

"German Credit" scoring data analysis report

Marta Karaś, Jan Idziak

May 5, 2015

Table of content

I	Introduction	3
1	Data analysis context	3
1.1	Motivation	3
1.2	Research questions	3
II	Materials and methods	4
2	Data set	4
2.1	Data set description	4
3	Feature selection	7
3.1	Feature selection algorithms	8
3.1.1	Algorithms for filtering attributes	8
3.1.2	Algorithms for wrapping classifiers and search attribute subset space	10
4	Classification	11
4.1	Classification algorithms	11
4.2	Classification performance metrics	13
5	Cluster analysis	14
5.1	Dimensionality reduction algorithms	14
5.2	Cluster analysis algorithms	14
5.3	Cluster analysis performance metrics	14
III	Results	15
IV	Discussion	16

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 program (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 applications 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 between explanatory variables and response variable, relations between 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 application 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 according 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

3 Feature selection

Following [1], feature selection is essentially a task to remove irrelevant and/or redundant features. *Irrelevant features* can 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 building. 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 overfitting, 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.

3.1 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.

3.1.1 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, rfc - the intercorrelation between attributes. We define *Attribute* score as:

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

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 continuous attributes basing on their Pearson's correlation with continuous class attribute.

Rank Correlation filter The algorithm finds weights of continuous attributes basing on their Spearman's correlation with continuous 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.

3.1.2 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 attribute 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 it 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.

4 Classification

4.1 Classification algorithms

kNN k- nearest neighbours Method is used for modeling in problem of regression or classification. It is simple algorithm 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.

Decision trees Decision tree is a method that perform recursive 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 measures. The most popular measures are Information GAIN or GINI. For continuous data it is desired to partition variable into categorical (It could cause loss of the information). Result is highly correlated with the learning set. Nonetheless it is easy to interpret, and attractive visually. Another plus is that decision trees do not have any assumptions about distribution of the data and algorithms works fast.

Random forest It is currently 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 decision trees. For this method in each step of decision tree creation there is taken random subset of the features and then one of them is taken for split. This is done until appropriate settled level. For Random forests the computation time is much higher than for decision trees. Mostly it is because not only one tree is fitted but usually 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 regular methods for different data sets.

Logistic regression The most popular method among application in banks, insurance companies and the industries for modeling binary data (It could serve 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 $\text{logit}(p) = \log(\frac{p}{1-p})$. It means that probability of occurrence particular event, is modeled indirectly, as an appropriate transformation.

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

Where p_j is estimated probability, β_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.

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

where $C = k$ represents particular class affiliation, x is observation vector and $f_k(x)$, has appropriate Gaussian distribution with the mean μ_k and covariance matrix variance Σ and π_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 discriminant analysis. It keeps assumption about normality, but in this case covariance matrices could differ. Appropriate probability function keep its form:

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

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 applications.

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

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.

4.2 Classification performance metrics

5 Cluster analysis

5.1 Dimensionality reduction algorithms

5.2 Cluster analysis algorithms

5.3 Cluster analysis performance metrics

Part III
Results

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.