# "German Credit" scoring data analysis report

Marta Karaś, Jan Idziak

May 18, 2015

# Table of content

```
## [1] "/home/martakarass/my-store/studies/PW/project-scoring-data"
```

# Part I
# Introduction

In this part of the report we provide answers to the following questions about the "German Credit" data analysis we performed.

1. *Why was the study undertaken?*
2. *What was the purpose of the research? What research questions were stated?*

## 1 Data analysis context

### 1.1 Motivation

This report presents results of the "German Credit" scoring data analysis which was performed as a project assignment for the "Pozyskiwanie Wiedzy" course, which we attended at Wroclaw University of Technology, Faculty of Fundamental Problems of Technology (W-11), Mathematics programm (Master) in the 2014/15 summer semester. The lecturer of the course (both lectures and laboratories) is Ph.D. Adam Zagdański.

The main goal of the project is to make use of the variety of data-mining methods we have become familiar with during the course, in order to perform complete data analysis of selected data set. We also aim to pay attention to the practical appliacnces of some parts of our work.

### 1.2 Research questions

We stated the following research purposes for our analysis.

1. Find and describe relations in the data (relations bewteen explanatory variables and response variable, relations bewteen explanatory variables).
2. Compare different methods / algorithms to perform exploratory data analysis and predictive data analysis.
3. Provide a summary of the analysis, containing suggestions of practical appliance and remarks regarding possible further research.

## Part II
# Materials and methods

In this part of the report we describe the data set we obtained and the methods we use in the analysis.

This section is rather of the decriptional / theoretical character. For a list of actual analysis steps, the outputs of the methods and more, please refer to the III part of this report.

## 2 Data set

We perform analysis with the use of The (Statlog) German Credit Data we have obtained from the UCI Machine Learning Repository site.

### 2.1 Data set description

The data set contains data on 20 variables and the classification whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants. The file provided contains variables with values encoded accoring to the following schema:

- Attribute 1: (qualitative) Status of existing checking account
  A11 : ... < 0 DM
  A12 : 0 <= ... < 200 DM
  A13 : ... >= 200 DM / salary assignments for at least 1 year
  A14 : no checking account
- Attribute 2: (numerical) Duration in month
- Attribute 3: (qualitative) Credit history
  A30 : no credits taken/ all credits paid back duly
  A31 : all credits at this bank paid back duly
  A32 : existing credits paid back duly till now
  A33 : delay in paying off in the past
  A34 : critical account/ other credits existing (not at this bank)
- Attribute 4: (qualitative) Purpose
  A40 : car (new)
  A41 : car (used)
  A42 : furniture/equipment
  A43 : radio/television
  A44 : domestic appliances
  A45 : repairs
  A46 : education
  A47 : (vacation - does not exist?)
  A48 : retraining
  A49 : business
  A410 : others

- Attribute 5: (numerical) Credit amount
- Attribute 6: (qualitative) Savings account/bonds
  A61 : ... < 100 DM
  A62 : 100 <= ... < 500 DM
  A63 : 500 <= ... < 1000 DM
  A64 : .. >= 1000 DM
  A65 : unknown/ no savings account
- Attribute 7: (qualitative) Present employment since
  A71 : unemployed
  A72 : ... < 1 year
  A73 : 1 <= ... < 4 years
  A74 : 4 <= ... < 7 years
  A75 : .. >= 7 years
- Attribute 8: (numerical) Installment rate in percentage of disposable income
- Attribute 9: (qualitative) Personal status and sex
  A91 : male : divorced/separated
  A92 : female : divorced/separated/married
  A93 : male : single
  A94 : male : married/widowed
  A95 : female : single
- Attribute 10: (qualitative) Other debtors / guarantors
  A101 : none
  A102 : co-applicant
  A103 : guarantor
- Attribute 11: (numerical) Present residence since
- Attribute 12: (qualitative) Property
  A121 : real estate
  A122 : if not A121 : building society savings agreement/ life insurance
  A123 : if not A121/A122 : car or other, not in attribute 6
  A124 : unknown / no property
- Attribute 13: (numerical) Age in years
- Attribute 14: (qualitative) Other installment plans
  A141 : bank
  A142 : stores
  A143 : none
- Attribute 15: (qualitative) Housing
  A151 : rent
  A152 : own
  A153 : for free
- Attribute 16: (numerical) Number of existing credits at this bank

- Attribute 17: (qualitative) Job
  A171 : unemployed/ unskilled - non-resident
  A172 : unskilled - resident
  A173 : skilled employee / official
  A174 : management/ self-employed/ highly qualified employee/ officer

- Attribute 18: (numerical) Number of people being liable to provide maintenance for

- Attribute 19: (qualitative) Telephone
  A191 : none
  A192 : yes, registered under the customers name

- Attribute 20: (qualitative) Foreign worker
  A201 : yes
  202 : no

The classification variable states whether there was a default case ('bad' client - failed to pay off the credit) or not ('good' client).

