Predictive Analytics: Credit Risk Scorecard Application Case Study: Group 15

Jacob Heye Hilbrands (12229285) Mustafa Alsudani (1214099)
Moritz Renkin (11807211) Nils Klüwer (12229263)

November, 2022

Abstract

Credit scorecards are crucial tools in the credit assessment process. They are based on the prior performance of clients that share the same traits as a new client. Consequently, the goal of a credit scorecard is to estimate risk because credit scorecards are based on the past behavior of clients that share the same qualities as a prospective client. Therefore, the primary purpose of it is to either approve or reject a new client's loan request. The scorecard's function is to support this option. In this assignment, we'll compare and contrast the binning techniques known as weight-of-evidence (woe) binning and group binning. That's why we have been using the existing scorecard. Moreover, certain model changes and binning splits will be made as a result of the addition of the 7 variables. To evaluate the effectiveness of the scorecard, the Gini coefficient is used. The major results of this research were to: demonstrate that group binning outperforms woe binning in out-of-sample predictions as measured by the Gini coefficient. The dependent variable, credibility, may be predicted more precisely when more independent variables are included.

Contents

Abstract

Credit scorecards are crucial tools in the credit assessment process. They are based on the prior performance of clients that share the same traits as a new client. Consequently, the goal of a credit scorecard is to estimate risk because credit scorecards are based on the past behavior of clients that share the same qualities as a prospective client. Therefore, the primary purpose of it is to either approve or reject a new client's loan request. The scorecard's function is to support this option. In this assignment, we'll compare and contrast the binning techniques known as weight-of-evidence (woe) binning and group binning. That's why we have been using the existing scorecard. Moreover, certain model changes and binning splits will be made as a result of the addition of the 7 variables. To evaluate the effectiveness of the scorecard, the Gini coefficient is used. The major results of this research were to: demonstrate that group binning outperforms woe binning in out-of-sample predictions as measured by the Gini coefficient. The dependent variable, credibility, may be predicted more precisely when more independent variables are included.

1 Introduction

In the form of a score and a likelihood of default, credit scoring models and scorecards estimate the risk that a borrower won't return a loan.

For instance, a credit scorecard may award a borrower points based on the following table for their age and income. As previously noted, a single dependent variable, credibility, was predicted using seven independent variables. As a binning strategy, the Weight-of-Evidence technique was adopted.

Therefore, group binning will be used as an extra binning strategy for this project's execution. To increase the model's Gini coefficient of prediction, a seventh independent variable was added, and the binning breaks were modified.

2 Predictive Analytics Research Methodology

2.1 Predictor Variable Transformation

Transforming predictor variables with the Weight of Evidence (WOE) approach. The WOE describes how well an independent variable may predict the outcome of a dependent variable.

The WOE cannot be determined until the data has been separated into bins. This value is used for future calculations in place of the original value once the WOE has been located. As a result, upon transformation, the WOE will be the same for all lines that include the variable in the same bin.

If the WOE is greater than one, then the Distribution of the Goods is greater than the Bads, if the WOE is smaller than one, then the Distribution of the Goods is less than the Distribution of the Bads.

There are benefits to this transformation: Because the data is categorized with the simple treatment of the missing values, outliers are no longer a concern. Additionally, it supports categorical and continuous numbers.

Using the two functions in R:

woebin: This function determines the best binning.

woebin_ply: This function applies the provided binning information to convert the values in the original data to WOE values.

There will be some drawbacks for WOE as well. One is that binning may result in information loss. Another is the possibility of an unresearched link between the independent variables.

$$WOE = \frac{Distribution of Goods}{Distribution of Bads} \tag{1}$$

2.2 Logistic Regression Analysis

Logistic Regression (also known as Logit-Model) is a statistical Model that can estimate probability of a certain event happening based on one or more independent variables. Its application is widespread in various statistical methods, especially in classification problems and prediction analyses. Contrary to linear regression, the predicted variable in logistic regression is a Bernoulli variable, i.e. a binary random variable k with:

$$k \in \{0, 1\} \tag{2}$$

Formally, the logistic regression model estimates probability p of the Bernoulli k being 1, corresponding the the event in question happening. The logistic function defining $p(x)$ takes on the form:

$$p(k) = \frac{e^{\beta_0 + \beta_1 k}}{1 + e^{\beta_0 + \beta_1 k}} = \frac{1}{1 + e^{-}(\beta_0 + \beta_1 k)} \tag{3}$$

Figure TODO depicts the logistic regression for two examplary attributes.

