Credit Risk Scorecard

Group 02

Lukas Arnhold, 11709460 Michael Schmiedmayer, 01610291
Laura Maria Vigl, 51834603 Tobias Schwab, 51833439

Nov 28, 2022

Contents

1 Abstract

In this paper we implement a credit-risk scorecard for assessing the credit risk based on characteristics of a customer. For this we use a logistic-regression-credit-scorecard-model based on 7 chosen variables. This model is created two times once using Group (GRP)-binning and once with Weight-of-Evidence (WoE)-binning. The training of the model is done by using the different methods of pre-processing the historical data. The models are then validated using the in- and out-of-sample Gini coefficient.

2 Introduction

As an increase of computer integration in the financial sector, financial institutions like banks use them more and more in their day-to-day operation such as decision support if it comes to loan approval of an applicant.

Using computers one can create more complex models, that can be practically implemented, than before. And one type of these more complex model is a scorecard.

In this paper, we will have a look at how the development process of such a scorecard model can be done.

First we will describe the methodology and the concept we use. Therefore, we describe the process we are going to use to transform our predictor values. Then the logistic regression and finally the concept of how to build a scorecard and how to test its accuracy in the context of its forecasting abilities.

The second part is the practical one. Here we develop a credit risk scorecard by using R and the library scorecard. We try to maximize the out-of-sample gini-coefficient and will only be using 7 variable to create the scorecard-model. However, we will create two different models: one using WoE-binning and the other one GRP-binning.

3 Methodology

3.1 Predictor variable transformation

For transforming the predictor values we use two different types. The first one being WoE-binning and the second one being GRP.

First we look at the meaning of binning. Binning is a classification process. In general this means that the values are transformed from individual values to a number of so called bins. For example this could mean that instead of the duration in months we put it into bins which are either long-term, middle-term or short-term loan.

The next question we need to answer is how to choose the different bins for each variable. For this we will use the tree based and the chimerge approach. These two are both implemented in the scorecard library in R. We then compare both binning methods and chose the one that has the better results in the Akaike information criterion(AIC).

Now that we know what binning is and which method we are going to use, we can look at two different transformation types.

The first transformation we will look at is the transformation into GRP-values, which is the more basic transformation we are using in the paper. Within the transformation the value of variables take on the value of the corresponding bin. Non-numerical values are first encoded, which is done automatically by the tree based and the chimerge approach.

The second transformation is using the WoE binning approach, which can be calculated with the following formula (Persson (2021))

$$WoE = ln\left(\frac{\frac{n_{def} \text{ of variable in bin}}{\text{total } n_{def} \text{ of variable}}}{\frac{n_{non-def} \text{ of variable in bin}}{\text{total } n_{non-def} \text{of variable}}}\right)$$

With the formula we can decide if a value in the bin supports (WoE > 0) or if it does not support (WoE < 0) as an outcome, which in our case is the customers probability of default.

3.2 Logistic regression analysis

Logistic regression is about the effects of different explanatory variables on the response variable. In principle, logistic regression is very similar to linear regression. The difference lies in the binomial response variables. In addition, continuous explanatory variables can also be used.

If several explanatory variables are considered independently of each other, the covariance between the variables is neglected and hidden effects occur.

In logistic regression, the probability of an outcome is modeled as a function of individual characteristics. Since the probability is a ratio, the logarithm of the probability is modeled. This can be represented by the following formula

$$log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)$$

π indicates the probability of an event. β_i represent the regression coefficients associated with the reference group and the explanatory variables x_i. (Sperandei (2014))

3.3 Building scorecards

Regarding the history of scorecards, it can be said that they used to be literally a simple sheet of paper on which a statistically based distribution of points was printed to be added up by the lender. Based on an opportunity-based risk prediction, the earlier scorecards served as a tool to determine whether a loan should be granted to a potential applicant. Today, paper cards no longer exist, as scorecards are now embedded in sophisticated software packages and computer interfaces as the introduction of a formulaic credit scorecard can facilitate time-consuming manual reviews and ensure more consistent credit decisions (Poon (2007)). In summary, it can be said that the purpose of a scorecard is to support the decision whether to accept an application from a new customer for credit. In other words, the credit scoring scorecard is used to assess the risk of whether a prospective customer would default.