- Classification: (qualitative) Default
  1 (default)
  0 (non-default)

# 3 Binning countinuous variables

In credit scoring, Information Value (IV) is frequently used to compare predictive power among variables. When developing new scorecards using logistic regression, variables are often binned and recoded using WoE concept.

## 3.1 Weight of Evidence (WoE)

One of our goals when binning variables is to maximize Information Value. Weight of Evidence (WoE) for single bin is defined as:

$$WoE = \left[ ln \left( \frac{\text{Relative Frequency of Goods}}{\text{Relative Frequency of Bads}} \right) \right] \times 100.$$

We can see that value of WoE will be 0 if the odds of Relative Frequency of Goods / Relative Frequency Bads is equal to 1. If the Relative Frequency of Bads in a group is greater than the Relative Frequency of Goods, the odds ratio will be less than 1 and the WoE will be a negative number; if the Relative Frequency of Goods is greater than the Relative Frequency of Bads in a group, the WoE value will be a positive number.

## 3.2 Information Value (IV)

We define Information Value of the variable as follow:

$$IV = \sum_{i=1}^{k} \left[ (\text{Relative Frequency of Goods}_i - \text{Relative Frequency of Bads}_i) \times ln \left( \frac{\text{Relative Frequency of Goods}}{\text{Relative Frequency of Bads}} \right) \right],$$

By convention the values of the IV statistic can be interpreted as follows. If the IV statistic is:

- Less than 0.02, then the predictor is not useful for modeling (separating the Goods from the Bads),
- 0.02 to 0.1, then the predictor has only a weak relationship to the Goods/Bads odds ratio,
- 0.1 to 0.3, then the predictor has a medium strength relationship to the Goods/Bads odds ratio,
- 0.3 or higher, then the predictor has a strong relationship to the Goods/Bads odds ratio.

## 3.3 Motivation

The WoE recoding of predictors is particularly well suited for subsequent modeling using Logistic Regression. Specifically, logistic regression will fit a linear regression equation of predictors (or WoE-recoded continuous predictors) to predict the logit-transformed binary Goods/Bads variable. Therefore, by using WoE-recoded predictors in logistic regression the predictors are all prepared and coded to the same WoE scale, and the parameters in the linear logistic regression equation can be directly compared. ([8])

# 4 Feature selection

Following [1], feature selection is essentially a task to remove irrelevant and/or redundant features. *Irrelevant features* cam be removed without affecting learning performance. *Redundant features* are a type of irrelevant feature. The distinction is that redundant feature implies the co-presence of another feature; individually, each feature is relevant, but the removal of one of them will not affect learning performance.

The selection of features may be achieved in two ways:

1. **Feature ranking**. The idea is to rank features according to some criterion and select the top $k$ features.

2. **Subset selection**. The idea is to select a minimum subset of features without learning performance deterioration.

In other words, subset selection algorithms can automatically determine the number of selected features, while feature ranking algorithms need to rely on some given threshold to select features.

The tree typical feature selection models are:

1. **Filter**. In a filter model, one selects the features firstly and then uses this subset to execute a classification algorithm.

2. **Wrapper**. In a wrapper model, one employs a learning algorithm and uses its performance to determine the quality of selected features.

3. **Embedded**. An embedded model of features selection integrates the selection of features in model builidng. An example of such model is a decision tree induction algorithm, in which at each branching node, a feature has to be selected.

In literature, various search strategies are proposed, including: forward, backward, floating, branch-and-bound, and randomized. A relevant issue, regarding exhaustive and heuristic searches is whether there is any reason to perform exhaustive searches if time complexity were not a concern. Research shows that exhaustive search can lead to the features that exacerbate data oerfitting, while heuristic search is less prone to data overfitting in feature selection, facing small data samples.

The evaluation of feature selection often entails two tasks:

1. One is to compare two cases: before and after feature selection. The goal of this task is to observe if feature selection achieves its intended objectives. The aspects of evaluation may include the number of selected features, time, scalability and learning model's performance.