In the domain of Credit Scoring, logistic regression is one of the most widely used statistical models. (Bolton u. a., 2009, p. 19)

2.3 Building Scorecards

lorem impsum

2.4 Forecasting Scorepoints and default Probabilities

lorem impsum

2.5 Forecasting Accuracy Testing

Include test-train split

3 Empirical results

lorem impsum

3.1 Loading and Preparing the Data

The dataset “germancredit” is used from the library “scorecard”. Below the import process is shown. The “germancredit” data has 21 variables, 7 will remain for our scorecard model and 1 dependent variable the “creditability”. In the following the explained theoretical approach is used to filter out the 7 most significant predictor variables, relevant for the credibility of a possible customer.

```r
# Importing of library and data
library(scorecard)
data("germancredit")
ncol(germancredit)
```

The first step of finding relevant predictor variables is done with the “var_filter()” function. The default limits and rates for iv_limit, missing_rate & identical_rate mentioned below are used. The iv_limit excludes every variables whose information value is lower or equal to 0.02 in respect to our depedent variable “creditability”. Through this filtering process 7 predictor variables are already excluded which are not eligible for the scorecard model, 13 are remaining.

```
# Filtering of variables with iv_limit >= 0.02, missing_rate <= 0.95
# and identical_rate <= 0.95
data_f.df = var_filter(germancredit, y="creditability")
ncol(data_f.df)
```

3.2 Splitting the Data into Train and Test Samples

The remaining 13 possible predictor varibales are splitted with the split_df function into train and test data with a ratio fo 75% train and 25% test data. After that data_f.list is reformated to be useable in the later process.

(Wozu gehört das)?? It is essential to draw conclusions from historical data and use the information gained to make precise predictions about credibility in order to have a scorecard that can be relied upon. As a result, we must divide the data into two sets that we loaded and filtered in the earlier phases. The sets consist of a training set and a test set, with the training set accounting for 75% and the test set for 25% of the total dataset.

```
# Splitting data into train and test data with ratio 0.75
data_f.list = split_df(data_f.df,"creditability",ratio=c(0.75,0.25))
class(data_f.list)
lapply(data_f.list,class)
lapply(data_f.list, dim)
```

Additional we generate a list for repsonse vaibles for later In-Sample and Out-of-Sample testing.

(Wo soll das hin) The model can make a forecast that is as accurate as feasible since there is enough training data available. To determine if the learning process was sufficiently effective, the test set is employed.

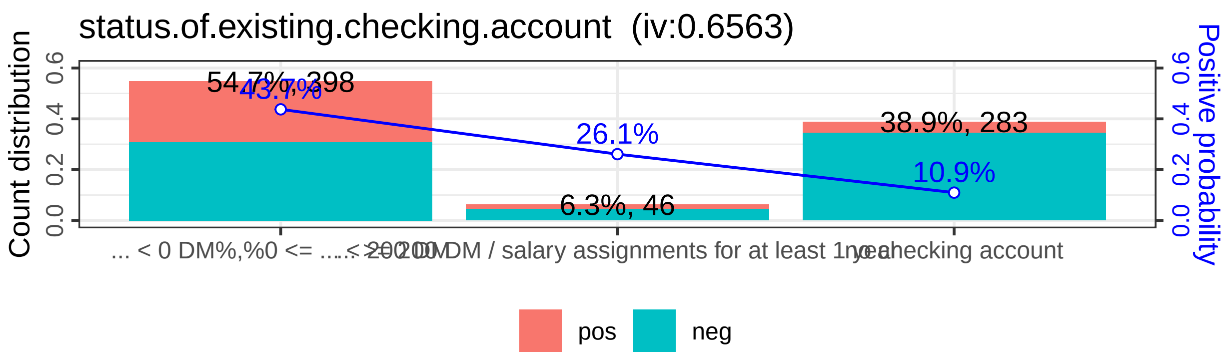
3.3 Weight-Of-Evidence (WOE)-Binning

In the following the WOE-Binning is done with the woebin() and woebin_ply function. The breaks are generated automated by the function and are saved in the “breaks.list”. Below the 13 predictor varibales are shown with their bins and information values.