To create the scorecard, the following procedure was followed:

To avoid distortions, the first step is to delete all cells that do not contain any values. Then the values are bound. Thus, a numeric characteristic is converted into a categorical characteristic. Categorical features are regrouped and consolidated (e.g. age grouping). In the next step, the WoE is calculated. In the second calculation step, the information value (IV) comes into focus. This is used to assess the overall predictive power of a feature. The logistic regression model is used to model the scoring function and estimate the credit score. The regression coefficient is used for scaling (fitting to specific ranges of values). Finally, the final credit score is a simple sum of the individual score values taken from the scorecard.

3.4 Forecasting scorepoints and default probabilities

The probability of default of a customer can be calculated by forecasting the scorepoints (= probabilities of default from all samples of the scorecard). The prediction or forecast of the scorepoints results from training the model using historical data (split into training and test data to check accuracy). This ensures that the parameters are set correctly.

3.5 Forecasting accuracy testing

Among the most common variants of scorecard performance measurement are the calculation of the Gini coefficient on the one hand and the mean difference on the other. Both methods are based on the same principle.

The Lorenz diagram is used to explain the Gini coefficient. In this diagram, the horizontal axis represents the proportion of goods rejected $Pr_b(s)$ from the scorecard for each score $s \in [0, 1]$ included in the data collection. The vertical axis represents the proportion of goods $Pr_g(s)$ for which 0 is the lowest score and 1 is the highest possible score. The score here is the predicted value generated by the scorecard.

The curve below the diagonal line represents a set of proportions $Pr_b(s)$ and $Pr_g(s)$ for each score generated by the scorecard and is called Receiver Operating Characteristic (ROC). The better the scorecard, the greater the difference between the proportions of good and bad for each score. The farther the ROC curve is from the diagonal, the better the scorecard (Peussa (2016)).

The Gini coefficient is defined as the following:

$$gini = \frac{DR}{0.5} = 2 * (0.5 - AR) = 1 - 2 * AR$$

$$gini = 1 - \sum_{s=0}^{1} Pr_b(s) * [Pr_g(s-1) + Pr_g(s)]$$

To sum it up, the result of gini can range from 0 to 1. A perfect scorecard would have a value of 1 and a weak scorecard would have a value close to 0.

4 Empirical Result

Loading library

```
library(scorecard)
```

The variables are distinguished among predictor (feature, independent) variables and response (label, dependent) variables.

One response variable: creditability.

The var_filter function drops column variables that do not meet the thresholds for missing rate ($> 95\%$ by default), information value (< 0.02 by default), or identical value rate ($> 95\%$ by default).

When building scorecard models, a subset of the observations should be held out from the data used to train the model (similar to most other traditional modeling approaches), and instead be apportioned to the test set. We can perform this sampling to create the train and test datasets using the split_df function.

```
data("germancredit")
dt_f <- var_filter(germancredit, y = "creditability")
```

```
## v Variable filtering on 1000 rows and 20 columns in 00:00:00
## v 7 variables are removed
```

```
dt_list <- split_df(dt_f, y = "creditability", ratios = c(0.75, 0.25))
```

4.1 Selected WoE-binnig and GRP-binning

Following section describes the binning approach and following process of selecting the most meaningful predictors for building the scorecard.

4.1.1 Weight-Of-Evidence Binning

The default binning method is “width” which only supports numerical values. Same accounts for “freq”. The other two methods “tree” and “chimerge” support both numerical and categorical variables. Therefore, we will only consider later two to keep categorical variables human-readable.

```
bin_tree_standard = woebin(dt_list$train, "creditability", save_breaks_list = "breaks.list", method="tr
```

```
## v Binning on 727 rows and 14 columns in 00:00:02
```

```
bin_chi_standard = woebin(dt_list$train, "creditability", save_breaks_list = "breaks.list", method="chi
```

```
## v Binning on 727 rows and 14 columns in 00:00:01
```

```
#woebin_plot(bin_tree_standard)
#woebin_plot(bin_chi_standard)
```

As for further evaluation we only want to focus on one binning method, those default binnings are now being evaluated and the better one further tuned for the usage in the scorecard.

```
#data_woe_chi_standard <- lapply(dt_list, function(x) woebin_ply(x, bin_chi_standard))
#summary(glm(creditability ~ ., family = binomial(), data = data_woe_chi_standard$train))
```

```
#data_woe_tree_standard <- lapply(dt_list, function(x) woebin_ply(x, bin_tree_standard))
#summary(glm(creditability ~ ., family = binomial(), data = data_woe_tree_standard$train))
```

With an AIC-value of 686.56 the tree method shows a better fit to our simple general regression model than chi-merge with 692.69, therefore we will continue to use only the tree method for further evaluation.