2. The second task is to compare two feature selection algorithms to see if one is better than other for a certain task.

## 4.1 Feature selection algorithms

In this subsection we describe methods for feature selection we use in our analysis. In general, we use the FSelector R package exhaustively. This package contains both algorithms for filtering attributes and algorithms for wrapping classifiers and search attribute subset space.

### 4.1.1 Algorithms for filtering attributes

**CFS filter** CFS is a correlation-based filter method CFS from [2]. It gives high scores to subsets that include features that are highly correlated to the class attribute but have low correlation to each other. Let $Attribute$ be an attribute subset that has $k$ attributes, $rcf$ models the correlation of the attributes to the class attribute, $rff$ - the intercorrelation between attributes. We define $Attribute$ score as:

$$CfsScore(Attribute) = \frac{k\ rcf}{\sqrt{k + k(k-1)rff}}.$$

The algorithm from FSelector R package makes use of *Best-first search* for searching the attribute subset space. In *Best-first search*, the algorithm chooses the best node from all already evaluated ones and evaluates it. The selection of the best node is repeated approximately $max.brackets$ times in case no better node found.

**Chi-squared filter** The algorithm evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.

**Information Gain filter** One of the entropy-based filters. Algorithm evaluates the worth of an attribute by measuring the information gain with respect to the class.

$$InfoGain(Class, Attribute) = H(Class) + H(Attribute) - H(Class|Attribute),$$

where $H$ is the information entropy.

**Gain Ratio filter** One of the entropy-based filters. Algorithm evaluates the worth of an attribute by measuring the gain ratio with respect to the class.

$$GainR(Class, Attribute) = \frac{H(Class) + H(Attribute) - H(Class|Attribute)}{H(Attribute)},$$

where $H$ is the information entropy.

**Symmetrical Uncertainty filter** One of the entropy-based filters. Algorithm evaluates the worth of a set attributes by measuring the symmetrical uncertainty with respect to another set of attributes.

$$SymmU(Class, Attribute) = 2\frac{H(Class) + H(Attribute) - H(Class|Attribute)}{H(Attribute) + H(Class)},$$

where $H$ is the information entropy.

**Linear Correlation filter** The algorithm finds weights of continous attributes basing on their Pearson's correlation with continous class attribute.

**Rank Correlation filter** The algorithm finds weights of continous attributes basing on their Spearman's correlation with continous class attribute.

**OneR algorithm** The algorithms find weights of discrete attributes basing on very simple association rules involving only one attribute in condition part. In other words, it uses the minimum-error attribute for prediction, discretizing numeric attributes. For more information, see [4].

**RReliefF filter** The algorithm evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Considering that result, it evaluates weights of attributes. Can operate on both discrete and continuous class data. For more information see [5,6,7].

**Consistency-based filter** Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. Consistency of any subset can never be lower than that of the full set of attributes, hence the usual practice is to use this subset evaluator in conjunction with a Random or Exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes. The FSelector R package implementation makes use of *Best-first search* for searching the attribute subset space. Works for continuous and discrete data.

**RandomForest filter** It is a wrapper for variable importance measure produced by randomForest algorithm. The FSelector R package implementation allows for two types of importance measure:

1. mean decrease in accuracy,
2. mean decrease in node impurity.

The first measure is computed from permuting OOB (out-of-bound) data: For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. If the standard deviation of the differences is equal to 0 for a variable, the division is not done (but the average is almost always equal to 0 in that case).
The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares.

### 4.1.2 Algorithms for wrapping classifiers and search attribute subset space

In general, the wrapper approach depends on the so called *evaluation function* that is used to return a numeric value (a score) indicating how important a given subset of features is. Typically, one uses the classification-accuracy (usually based on cross-validation) as the score for the subset.

Below we provide a brief description of the algorithms for searching atrribute subset space.

**Greedy search** At first, greedy search algorithms expand starting node, evaluate its children and choose the best one which becomes a new starting node. This process goes only in one direction. *Forward search* starts from an empty and *backward search* from a full set of attributes.

**Best-first search** The algorithm is similar to *Forward search* besides the fact that is chooses the best node from all already evaluated ones and evaluates it. In the FSelector R package implementation, the selection of the best node is repeated approximately *max.brackets* times in case no better node found.

**Hill climbing search** The algorithm starts with a random attribute set. Then it evaluates all its neighbours and chooses the best one. It might be susceptible to local maximum.

