Yang Liu

The evaluation of classification models for credit scoring
Contents

1 Introduction .................................................................................................................................. 1

2 Reasons of the model evaluation................................................................................................. 3

3 Criterions of the model evaluation ............................................................................................... 6
  3.1 The classification accuracy ........................................................................................................ 6
      3.1.1 The estimation of true error rates .................................................................................. 6
      3.1.2 Sampling of train and test data ...................................................................................... 7
          3.1.2.1 Sampling for the adjustment of model structures .................................................. 7
          3.1.2.2 Sampling for population drift .................................................................................. 8
      3.1.3 The size of train and test data ......................................................................................... 9
  3.2 Practicability criterions ............................................................................................................. 14

4 Methods of the model evaluation ................................................................................................. 16
  4.1 Confusion matrix and two types of errors ............................................................................... 16
  4.2 The tradeoff of two types of error rates .................................................................................. 17
      4.2.1 ROC Curve ..................................................................................................................... 17
      4.2.2 Cost function .................................................................................................................. 19
      4.2.3 The determination of the threshold value ....................................................................... 21
  4.3 Incremental analysis ................................................................................................................ 24
      4.3.1 Learning curve ................................................................................................................ 24
      4.3.2 Incremental case analysis .............................................................................................. 24
      4.3.3 Incremental complexity analysis .................................................................................... 25
      4.3.4 The basic phenomenology of incremental analysis ....................................................... 26

5 An example of the credit scoring model evaluation ................................................................. 28
  5.1 Problem definition and data preparation ................................................................................. 28
      5.1.1 Definition of the objective of scoring models and the risk classes .............................. 28
      5.1.2 Preparation of the data ................................................................................................. 32
  5.2 The process of the evaluation .................................................................................................. 34
      5.2.1 Incremental analysis ....................................................................................................... 35
      5.2.2 Comparison analysis ..................................................................................................... 36
5.3 Results of the evaluation