After this selection we now continue to fine-tune the selections of bins. In the following the results of certain default-binnings are presented that show some semantic inconsistencies.

In the case of the "purpose"-column "others" is thrown together with "education". Although the overall count is low, "education" should semantically be separated from "others".

```
bin_tree_standard$purpose[, c("variable", "bin", "count", "bin_iv")]
```

```
##    variable                                          bin count       bin_iv
## 1:  purpose retraining%,%car (used)%,%radio/television   284 0.1038486990
## 2:  purpose              furniture/equipment%,%repairs   144 0.0002242187
## 3:  purpose car (new)%,%domestic appliances%,%business   253 0.0487911020
## 4:  purpose                          education%,%others    46 0.0324807583
```

Also in the case of "savings.account.and.bonds", the categories "unknown" and "no savings account" should be seperated from a petitioner with an actual savings account.

```
bin_tree_standard$savings.account.and.bonds[, c("variable", "bin", "count", "bin_iv")]
```

```
##                     variable
## 1: savings.account.and.bonds
## 2: savings.account.and.bonds
## 3: savings.account.and.bonds
##                                                                    bin count
## 1:                                                          ... < 100 DM   431
## 2:                                                  100 <= ... < 500 DM    78
## 3: 500 <= ... < 1000 DM%,%... >= 1000 DM%,%unknown/ no savings account   218
##          bin_iv
## 1: 0.0445409119
## 2: 0.0002055698
## 3: 0.1146240838
```

Furthermore, in the case of "other.debtors.or.guarantors", the predictors "none" and "co-aplicant" will be seperated. Together they form a rather large bin, so that the overall predictor would not be as useful with this combined bin.

```
bin_tree_standard$other.debtors.or.guarantors[, c("variable", "bin", "count", "bin_iv")]
```

```
##                         variable                 bin count      bin_iv
## 1: other.debtors.or.guarantors none%,%co-applicant   686 0.001245696
## 2: other.debtors.or.guarantors           guarantor    41 0.024978478
```

Lastly the predictor "other.installment.plans" contains only 3 categorical variables out of which 2 are grouped into the same bin. Out of scarcity of categories we decided to also break up this bin.

```
bin_tree_standard$other.installment.plans[, c("variable", "bin", "count", "bin_iv")]
```

```
##                    variable           bin count      bin_iv
## 1: other.installment.plans bank%,%stores   130 0.023570253
## 2: other.installment.plans          none   597 0.005558214
```

The following block applies our before-mentioned rules to both the tree binning as well as the chi-merge approach for further evaluation.

```
bin_tree_opt = woebin(dt_list$train, "creditability", save_breaks_list = "breaks.list", method="tree", s
  purpose = c("others", "retraining%,%car (used)%,%radio/television", " furniture/equipment%,%repairs",
  savings.account.and.bonds = c("unknown/ no savings account", "    ... < 100 DM", "100 <= ... < 500 DM
  other.debtors.or.guarantors = c("none", "co-applicant", "guarantor"),
  other.installment.plans = c("bank", "stores", "none")
))
```

```
## v Binning on 727 rows and 14 columns in 00:00:02
```

```
#woebin_plot(bin_tree_opt)

bin_chi_opt = woebin(dt_list$train, "creditability", save_breaks_list = "breaks.list", method="chimerge
  purpose = c("others", "retraining%,%car (used)%,%radio/television", " furniture/equipment%,%repairs",
  savings.account.and.bonds = c("unknown/ no savings account", "    ... < 100 DM", "100 <= ... < 500 DM
  other.debtors.or.guarantors = c("none", "co-applicant", "guarantor"),
  other.installment.plans = c("bank", "stores", "none")
))
```

```
## v Binning on 727 rows and 14 columns in 00:00:01
```

```
#woebin_plot(bin_tree_opt)
```

4.2 Logistic regression results and parameter testing for WoE- & GRP-binning

In this section we are going to analyze the information values for predictors of the selected tree-based binning to make a selection of the seven most meaningful ones.

To do this, we are going to fit a general linear regression model to the given parameters (in their corresponding bins) and calculate probability that each parameter indicates a default.