**Exhaustive search** The algorithm searches the whole attribute subset space in breadth-first order.

# 5 Classification

## 5.1 Classification algorithms

**kNN k- nearest neighbours** Method is used for modeling in problem of regression or classification. It is simple algorithym using lazy learning. There is no actual model so all the computation is done while classification. In the problem of classification the result for every single observation is a class for which in k closest neighbours from the training set is the most popular.

**Decission trees** Decission tree is a method that perform recussive partition of the set for every predictor. In each step there is chosen split that separates the set between classes the most according to one of the meassures. The most populat measures are Information GAIN or GINI. For continous data it is desired to partition variable into categorical (It could cause loss of the information). Result is highly corelatet with the learning set. Nonetheless it is easy to interpret, and attractive visually. Another plus is that decission trees do not have any assumptions about distribution of the data and algorithms works fast.

**Random forest** It is curently one of the most popular method in machine learning. Its popularity grows thanks to good performance and small assumptions. However it performs well, it is hard to interpret the results, as long as model is complicated and consists many decission trees. For this method in each step of decission tree creation there is taken random subset of the features and then one of them is taken for split. This is done untli appropriate settled level. For Random forests the computatio time is much higher than for decission trees. Mostly it is because not only one tree is fitted but usualy much more. One of the biggest disadvantages of this model is hard interpretation of the output. Even though the subset of predictors is only taken it shows much better results than other regulat methods for different data sets.

**Logistic regression** The most popular method among application in banks, insurance companies and the industrie for modeling binary data (It could servs also for prediction multiclass data). It owes popularity to its simplicity, easy open form and straight interpretation. It is subject to produce Score Card. Method is a particular type of generalized linear model where link function has logit form $logit(p) = log(\frac{p}{1-p})$. It means that probability of occurance particular event, is modeled indirectly, as a appropriate transformation.

$$logit(p_j) = log(\frac{p_j}{1-p_j}) = \Sigma_{i=1}^{n}\beta_i X_{i,j}$$

Where $p_j$ is estimated probability, $\beta_i$ is factor for $X_{i,j}$ and X represents the features of observation.

**Linear discriminant analysis** It is another linear method. Under the assumption of normality and equality of covariance matrices within classes.

$$Pr(C = k|X = x) = \frac{f_k(x)\pi_k}{\Sigma_{l=1}^{K} f_l(x)\pi_l}$$

where $C = k$ represents particular class affiliation, $x$ is observation vector and $f_k(x)$, has appropriate Gaussian distribution with the mean $\mu_k$ and covariance matrixvariance $\Sigma$ and $\pi_k$ is a-priori classes probability. It is enough to compare numerator as long as denominator for all classes would be the same.

**Quadric discriminant analysis** It is similar method to the linear dyscriminant analysis. It keeps assumption about normality, but in this case covarance matrices could differ. Aproppriate probability function keep its form:

$$Pr(C = k|X = x) = \frac{f_k(x)\pi_k}{\Sigma_{l=1}^{K} f_l(x)\pi_l}$$

As before it is enough to compare numerators for all classes.

**Naive Bayes** Another method that uses Bayesian rule. It is called Naive Bayes as long as it has a naive assumption about loss of correlation between predictors. However this model has easy form, it also perform well in many appliactions.

$$P(Y = k|X = x) = \frac{P(X = x|Y = y)}{P(X = x)} = \frac{P(X_1 = x_1, \ldots X_n = x_n|Y = y)}{P(X = x)}$$

In this case probabilities are just taken as an empirical realisation of the data. It could also fall into problem of zero class frequencies. To omit this situation it is recommended to use one of the smoothing methods.

## 5.2 Classification performance metrics

**Confussion matrix** This is one of simple method to grade quality of the classification. It serves to compare acctual class of an observation to one predicted by model.

| | | Predicted Class | |
|---|---|---|---|
| | | True | False |
| Actuall Class | True | True Positive (TP) | False Negative (FN) |
| | False | False Positive (FP) | True Negative (TN) |

Using confussion matrix it is easier to calculate many of the goodnes of fit measures for the models such as sensitivity, specifity or many more.

