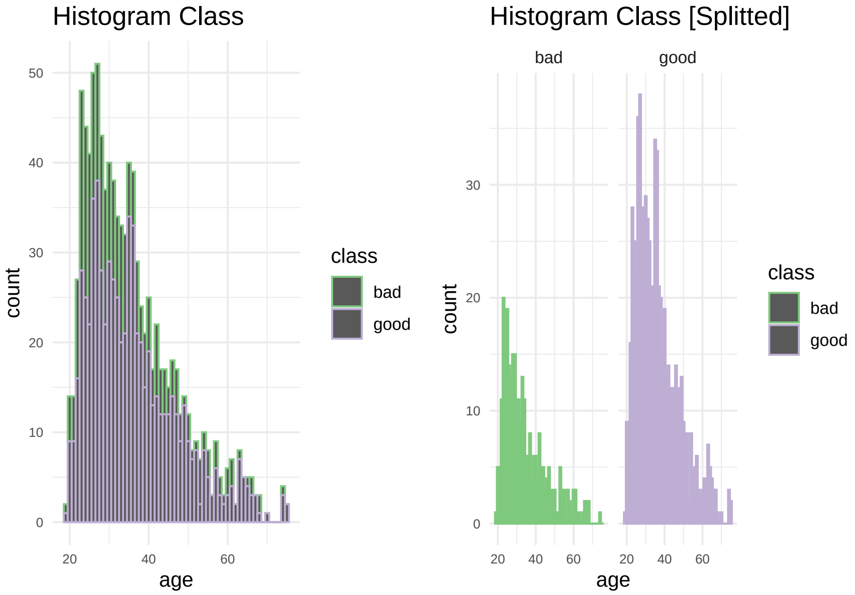
The lead / responsibility for this model has Matthias De Paolis.

Disclaimer: The results vary slightly each time the model is run, which means that the associated text does not fit. Therefore, the script was run through once and the corresponding results including graphics were inserted afterwards. This procedure was chosen only for presentation purposes and does not falsify the results in any way.

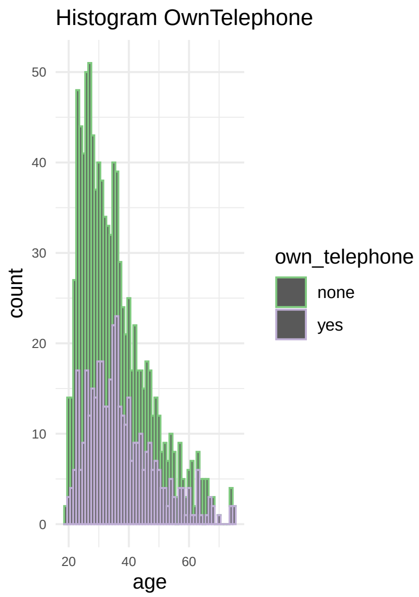
In this chapter we are building an artificial neural network. The goal is to fit a model which accurately predicts whether a client has a good or bad rating. We will also try to predict if a customer owns a telephone or not since the attribute 'own_telephone' looked promising in the analysis of the preciding chapters.

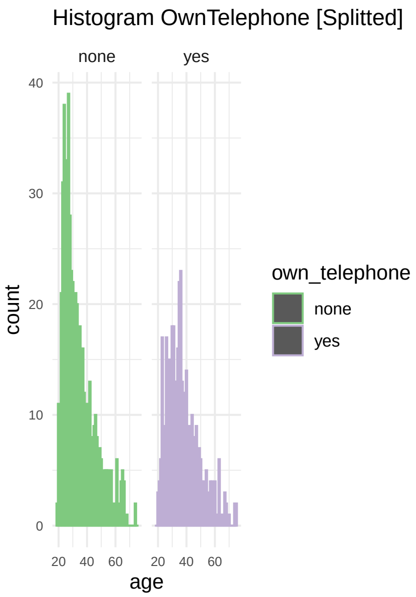
In the first step the data is displayed graphically to get an overview of the data.