As a baseline, to compare later selections that are calculated by fitting a logistic regression function, we use the information value as calculated by the scorecard R package.

```
iv(dt_list$train,y="creditability")
```

```
##                                        variable  info_value
##  1:  status.of.existing.checking.account 0.670885673
##  2:                    duration.in.month 0.374915860
##  3:                         age.in.years 0.333046200
##  4:                       credit.history 0.331527725
##  5:                              purpose 0.197310968
```

```
##  6:                              savings.account.and.bonds 0.159755734
##  7:                               present.employment.since 0.094729485
##  8:                                               property 0.079993897
##  9:                                                housing 0.067756976
## 10:                           other.debtors.or.guarantors 0.047719305
## 11:                               other.installment.plans 0.029132680
## 12:                                         credit.amount 0.028294397
## 13: installment.rate.in.percentage.of.disposable.income 0.003635227
```

4.2.1 Bin-WoE transformation of predictor variables

First the individual bins have to be transformed into a format that the general linear regression can ingest.

```
data_woe_tree_opt <- lapply(dt_list, function(x) woebin_ply(x, bin_tree_opt))
```

```
## v Woe transformating on 727 rows and 13 columns in 00:00:01
## v Woe transformating on 273 rows and 13 columns in 00:00:01
```

Afterwards we calculate statistics on the fit of individual predictors using the summary function.

```
summary(glm(creditability ~ ., family = binomial(), data = data_woe_tree_opt$train))
```

```
##
## Call:
## glm(formula = creditability ~ ., family = binomial(), data = data_woe_tree_opt$train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2121  -0.6993  -0.3780   0.7520   2.4669
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                        -0.85412    0.09949
## status.of.existing.checking.account_woe             0.81053    0.12738
## duration.in.month_woe                               0.75696    0.19609
## credit.history_woe                                  0.67785    0.18097
## purpose_woe                                         0.97945    0.23327
## credit.amount_woe                                   0.72113    0.21859
## savings.account.and.bonds_woe                       0.78679    0.25880
## present.employment.since_woe                        0.58633    0.33073
## installment.rate.in.percentage.of.disposable.income_woe  2.86724    1.62629
## other.debtors.or.guarantors_woe                     1.00355    0.44167
## property_woe                                        0.14173    0.39706
## age.in.years_woe                                    0.97846    0.26132
## other.installment.plans_woe                         0.59294    0.58173
## housing_woe                                         0.34542    0.40067
##                                                    z value Pr(>|z|)
## (Intercept)                                         -8.585  < 2e-16 ***
## status.of.existing.checking.account_woe              6.363 1.97e-10 ***
## duration.in.month_woe                                3.860 0.000113 ***
## credit.history_woe                                   3.746 0.000180 ***
## purpose_woe                                          4.199 2.68e-05 ***
## credit.amount_woe                                    3.299 0.000970 ***
## savings.account.and.bonds_woe                        3.040 0.002365 **
## present.employment.since_woe                         1.773 0.076254 .
## installment.rate.in.percentage.of.disposable.income_woe   1.763 0.077891 .
```

```
## other.debtors.or.guarantors_woe                              2.272 0.023078 *
## property_woe                                                 0.357 0.721121
## age.in.years_woe                                             3.744 0.000181 ***
## other.installment.plans_woe                                  1.019 0.308068
## housing_woe                                                  0.862 0.388625
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 886.32  on 726  degrees of freedom
## Residual deviance: 658.60  on 713  degrees of freedom
## AIC: 686.6
##
## Number of Fisher Scoring iterations: 5
```