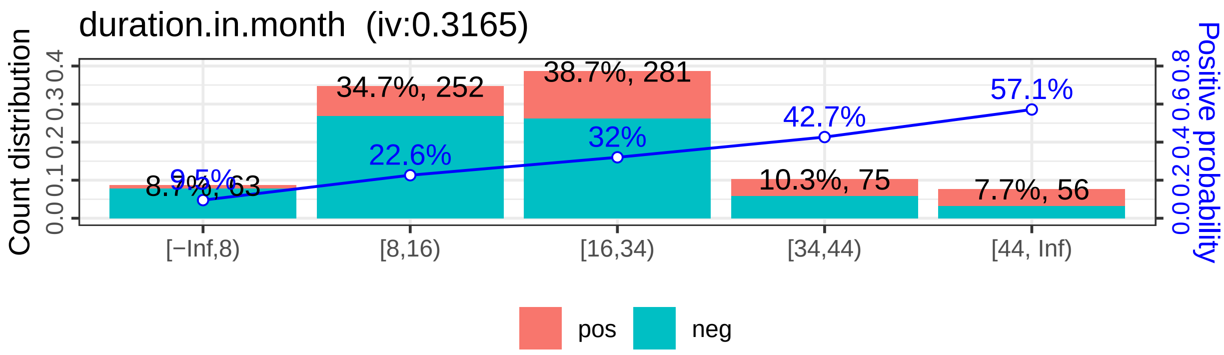
- WOE-Binning of predictor variables (used method: “width”):

```
# Binning of train data
# breaks.list is saved and imported seperately
bins.list = woebin(data_f.list$train,
                   "creditability",
                   save_breaks_list = "breaks.list")
```

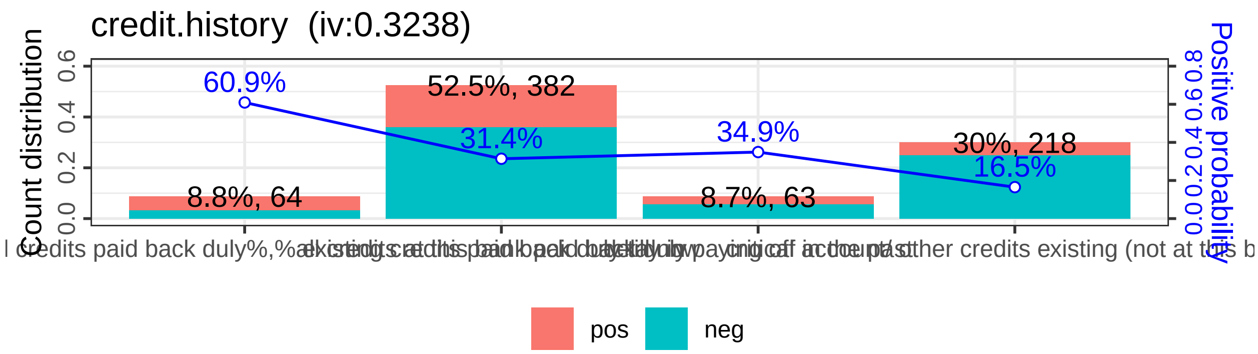
```
## $status.of.existing.checking.account
```