5.3.1 Results of the incremental analysis

5.3.1.1 Linear discriminant analysis and logistic regression

5.3.1.2 Neural networks

5.3.1.3 K-nearest-neighbors

5.3.1.4 Decision trees

5.3.2 Results of the comparison analysis

5.3.2.1 Classification accuracy

5.3.2.2 Practicability criterions

5.4 The comparison of the scoring models with the rating system

6 Summary and conclusion

Appendices

Appendix A. Explanations of some variables in the data set

Appendix B. Brief summary of software and toolsets used in this study

Appendix C. The derived model tree

Literature
Figures

Figure 1/1: The overall data mining process for credit scoring ........................................ 1
Figure 2/1: Part of the process of building various classifiers ........................................... 3
Figure 2/2: The space of classifiers .................................................................................. 4
Figure 2/3: Comparison of classifiers within one algorithm ............................................. 5
Figure 2/4: Comparison of classifiers across algorithms.................................................... 5
Figure 2/5: Comparison of classifiers across a group of preprocessing techniques ...... 5
Figure 3.1.1/1: Three kinds of error rates ........................................................................ 7
Figure 3.1.2.1/1: Three samples for training and testing .................................................. 8
Figure 3.1.2.2/1: Train/Test procedure for nonstationary population .............................. 9
Figure 3.1.3/1: Test sample sizes and statistic significance ............................................. 13
Figure 4.2.1/1: An example of ROC curve ..................................................................... 18
Figure 4.2.1/2: Comparing ROC curves ......................................................................... 19
Figure 4.2.3/1: Graphical way of finding the optimal threshold value .............................. 22
Figure 4.2.3/2: Finding optimal threshold values with a possible range of cost ratios .. 23
Figure 4.3.3/1: Test error tradeoff in terms of model size .............................................. 26
Figure 4.3.4/1: Learning curve for incremental case analysis ........................................... 26
Figure 4.3.4/2: Learning curve for incremental complexity analysis ............................... 27
Figure 5.1.1/1: Input and output of scoring model .......................................................... 31
Figure 5.1.1/2: The distribution of the variable "ZAHLWEISE" for different groups ...... 32
Figure 5.1.2/1: Sampling from available cases for training and testing ............................ 33
Figure 5.2.1/1: Process of the incremental analysis ...................................................... 36
Figure 5.2.2/1: Process of the comparison analysis ....................................................... 37
Figure 5.3.1.1/1: Incremental case analysis of linear discriminant analysis ..................... 38
Figure 5.3.1.1/2: Incremental case analysis of logistic regression .................................. 38
Figure 5.3.1.2/1: Incremental complexity analysis of MLP ............................................. 39
Figure 5.3.1.2/2: Incremental case analysis of MLP ...................................................... 39
Figure 5.3.1.3/1: Incremental case analysis of k-nearest-neighbors ............................... 40
Figure 5.3.1.3/2: Incremental complexity analysis of k-nearest-neighbors .................... 40
Figure 5.3.1.4/1: Incremental complexity analysis of M5 ............................................. 41
Figure 5.3.1.4/2: Incremental case analysis of M5 ......................................................... 41
Figure 5.3.2/1: The model structures and train sample sizes of five models ..............42
Figure 5.3.2.1/1: ROC curves for five algorithms ...................................................43
Figure 5.4/1: The distribution of the scores of A/B companies ................................48
Figure 5.4/2: The distribution of the scores of C companies .....................................48
Figure 5.4/3: The distribution of the scores of D companies .....................................48

Tables

Table 4.1/1: Confusion matrix for two-class problem .............................................16
Table 4.2.2/1: Components of Type I error cost and Type II error cost ...............20
Table 5.1.1/1: The ratings of companies given by the current system .................29
Table 5.1.1/2: Crosstabulation of three clusters and ratings ..................................30
Table 5.1.1/3: Crosstabulation of four clusters and ratings ....................................30
Table 5.1.2/1: List of input variables .....................................................................34
Table 5.3.2.1/1: AUC for five algorithms ...............................................................43
Table 5.3.2.1/2: Optimal models for S = 1/4 .........................................................44
Table 5.3.2.2/1: The comparison of the speed of five models ...............................44
Table 5.3.2.2/2: Variables' coefficients for LDA and LR functions .....................46
Table 5.4/1: Crosstabulation of ratings and score classes .......................................50
1 Introduction

The quantitative method known as credit scoring has been developed for the credit assessment problem. Credit scoring is essentially an application of classification techniques, which classify credit customers into different risk groups. Since the classification technique is one of the data mining techniques, the process of credit scoring can be seen as the process of a data mining application. It utilizes new developed data mining techniques to preprocess input data and to build classification models.

The overall process of data mining is not automatic. A practical solution cannot be found by the simple using of a particular algorithm on the data in databases. The process for credit scoring can be generated as a data mining framework which consists of three stages (cf. Liuy02, Chapter 5). Figure 1/1 shows the three-stage process of a data mining application for credit scoring.

![Figure 1/1: The overall data mining process for credit scoring](image-url)

During this process many human interventions are involved in order to get the optimal solution. In the first stage, data mining experts should cooperate with domain experts to define the problem to be solved by scoring models, to collect and prepare the relevant data.
In the third stage, after scoring models have been built, they are applied in a practical decision process. Models’ performances in reality should be observed. The models usually should be rebuilt after a period of time in order to generate new rules for the changed environment.

In the second stage (data analysis and model building stage), in order to get more reliable results any data mining application should unavoidably experience a process that consists of many steps. These steps can be classified as four parts (see Figure 1/1). Various data mining techniques are included in the process. Many judgments need to be made during this process, e.g. whether a kind of technique should be used; which technique is most suitable; how many training examples should be used to get reliable results; how to choose and tuning the model parameters, etc. The model building process is the process to search the optimal solution for the question at hand.