Additionally, we are doing an analysis of variance (anova).

```
anova(glm(creditability ~ ., family = binomial(), data = data_woe_tree_opt$train), test="F")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: creditability
##
## Terms added sequentially (first to last)
##
##
##                                                    Df Deviance Resid. Df
## NULL                                                                 726
## status.of.existing.checking.account_woe             1   92.489       725
## duration.in.month_woe                               1   39.868       724
## credit.history_woe                                  1   23.016       723
## purpose_woe                                         1   17.614       722
## credit.amount_woe                                   1   10.404       721
## savings.account.and.bonds_woe                       1   12.786       720
## present.employment.since_woe                        1    6.850       719
## installment.rate.in.percentage.of.disposable.income_woe  1    2.006       718
## other.debtors.or.guarantors_woe                     1    5.511       717
## property_woe                                        1    0.115       716
## age.in.years_woe                                    1   15.448       715
## other.installment.plans_woe                         1    0.880       714
## housing_woe                                         1    0.740       713
##                                                   Resid. Dev       F
## NULL                                                  886.32
## status.of.existing.checking.account_woe               793.84 92.4894
## duration.in.month_woe                                 753.97 39.8676
## credit.history_woe                                    730.95 23.0159
## purpose_woe                                           713.34 17.6140
## credit.amount_woe                                     702.93 10.4041
## savings.account.and.bonds_woe                         690.15 12.7859
## present.employment.since_woe                          683.30  6.8496
## installment.rate.in.percentage.of.disposable.income_woe  681.29  2.0057
## other.debtors.or.guarantors_woe                       675.78  5.5107
```

```
## property_woe                                                675.67  0.1153
## age.in.years_woe                                            660.22 15.4479
## other.installment.plans_woe                                 659.34  0.8802
## housing_woe                                                 658.60  0.7399
##                                                            Pr(>F)
## NULL
## status.of.existing.checking.account_woe                < 2.2e-16 ***
## duration.in.month_woe                                  2.718e-10 ***
## credit.history_woe                                     1.607e-06 ***
## purpose_woe                                            2.706e-05 ***
## credit.amount_woe                                      0.0012574 **
## savings.account.and.bonds_woe                          0.0003492 ***
## present.employment.since_woe                           0.0088661 **
## installment.rate.in.percentage.of.disposable.income_woe 0.1567113
## other.debtors.or.guarantors_woe                        0.0189005 *
## property_woe                                           0.7342424
## age.in.years_woe                                       8.481e-05 ***
## other.installment.plans_woe                            0.3481364
## housing_woe                                            0.3897083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to above analysis following 7 predictors create the best result considering WoE-binning:

- status.of.existing.checking.account_woe
- duration.in.month_woe
- credit.history_woe
- purpose_woe
- credit.amount_woe
- savings.account.and.bonds_woe
- age.in.years

4.2.2 Bin-Group transformation of predictor variables

In order to group the given bins without the WoE-transformation, the categorical values have to be transformed using one-hot encoding.

```
non_numerical_variables = c("status.of.existing.checking.account_bin", "credit.history_bin", "purpose_bi
"savings.account.and.bonds_bin", "present.employment.since_bin",
"other.debtors.or.guarantors_bin", "property_bin", "other.installment.plans_bin", "housing_bin")

# getting bins
data_grp_tree_opt = lapply(dt_list, function(x) woebin_ply(x, bin_tree_opt, to = 'bin'))
```

```
## v Woe transformating on 727 rows and 13 columns in 00:00:01
## v Woe transformating on 273 rows and 13 columns in 00:00:01
```

```
# one hot encoding to use with linear regression
data_grp_1h<-list()
data_grp_1h$train <- one_hot(data_grp_tree_opt$train, var_encode = non_numerical_variables)

ata_grp_tree_opt <- glm(creditability ~ ., family = binomial(), data = data_grp_tree_opt$train)
```

After fitting the linear regression model the scores are calculated in the same fashion as was the case using the WoE binning. For readability reasons we exclude the calculation of the values and present the resulting selected variables below.

```
#anova(ata_grp_tree_opt, test="F")
```

The resulting predictors using GRP binning are following:

- status.of.existing.checking.account
- duration.in.month
- credit.history
- purpose
- credit.amount
- savings.account.and.bonds
- age.in.years

4.3 Credit scorecard model validation with in- and out-of-sample Gini coefficients (WoE- & GRP)

Now we can use WoE and GRP-bins to calculate the Gini coefficient:

Calculate Gini using perf_eva, which calculates metrics to evaluate the performance of the binomial classification model.

```
woe_vars_sel <- list("creditability", "status.of.existing.checking.account", "duration.in.month", "credi
                 "purpose", "credit.amount", "savings.account.and.bonds", "age.in.years")

dt_f_woe_sel <- var_filter(germancredit[, names(germancredit) %in% woe_vars_sel], y = "creditability")
```

```
## v Variable filtering on 1000 rows and 7 columns in 00:00:00
## v 0 variables are removed
```

```
dt_woe_list_sel <- split_df(dt_f_woe_sel, y = "creditability", ratios = c(0.75, 0.25))

woe_label_sel <- lapply(dt_woe_list_sel, function(x) x$creditability)

data_woe_chi_opt_sel <- lapply(dt_woe_list_sel, function(x) woebin_ply(x, bin_chi_opt))
```

```
## v Woe transformating on 727 rows and 7 columns in 00:00:01
## v Woe transformating on 273 rows and 7 columns in 00:00:01
```