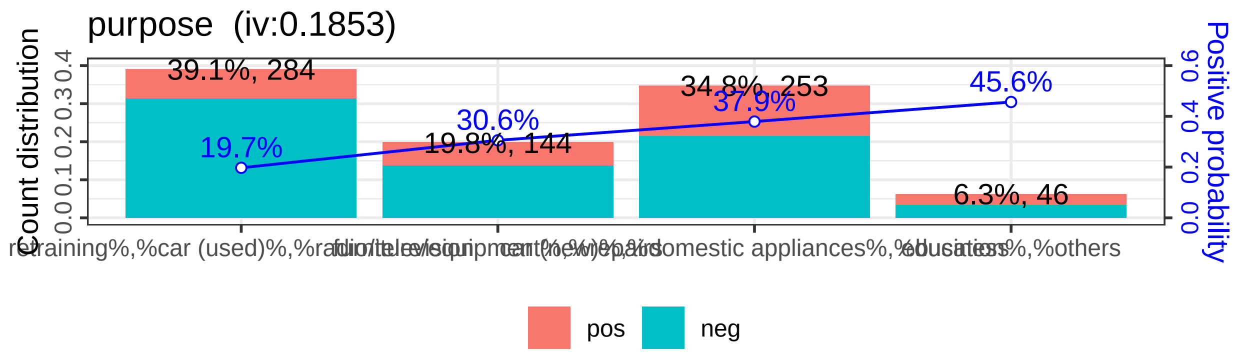

```
##
## $duration.in.month
```



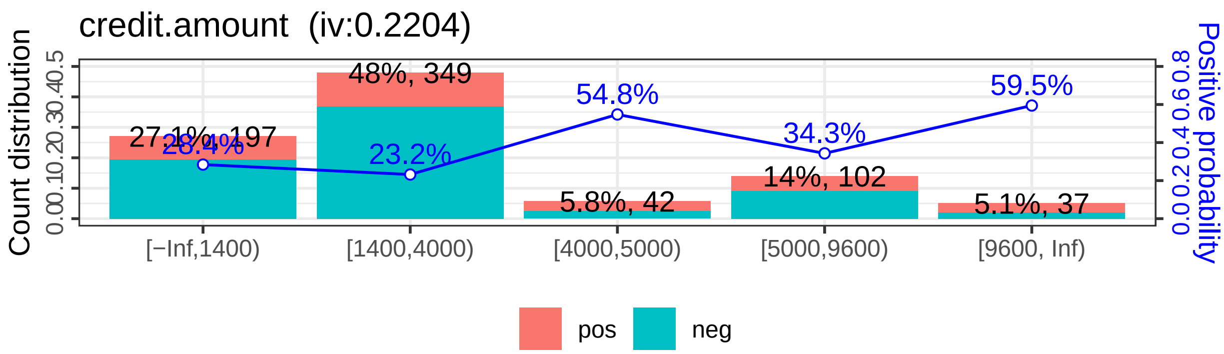

```
##
## $credit.history
```




```
##
## $purpose
```