Overview Distribution Age vs. Class



Overview Distribution Age vs. OwnTelephone





In the following sub chapters we are building and testing a neural network for each case and try to maximize the performance of those models.

6.1 ANN - Class Prediction

Step 1: Data Preparation

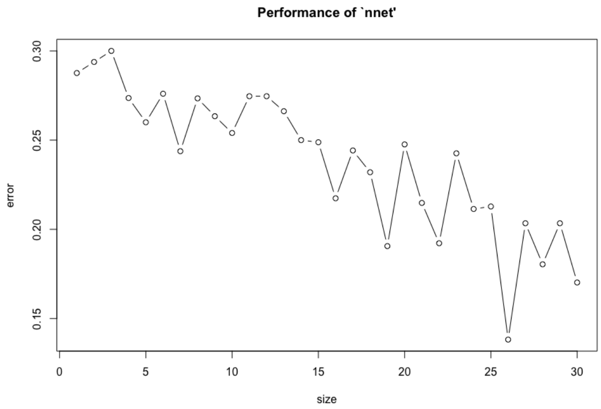
To achieve the best possible results it is advise to first shuffle the data set.

```
set.seed(123)
# shuffle the dataframe by rows
credit <- credit[sample(1:nrow(credit)),]
```

Now we can build our neural network. Our neural network should divide the data set into the class 'good' and 'bad'. The entire data set is passed to the model. The tune.nnet function automatically performs a 10x cross validation, which eliminates the need to split the data into training and test data.

Step 2: Train with nnet using Cross Validation

```
# nnet 10-fold cross-validation
tmodel <- tune.nnet(class ~ ., data = credit, size=1:30, MaxNWts = 10000)
summary(tmodel)

#best parameters: size 26
plot(tmodel)
tmodel$best.model
# a 49-26-1 network with 1351 weights
```



The tune.nnet function estimates the best possible network to be a 49-26-1 network. With this new understanding we can create and plot of our model.

Step 3: Build network with nnet

```
credit_net <- nnet(class ~ ., data = credit, size=26, maxit=100, range=0.1, decay=5e-4, MaxNWts = 10000)

credit_net
```

We build our model based on the parameters of the tune.nnet function. We use 26 units for the hidden layer. Now we can make some calculations to analyze our model performance.

Step 4: Analyze Model Performance

Make Prediction

```
    true
```

pred bad good

bad 251 162

good 49 538

The confusion matrix looks very promising. Let's check the accuracy.

Calculating the accuracy

```
sum(diag(cm_nn))/sum(sum(cm_nn))
```

The **accuracy** has a value of 78.9% which describes how the model performs across all classes.

Accuracy is a metric that generally describes how the model performs across all classes. It is calculated as the ratio between the number of correct predictions to the total number of predictions. Now let's take a look at the precision.

Calculating the precision

```
cm_nn[1, 1]/(cm_nn[1, 1] + cm_nn[1, 2])
```

The **precision** has a value of 60.7% which describes ratio between the correctly classified classes.

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

Using the confusionMatrix function

```
confusionMatrix(as.factor(pred), as.factor(credit$class))
```

Output fo the consuionMatrix function:

Confusion Matrix and Statistics

```
      Reference
```

Prediction bad good bad 251 162 good 49 538

```
           Accuracy : 0.789
             95% CI : (0.7624, 0.8139)
No Information Rate : 0.7
P-Value [Acc > NIR] : 1.396e-10

              Kappa : 0.5464
```

Mcnemar's Test P-Value : 1.254e-14

```
        Sensitivity : 0.8367
        Specificity : 0.7686
     Pos Pred Value : 0.6077
     Neg Pred Value : 0.9165
         Prevalence : 0.3000
     Detection Rate : 0.2510
```