**Sensitivity** One of the simple measures called also Recall or True Positive Rate. It measures proportion of predicted as true and actual true. Using confusion matrix it could be given as:

$$TPR = \frac{TP}{TP + FN}$$

**Specifity** This factor is also called True Negative Rate. It measures proportion predicted corretly as false and actuall number of false observations. It is given by:

$$SPC = \frac{TN}{TN + FP}$$

**Precission** Popular measure in information retrival. It repressents fraction of documents relevant to retrived. In binary classification it is defined as:

$$PPV = \frac{TP}{TP + FP}$$

**False discovery rate** It is complementary to the Precission measure that is getting more popular thanks to groving dimensionality of data sets. In many applications it is of an interest of scientist to control this factor. The formula for this coefficient is subsequent:

$$FDR = \frac{FP}{TP + FP} = 1 - PPV$$

**Accuracy** It is simple measure that could be taken as a good indicator for model performance. It takes proportion of all positive clasified to total number of observations. For not equaly distributed observations between groups (for example in spam detection where there is many spam files classificator that predicts everything as a spam would have high Accuracy)

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

**F-measure** F-measure is defined as a combination of Precission and Recall. Both of earlier described indexes gives some information about model, but using them separtely can effect with falling into missclassification for specific types of data.

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

#### 5.2.1 Separation measures

**Kolmogorow Smirnov statistics and distributions** Kolmogorov Smirnov statistics is the biggest distance between distribution functions of scores (probabilities) for accutal groups of True and False. Distribitions shows just simple cumulative distribution functions for classes.

**ROC curve and GINI, AUROC indexes** ROC or Reciver Operating Characteristic is a graphical ilustration that represents preformance of the binary classifier. It is being used in medicine, radiology, biometrics and many more applications It shows proportion of the True Positive rate (on the vertical axis) and False Positive rate (on the horisontal axis) at various thresholds. AUC is factor strongly connected with the ROC curve. Abbreviation stands for area under curve. It could be calculated as follows:

$$AUC = \int_{-\infty}^{\infty} TPR(x)FPR(x)dx$$

**Histogram Good vs Bad** It is just histogram showing distribution of scores (probabilities) within classes. For good models there suppose to be visible difference between hight of the scores for different classes.

# 6 Cluster analysis

## 6.1 Cluster analysis algorithms

## 6.2 Cluster analysis performance metrics

# 7 Dimensionality reduction

## 7.1 Dimensionality reduction algorithms

**Part III**

# Results

# 8 Data preprocessing results

In this section we present the results of data preprocessing we performed. The important parts of this process are: creating derived variables, binning countinuous variables and correcting bins (factor levels) if it seems reasonable.

The procedures and methods used to perform this part of the analysis include:

- visual inspection of density estimator plots and fitting distribution from different distribution families to numeric variables data,
- comparision of optimal discretization method (*smbinning* package) and equal frequency discretization with Information Value as criterion,
- using WoE criterion to define factor levels "similarity".

## 8.1 Searching for missing, corrupt and invalid data

We started the preprocessing in searching for missing, corrupt and invalid data. In our dataset most of the varaibles are factor variables ($PURPOSE$, $EMPLOYMENT$ etc.) Some of them are numeric but consist only of a few qunique values and thus may be seen rather like ordered factors ($NUM\_OF\_MAINTAINED\_PEOPLE$, $RESIDENCE$ etc.) We investigated frequency tables and did not notice anything anusual in the values.

Three variables in the data set are "truly" numeric: $DURATION$, $AMOUNT$ and $AGE$. We did not find anything particularly unusual in the values of these variables. To satisfy our curiosity, we tried to fit a probabilistic distribution to the values. We used $MASS :: fitdistr$ function to perform maximum-likelihood fitting of univariate distribution from selected distribution families. In each case we tried to fit parameters for three distributions families: *Gamma*, *Log-normal* and *Weibull*.

On the graphs below we can see kernel density estimates of variable density (black line) and curves representing density of the distribution fitted. We was not able to fit *Gamma* distribution to the $AMOUNT$ variable values (*Error in stats::optim(x = c(1169, 5951, 2096, 7882, 4870, 9055, 2835, : non-finite finite-difference value [1]*). It seems that *Log-normal* distribution fits quite well in each three cases.