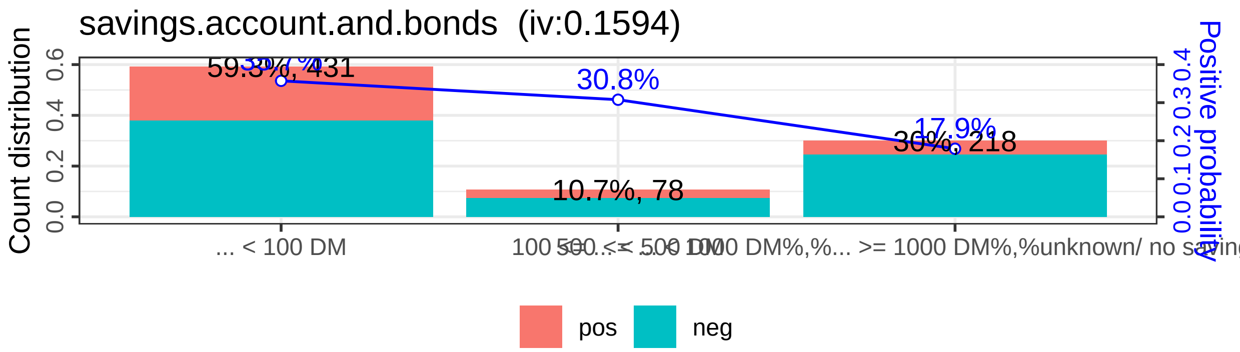



```
##
## $credit.amount
```

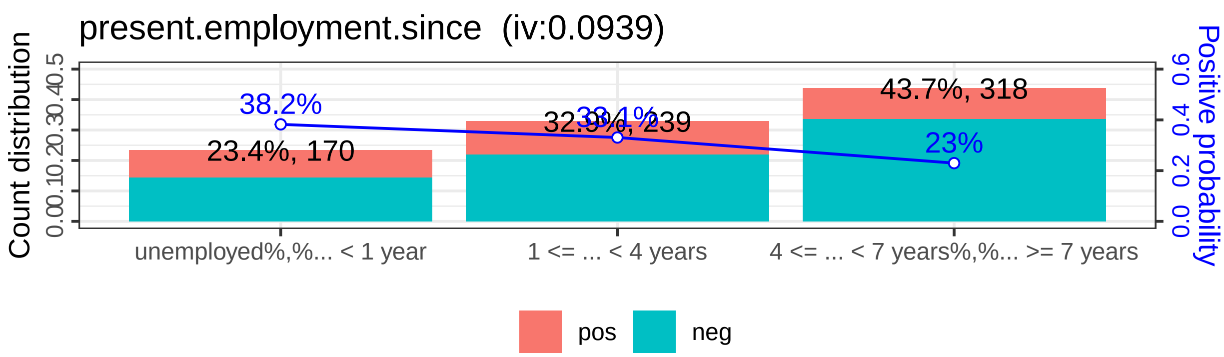



```
##
```

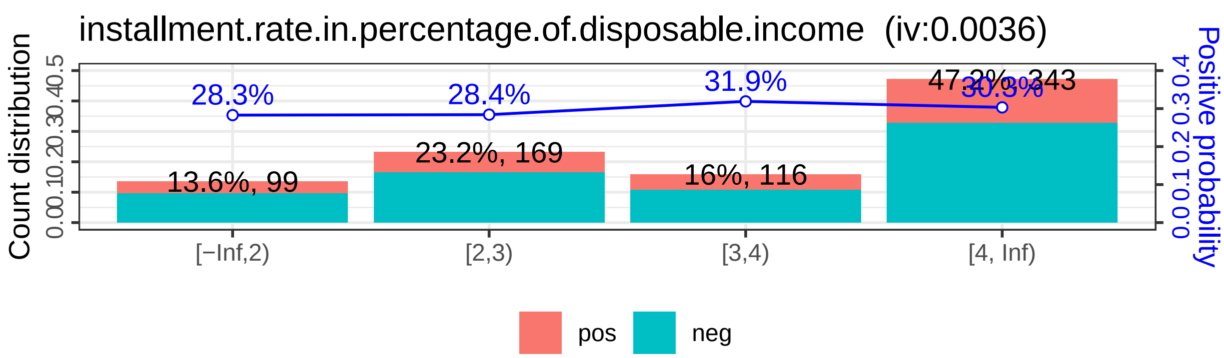
```
## $savings.account.and.bonds
```



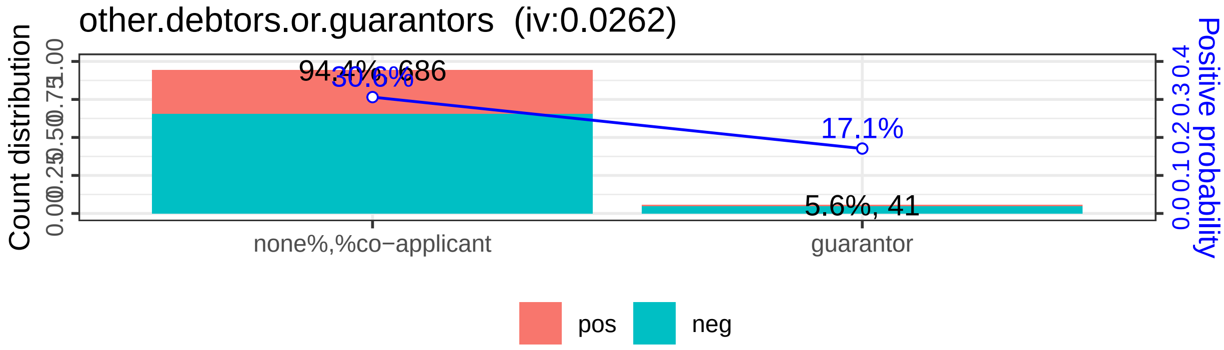

```
##
## $present.employment.since
```