Detection Prevalence : 0.4130
Balanced Accuracy : 0.8026

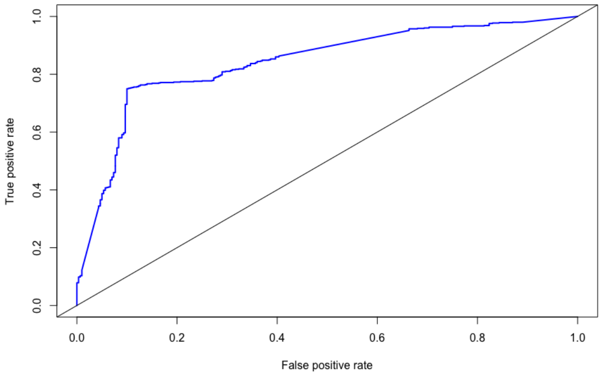
```
   'Positive' Class : bad
```

By looking at the summary of the confusionMatrix function we can again see the Accuracy of 78.9% but also the 'No Information Rate' which tells us that the larger group we classified makes 0.7 or 70% of the whole data set. We can also see that our model predicted 83.7% of the positive values correctly which by the way are class ‘bad’, and 76.9% of the negative predicted values correctly. The McNemar’s Test P-value tells us that model 1 (1st and 2nd row of the the confusion matrix) differs significantly from model 2 (1st an 2nd column of the confusion matrix). This information however does not apply to our model and can be ignored.

Next we look at the ROC Curves to visually show how good our model performs.

ROC Curve To indicate the model quality we are using the Receiver Operating Characteristic Curve (ROC). The ROC curve shows the trade-off between sensitivity (or True Positive Rate(TPR)) and specificity (1 – False Positive Rate (FPR)). Classifiers that give curves closer to the top-left corner indicate a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

```
pred_raw <- predict(credit_net, credit, type = "raw")
pred <- ROCR::prediction(pred_raw, credit$class)
perf <- ROCR::performance(pred, "tpr", "fpr")
plot(perf, lwd=2, col="blue")
abline(a=0, b=1)
```



Interpretation With a 49-26-1 network with 1351 weights we achieve:

- Accuracy of 78.9%: With an accuracy of 78.9% this model does look good.
- Sensivity of 83.7%: Rate of True Positives captured by the model (538/700)
- Specifity of 76.9%: Rate of True Negatives captured by the model (251/300)
- Pos. Pred. Value - 60.7%: Rate of Positives captured among the total Pos Predicted (251/412)

- Neg. Pred. Value - 91.7%: Rate of Negatives captured among the total Neg Predicted (538/587)

The ROC curve indicates that our model is much better than a random classifier. It seems to be a good model to classify customers according to their creditworthiness.

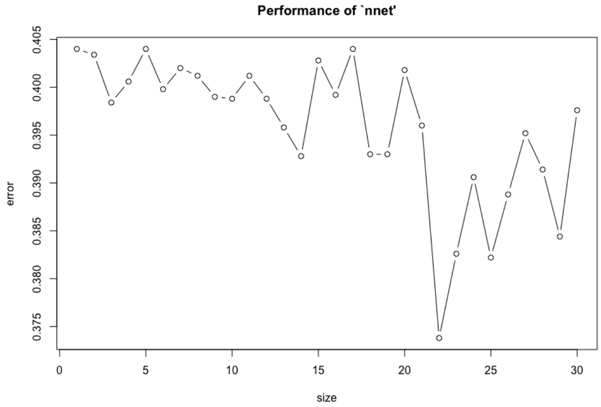
6.2 ANN - Own Telephone Prediction

Step 1: Data Preparation Since the data set has already been shuffled to improve the reliability of the model in 6.1. This step is skipped.

The neural network should divide the data set into the class ‘owntelephoneYes’ and ‘owntelephoneNone’. Like in 6.1 the entire data set is passed to the model. The 10-Cross Validation will be performed with the tune.nnet function, which eliminates the need to split the data into training and test data.