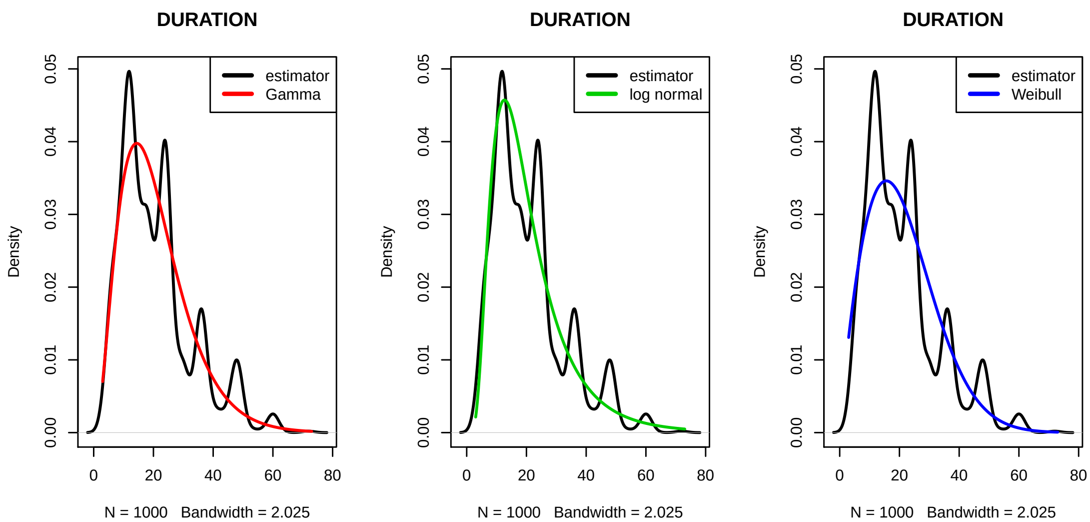


Figure 1: Kernel density estimate of $DURATION$ variable density (black line) and curves representing density of the distribution fitted.

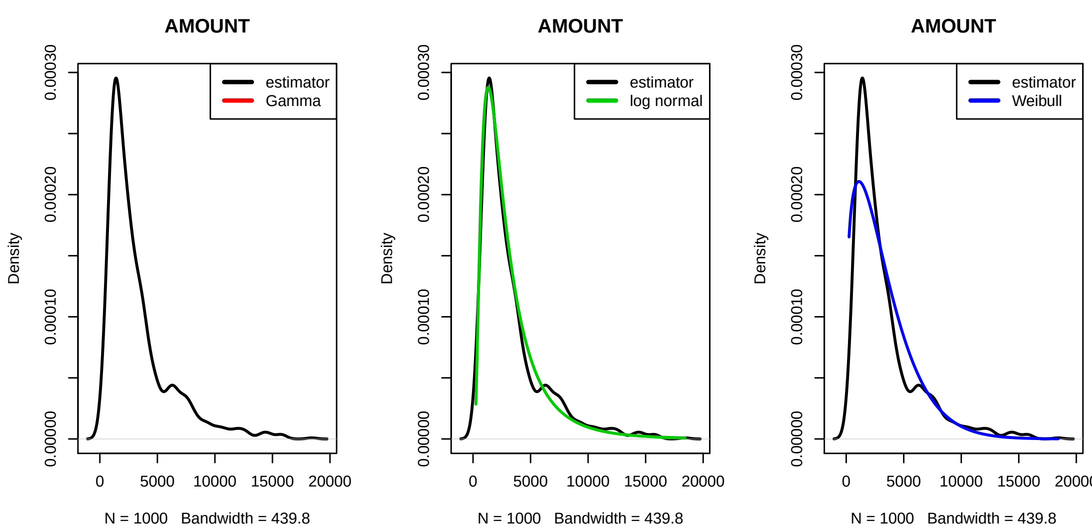


Figure 2: Kernel density estimate of $AMOUNT$ variable density (black line) and curves representing density of the distribution fitted.