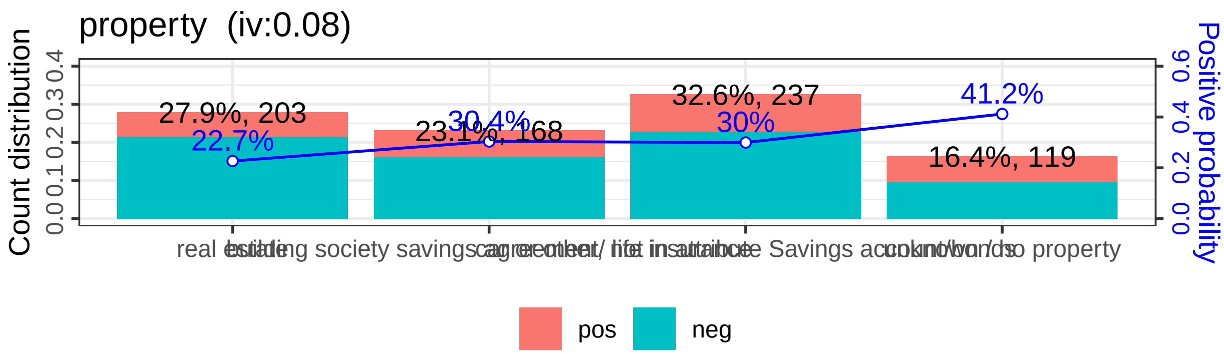

```
##
## $installment.rate.in.percentage.of.disposable.income
```



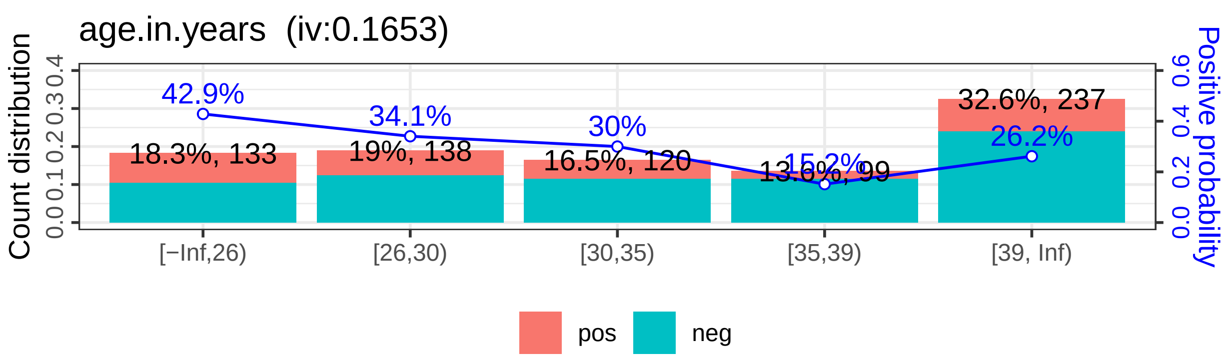

```
##
## $other.debtors.or.guarantors
```