Step 2: Train with nnet using Cross Validation

```
# nnet 10-fold cross-validation
tmodel <- tune.nnet(own_telephone ~ ., data = credit, size=1:30, MaxNWts = 10000)
summary(tmodel)

#best parameters: size 22
plot(tmodel)
tmodel$best.model
# a 49-22-1 network with 1334 weights
```



The tune.nnet function estimates the best possible network to be a 49-22-1 network. With this new understanding we can create and plot our model.

Step 3: Build network with nnet

```
credit_net <- nnet(own_telephone ~ ., data = credit, size=22, maxit=100, range=0.1, decay=5e-4, MaxNWts = 10000)

credit_net
```

Now we can make some calculations to analyze our model performance.

Step 4: Analyze Model Performance

Make Prediction

```
pred <- predict(credit_net, credit, type="class")
cm_nn <- table(pred=pred, true=credit$own_telephone)
cm_nn
```

```
  true
```

pred none yes

none 393 55

yes 203 349

The confusion matrix looks promising. Let’s check the summary of the confusionMatrix

confusionMatrix function

```
confusionMatrix(as.factor(pred), as.factor(credit$own_telephone))
```

Output fo the consuionMatrix function:

Confusion Matrix and Statistics

```
      Reference
```

Prediction none yes none 393 55 yes 203 349

```
           Accuracy : 0.742
             95% CI : (0.7137, 0.7689)
No Information Rate : 0.596
P-Value [Acc > NIR] : < 2.2e-16

              Kappa : 0.4941
```

Mcnemar’s Test P-Value : < 2.2e-16

```
        Sensitivity : 0.6594
        Specificity : 0.8639
     Pos Pred Value : 0.8772
     Neg Pred Value : 0.6322
         Prevalence : 0.5960
     Detection Rate : 0.3930
```

Detection Prevalence : 0.4480
Balanced Accuracy : 0.7616

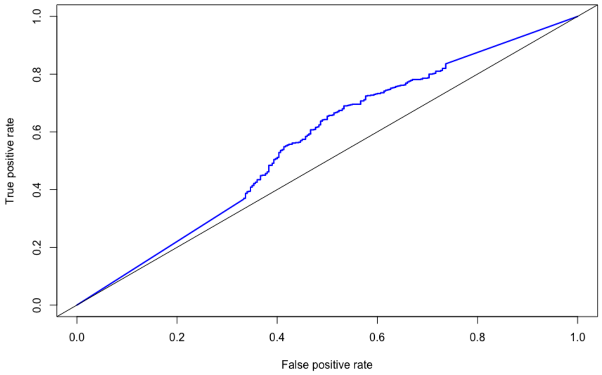
```
   'Positive' Class : none
```

By looking at the summary of the confusionMatrix function we can again see the Accuracy of 74.2%. We can also see that our model predicted 87.7% of the positive values correctly which by the way are class ‘bad’, and 63.2% of the negative predicted values correctly.

With the ROC Curves we can visually show how good our model performs.

ROC Curve To indicate the model quality we are using the Receiver Operating Characteristic Curve (ROC). For this we plot sensitivity against 1-specificity

```
pred_raw <- predict(credit_net, credit, type = "raw")
pred <- ROCR::prediction(pred_raw, credit$class)
perf <- ROCR::performance(pred, "tpr", "fpr")
plot(perf, lwd=2, col="blue")
abline(a=0, b=1)
```



Interpretation With a 49-22-1 network with 1334 weights we achieve:

- Accuracy of 74.9%: With an accuracy of 74.9% this model is quite good.
- Sensitivity of 65.9%: Rate of True Positives captured by the model (393/596)
- Specifity of 86.4%: Rate of True Negatives captured by the model (349/404)
- Pos. Pred. Value of 87.7%: Rate of Positives captured among the total Pos Predicted (393/448)
- Neg. Pred. Value of 63.2%: Rate of Negatives captured among the total Neg Predicted (349/552)

In our case the ROC curve does not behave better than a random classifier. Therefore, we can acknowledge that our model is not suited to classify customer based on the fact if they own a telephone or not.