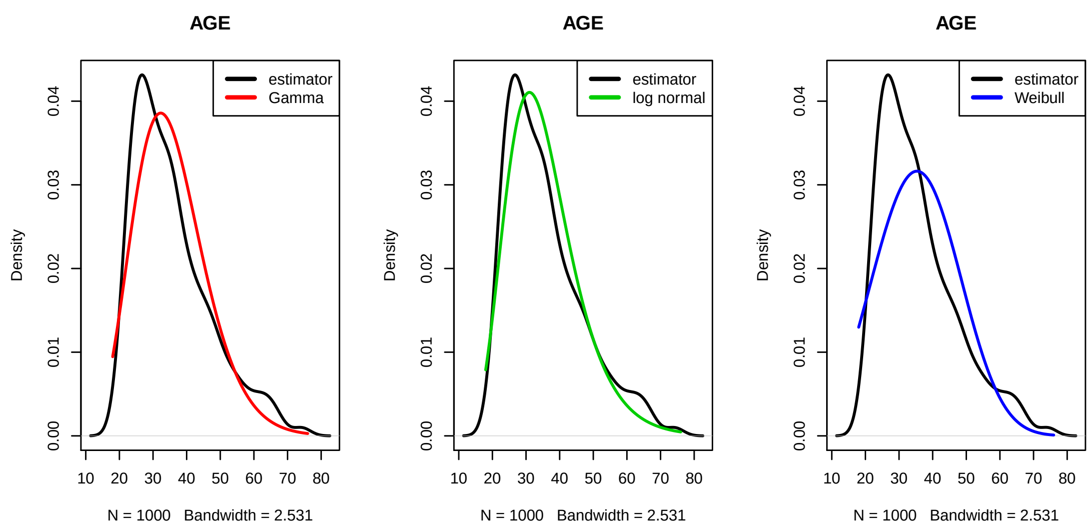


Figure 3: Kernel density estimate of *AGE* variable density (black line) and curves representing density of the distribution fitted.

## 8.2 Creating derived variables

We were considering possibilities of creating derived variables. We came up with propositions of the following formulas:

$$AMOUNT\_TO\_DURATION = AMOUNT/DURATION,$$

$$DURATION\_TO\_AGE = DURATION/AGE,$$

$$AMOUNT\_TO\_AGE = AMOUNT/AGE.$$

On the *Figure 4.* we can see boxplots of $DURATION$, $AGE$ and $AMOUNT$ across two levels of response variable $RES$. On the *Figure 5.* we can see boxplots of derived variables. This comparision can give us intuition how well our new variables separate good and bad bank clients.

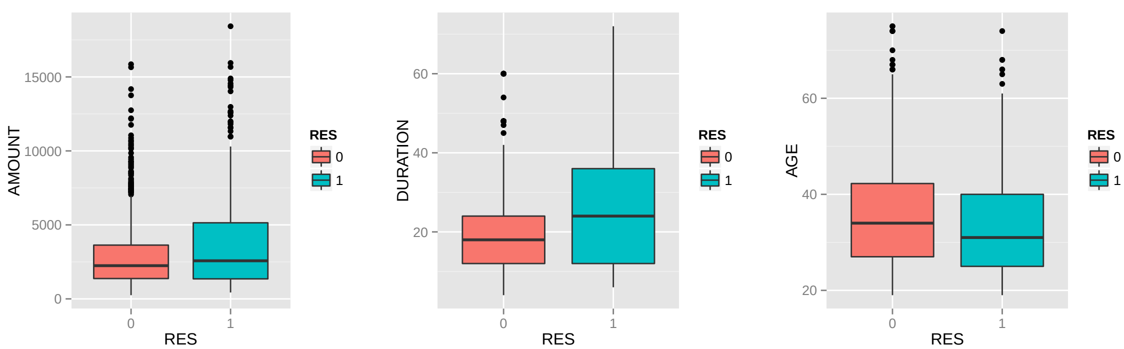


Figure 4: Boxplots of $AMOUNT$, $DURATION$ and $AGE$ variables across two levels of response variable $RES$.

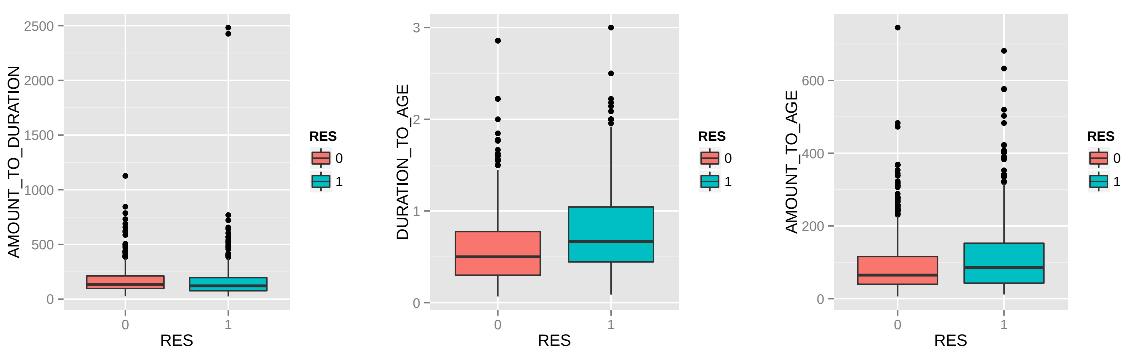


Figure 5: Boxplots of derived variables: $AMOUNT\_TO\_DURATION$, $DURATION\_TO\_AGE$ and $AMOUNT\_TO\_AGE$ across two levels of response variable $RES$.