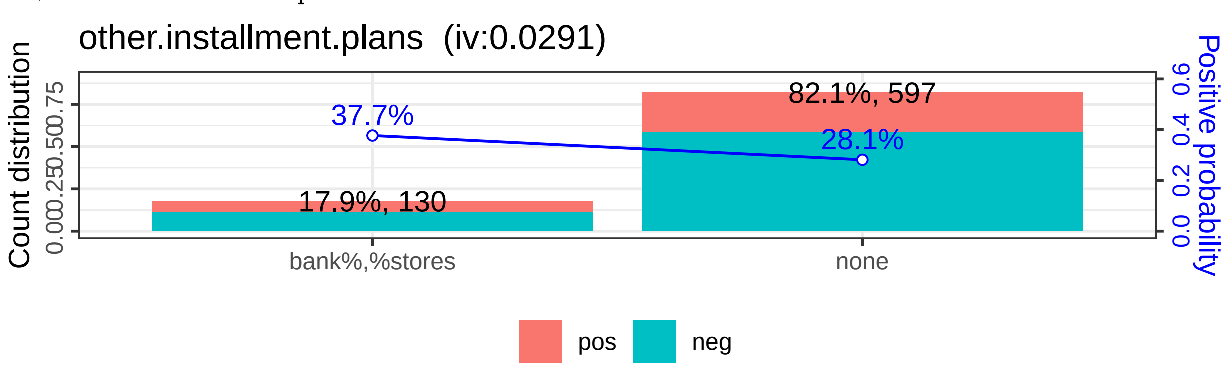

```
##
## $property
```




```
##
## $age.in.years
```

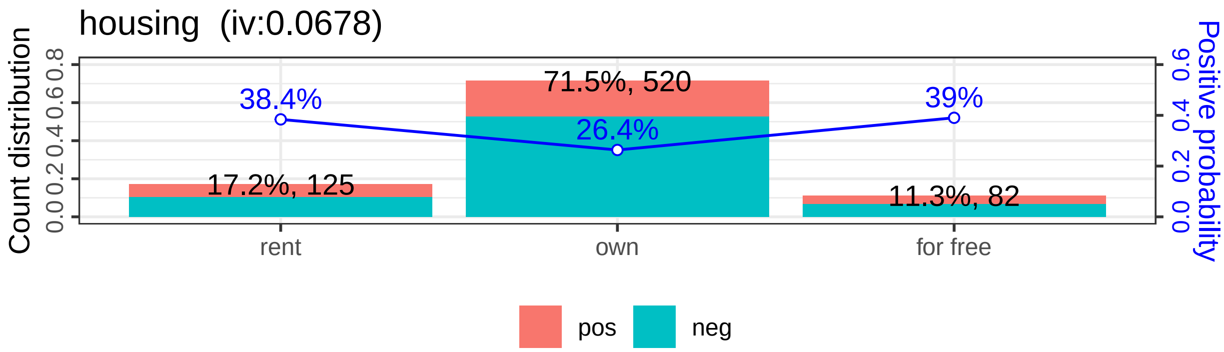



```
##
## $other.installment.plans
```




```
##
```

```
## $housing
```



After the WOE-Binning, the data is again separated into training and test data with a ratio of Transformation of predictor variables WOE and GRP:

```
# WOE-Transformation of train and test data
data_woe.list = lapply(data_f.list,
                       function(x) woebin_ply(x, bins.list))
lapply(data_woe.list, class)
lapply(data_woe.list, dim)

# Bin-Group (GRP) Transformation of train and test data
data_grp.list = lapply(data_f.list,
                       function(x) woebin_ply(x, bins.list, to = 'bin'))
lapply(data_grp.list, class)
lapply(data_grp.list, dim)
```

3.4 Generalized linear model (glm): Regressing response w.r.t. predictors

```
data_woe_first_iteration.glm <- glm(creditability ~ .,
                                    family = binomial(),
                                    data = data_woe.list$train)
```

3.5 Building the scorecard-model

3.5.1 Choosing 7 Variables with lowest Significance Value

```
data_grp_first_iteration.glm <- glm(creditability ~ .,
                                    family = binomial(),
                                    data = data_grp.list$train)
```

3.5.2 Creating a Credit Risk Scorecard with filtered 7 predictor variables

```
scorecard_woe_second_iteration.scm <- scorecard(bins.list,
                                      data_woe_second_iteration.glm)

score_woe_second_iteration.df = scorecard_ply(germancredit,
                                   scorecard_woe_second_iteration.scm, only_total_score = FALSE)

score_woe_second_iteration.list <- lapply(data_f.list,
                                      function(x) scorecard_ply(x, scorecard_woe_second_iteration.scm))
```