7. Summary

Linear Model:

When trying to predict the credit-amount, applying backwards selection and including all the important variables suggested yields in an adjusted r-squared of 63.9%. To obtain this result 12 explanatory variables are required from the dataset. The bank is now provided with a much simpler suggestion to compute a linear model. By only using credit-duration, installment-commitment and the job-status as explanatory variables, we are able to obtain a similar result in terms of adjusted r-squared (58.3%), while reducing the complexity drastically. This enables the bank to make a prediction for the credit-amount based on few information. On a daily basis the linear model could indicate whether the credit-amount a client would like to receive is significantly different from the average credit-amount with same prerequisites. The bank is able to adjust the credit-request also based on qualitative information which the model does not compute such as: prior credits, age and many more.

However, depending on the use-case of the linear model an explanation of the variance in the variable credit-amount of 58.3% might not be enough to obtain an accurate estimation on whether the credit requirements are feasible for an individual client. Other factors have to be taken into account as well.

GLM Count Data:

All categorical variables were compared with the count data. For this purpose, the poisson distribution was used in order to model models that were as realistic as possible. Most combinations yielded no insights. However, significant relationships were found between the Count Data Variable 'Duration' and the variables OwnTelephone, Property Magnitude and Class. These were described in Chapter 4.1 Count Data.

The analysis of count data can provide a bank with indications as to which relevant attributes are important in the event that, for example, no further information about the customer is available.

GLM Binomial:

The best model performance were achieved with the input variable "own telephone". In a logistic regression we have seen that that owning a telephone leads to an increasing credit amount. In general the model was strong in predicting if a person does not have telephone. (84,7 % were predicted right). However it is bad in predicting that a person have a telephone (only 35,4 % were predicted right).

As final advice to the bank we would suggest to ignore using GLM - Binomial for their predictions. Although owning a telephone achieves good results, it is difficult to apply the results in practice considering the proportion of owning a telephone in the total population of Germany.

SVM:

The prediction of the qualitative variable "Class" was the most accurate. Additionally a Support Vector Regression was conducted to predict the credit amount. The root mean squared error was a bit lower than for the linear regression, but very close to its value.

In general, SVM is strong in predicting the class for good bank customers, but not so strong in predicting bad bank customers. It can be concluded, that the best model contains all variables and uses a radial kernel. The decision for a slicer to produce meaning plots is very hard. It is possible to interpret the plots with a simple model. However, when a simple model based on a reduced dataset is applied, the model performance decreases.

The interpretation for the simple model consists of two approaches: 1) A classified good customer has a low credit amount and a short credit duration. If a loan defaults with a low amount, it does not mean a big loss for the bank. Thats why its reduces the bank's risk and therefore the client was classified as good.

The second approach is not that clear visible in all plots: 2) The credit amount is high and the duration is long. Since the duration is very long, the installment rate increases and thus the risk for the bank for high losses decreases. Because money has already been collected over all years via the high installment rates as compensation payments.

As final advice to the bank I would always use both models (complex and simple). Since SVM was bad in predicting bad classified customers has always to be checked manually by a bank employee. The Specificity is really high. Therefore a bank employee can trust the model very well when a customer is classified as good, and dont need to double check it manually.

Neural Network:

By applying the neural network to the variable class and ownTelephone, all available variables were used in each case. Random shuffling of the dataset and cross-validation ensured that the result was as general as possible. Thanks to the tune.nnet function, the optimal number of units in the hidden layer could be determined. Although both models gave good results according to the confusion matrix, the application of the ROC curve showed that the model for OwnTelephone was not a good model to determine if a customer has a telephone or not, while the model for the variable Class gave a good prediction for creditworthiness.

Nonetheless, the neural network model for classify creditworthiness is a reliable tool for banks to predict the the creditworthiness of their clients, provided that the necessary information about the customers is known.

Figure 2: Symbolic picture of a bat(bad) customer