The boxplots on the *Figure 5.* agree with our intuition: we expect higher *DURATION_TO_AGE* and *AMOUNT_TO_AGE* values for those clients who defaulted ($RES = 1$). On the other hand, we would expect the same for the *AMOUNT_TO_DURATION* variable whereas the plot shows something slightly opposite. However, the value differences on the *AMOUNT_TO_DURATION* boxplot are so fine that we suppose this variable is going to turn out to be of low information value and will not be consider as important one.

### 8.3 Binning continuous variables

In our analysis we compared three approaches of dividing continuous variables into categories:

1. equal frequency discretization (resulted bins are of equal number of observations),
2. supervised discretization which utilizes Recursive Partitioning to categorize the numeric characteristic and compute cutpoins based on Conditional Inference Trees algorithm (*smbinning* package),
3. categorize variable with simple tree model (from *rpart* package, with the use of default tree building parameters).

The IV comparision is presented on the *Figure 6.* below. Note that equal frequency discretization results are presented for:

- the same number of bins as in variable resulted from *smbinning* method; signature: *equal_nbins*,
- the same number of bins as in variable resulted from *rpart* method; signature: *equal_nrparts*.

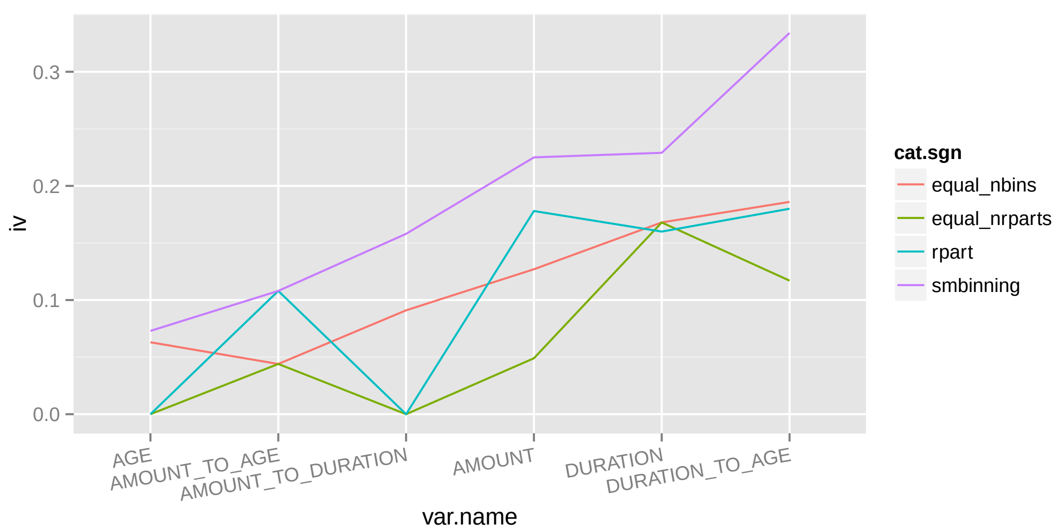


Figure 6: Comparision of Information Values of variables resulted from binning performed by means of 3 differend approaches.

We can see a few interesting things from the plot above:

- The *smbinning* function from the *smbinning* seems to beat other methods in terms of IV of resulted binned variable.
- Derived variable $DURATION\_TO\_AGE$ has a strong relationship to the Goods/Bads odds ratio (over 0.33 IV value), whereas two other derived variables ($AMOUNT\_TO\_AGE$, $AMOUNT\_TO\_DURATION$) seem to have not.

# 9 Classification modelling results

# 10 Cluster analysis results

# 11 Summary

**Part IV**

# Discussion

# References


[1] Computational Methods of Feature Selection (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series), Huan Liu, Hiroshi Motoda, 2007, ISBN-13: 978-1584888789

[2] Hall, M. A., Smith, L. A. (1998). Practical feature subset selection for machine learning. Australian Computer Science Conference. Springer. 181-191.

[3] Liu, H. and Setiono, R., Chi2: Feature selection and discretization of numeric attributes, Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, 338-391, 1995

[4] R.C. Holte (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning. 11:63-91.

[5] Kenji Kira, Larry A. Rendell: A Practical Approach to Feature Selection. In: Ninth International Workshop on Machine Learning, 249-256, 1992.

[6] Igor Kononenko: Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning, 171-182, 1994.

[7] Marko Robnik-Sikonja, Igor Kononenko: An adaptation of Relief for attribute estimation in regression. In: Fourteenth International Conference on Machine Learning, 296-304, 1997.

[8] StatSoft, Formula Guide Weight of Evidence Module http://documentation.statsoft.com/portals/0/formula