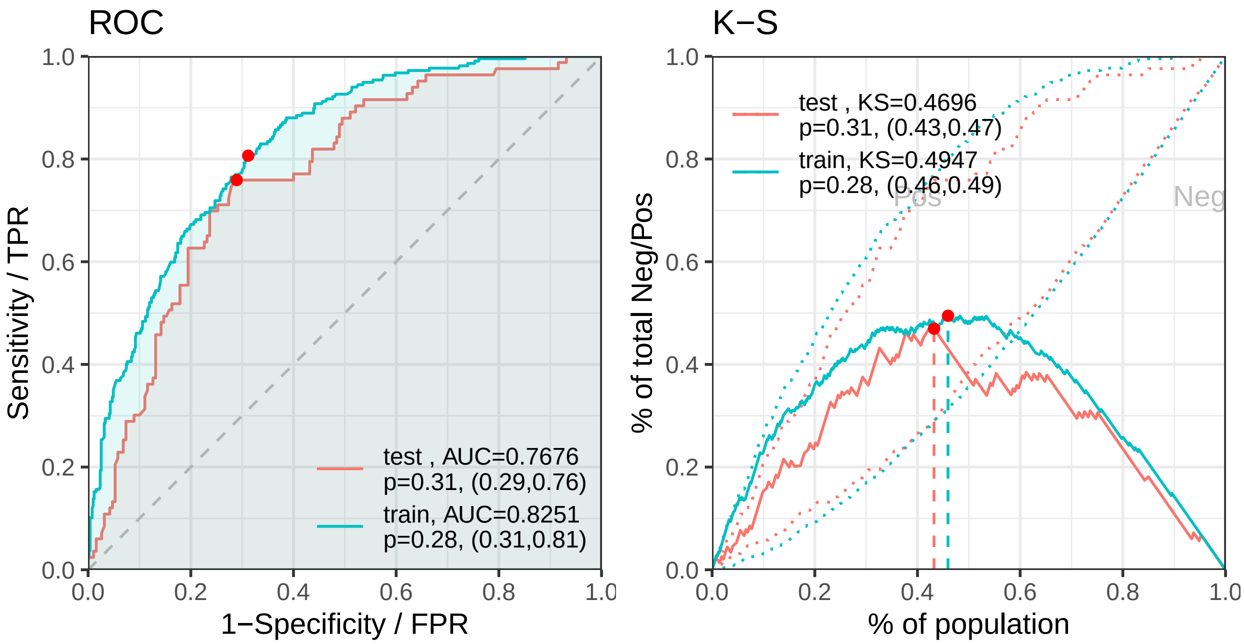
3.5.3 Calculating scorepoints for the splitted sample (train and test)

lorem ipsum

3.6 Predicting (forecasting) probabilities and scorepoints

todo

3.7 Gini Coefficient In-Sample and Out-of-Sample




```
## $binomial_metric
## $binomial_metric$train
##          Gini       AUC        R2      RMSE
## 1: 0.6502033 0.8251017 0.2775988 0.3889285
##
## $binomial_metric$test
##          Gini       AUC        R2      RMSE
## 1: 0.5352568 0.7676284 0.1618554 0.4211266
##
##
## $confusion_matrix
## $confusion_matrix$train
##    label pred_0 pred_1     error
## 1:     0    351    159 0.3117647
## 2:     1     42    175 0.1935484
## 3: total    393    334 0.2764787
##
## $confusion_matrix$test
##    label pred_0 pred_1     error
## 1:     0    128     62 0.3263158
## 2:     1     20     63 0.2409639
## 3: total    148    125 0.3003663
##
##
```

```
## $pic
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

4 Summary

TODO

References

[Bolton u. a. 2009] Bolton, C. ; Mathematics, University of Pretoria. Department o. ; Mathematics, Applied: *Logistic Regression and Its Application in Credit Scoring*. University of Pretoria, 2009 https://books.google.at/books?id=K7B3MwEACAAJ