```
ata_woe_chi_opt_train_sel <- glm(creditability ~ ., family = binomial(), data = data_woe_chi_opt_sel$tra

woe_gini_pred_train_sel <- lapply(data_woe_chi_opt_sel, function(x) predict(ata_woe_chi_opt_train_sel,

woe_gini_train_sel <- perf_eva(
         label = woe_label_sel,
         pred = woe_gini_pred_train_sel,
         binomial_metric = "gini",
         show_plot=FALSE
 )

woe_gini_train_sel
```

```
## $binomial_metric
## $binomial_metric$train
##         Gini
## 1: 0.6327731
##
## $binomial_metric$test
##         Gini
```

```
## 1: 0.5304375
```

```
grp_vars_sel <- list("creditability", "status.of.existing.checking.account", "duration.in.month", "credi
                "purpose", "credit.amount", "savings.account.and.bonds", "present.employment.since")

dt_f_grp_sel <- var_filter(germancredit[, names(germancredit) %in% grp_vars_sel], y = "creditability")
```

```
## v Variable filtering on 1000 rows and 7 columns in 00:00:00
## v 0 variables are removed
```

```
dt_grp_list_sel <- split_df(dt_f_grp_sel, y = "creditability", ratios = c(0.75, 0.25))

grp_label_sel <- lapply(dt_grp_list_sel, function(x) x$creditability)

data_grp_chi_opt_sel <- lapply(dt_grp_list_sel, function(x) woebin_ply(x, bin_chi_opt, to = 'bin'))
```

```
## v Woe transformating on 727 rows and 7 columns in 00:00:01
## v Woe transformating on 273 rows and 7 columns in 00:00:01
```

```
data_grp_1h_sel<-list()
data_grp_1h_sel$test <- one_hot(data_grp_chi_opt_sel$test, var_encode = non_numerical_variables)
data_grp_1h_sel$train <- one_hot(data_grp_chi_opt_sel$train, var_encode = non_numerical_variables)

ata_grp_chi_opt_train_sel <- glm(creditability ~ ., family = binomial(), data = data_grp_1h_sel$train)

grp_gini_pred_train_sel <- lapply(data_grp_1h_sel, function(x) predict(ata_grp_chi_opt_train_sel, type =


grp_gini_train_sel <- perf_eva(
        label = grp_label_sel,
        pred = grp_gini_pred_train_sel,
        binomial_metric = "gini",
        show_plot=FALSE
)

grp_gini_train_sel
```

```
## $binomial_metric
## $binomial_metric$test
##        Gini
## 1: 0.580279
##
## $binomial_metric$train
##         Gini
## 1: 0.6368483
```

5 Summary

The main purpose of this paper was to develop a credit risk scorecard model and evaluate it using the out-of-sample gini-coefficient. By doing this we got a small insight into the development process of such a model. As we can say that the banks will probably have greater amount of data, both in the amount of different variables and of data points.

At first, we looked at the theory behind the different methods we used. Here we talked about the meaning of binning and how we need to transform the data in order for us to use them in our model. Then the next part was looking at how the logistic regression model we were using to create our model works. After the

regression model we described how it can be used in our scenario of creating a scorecard model. With the scorecard model in place we then needed to evaluate it and looked at how accurate it predicts defaulting clients. For this evaluation we described the gini-coefficient.

The main part of this paper is to implement the theory into a practical scorecard model and to evaluate it. At first we needed to identify the best way to pre-process the data. We came to the conclusion that the best was to use the tree method as it showed a better fit than the ChiMerge binning. After defining the binning we created both a regression model using WoE-binning as well as one with GRP-binning. We then choose 7 variables that we then used to create our scorecard model. Then finally we evaluated both of these models using the out-of-sample gini-coefficient and came to the conclusion that the GRP-binning approach is the better one.

We cannot say for certain why the GRP-model is better than the WoE-model. However our guess is that the data-set is a bit too small and there is also a possibility of outliers that skew some binnings in different directions.

Finally we found that scorecard models, like the one we implemented, are great tools for financial institutions to improve their business by helping them decrease the likely-hood of a defulting customer.

References

Persson, R.
2021. Weight of evidence transformation in credit scoring models: How does it affect the discriminatory power? Student Paper.

Peussa, A.
2016. Credit risk scorecard estimation by logistic regression. master thesis. *Helsinki: University of Helsinki*.

Poon, M.
2007. Scorecards as devices for consumer credit: The case of fair. *The Sociological Review*, 55(2):284–306.

Sperandei, S.
2014. Understanding logistic regression analysis. *Biochemia Medica*, 24(1):12–18.