To search the best solution, the various models produced during the process need to be evaluated. Model evaluation techniques and methods are necessary tools to aid the searching and the choosing of the best solution. The evaluation of the generated models is necessary and useful for the selection of suitable techniques and the selection of final applicable models. The evaluation sub-process, one of the sub-processes in the second stage, plays a fundamental role for a credit scoring project. This paper investigates the issues about this sub-process: the evaluation of classification models for a credit scoring problem.

Chapter 2 clarifies the reasons of the model evaluation and indicates the candidate models that should be evaluated during the model building process. Chapter 3 suggests the evaluation criterions for scoring models, which reflect different aspects of models’ performance. The evaluation methods that include sundry tools are investigated in chapter 4. Based on these evaluation criterions and methods, a case study with actual credit data is shown in chapter 5. The process of utilizing multiple algorithms to build models and evaluating them to reach the optimal solution is elaborated. Finally, the results of the research are summarized in Chapter 6.

---

1 The range of the research in this paper shown in the grid area in Figure 1/1 is part of the whole project for a dissertation.
2 Reasons of the model evaluation

Credit scoring is a complex data mining process. When various innovative data mining techniques are introduced into the process of model building, a large number of classification models (also called classifiers) may be produced. For example, one can use different feature selection techniques to choose input variables; one can select different classification algorithms to train models. One fraction of the process of generating various classifiers is shown in Figure 2/1.

![Diagram](image)

*Figure 2/1: Part of the process of building various classifiers*

Therefore, a classifier is built through a series of choices: the choice of a specific algorithm with particular parameters, as well as the choice of a train sample preprocessed by various techniques.

The classifiers can be shown in a space with multiple effect factors which would be considered to build a classifier (see Figure 2/2). The generated models are determined by two categories of variations: the variation of model algorithms and the variation of the input data. The variation of model algorithms can be resulted from using different model algorithms and setting different parameters of a particular algorithm. The variation of the input data depends on the different sizes of the used sample, or various preprocessing of the input data, such as the selected features.
<table>
<thead>
<tr>
<th>Parameters of models</th>
<th>Missing value treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variation of algorithms</td>
<td>Features in the input data</td>
</tr>
<tr>
<td>Variation of input data</td>
<td>Other changes in the input data</td>
</tr>
</tbody>
</table>

**Figure 2/2: The space of classifiers**

The classifiers in the space, which are generated during the process of model building, are the candidates to be evaluated. We evaluate classifiers in order to determine how the factors affect the performance of the generated classifiers. Although the variation of the generated classifiers may caused by multiple factors, the evaluation of classifiers can be done only in a two-dimensional space. For one particular comparison, other factors should be fixed except the factor to be compared. For example, the comparison may happen in the following situations:

- Within one algorithm: for example, comparing the classifiers generated by a k-nearest-neighbors algorithm with different parameter k (see Figure 2/3).

- Across algorithms: for example, comparing the classifiers generated by different classification algorithms using the same input data (see Figure 2/4).

- Across a group of preprocessing techniques: for example, comparing the classifiers generated by the k-nearest neighbors algorithm using different feature selection techniques (see Figure 2/5).

The aims of the evaluation are different in each situation: finding optimal parameters, selecting optimal algorithms, or selecting suitable preprocessing techniques for the data set at hand, etc.
Reasons of the model evaluation

**Figure 2/3: Comparison of classifiers within one algorithm**

**Figure 2/4: Comparison of classifiers across algorithms**

**Figure 2/5: Comparison of classifiers across a group of preprocessing techniques**
3 Criterions of the model evaluation

3.1 The classification accuracy

3.1.1 The estimation of true error rates

To evaluate classification models the most direct criterion that can be measured quantitatively is the classification accuracy. The equivalent criterion is the misclassification error rate:

$$\text{Error Rate} = \frac{\text{number of incorrectly classified cases}}{\text{total number of cases}}$$  \hspace{1cm} \text{Formula 3.1.1/1}

For a given two-class classification problem, this number varies between the default prediction error rate (of a default model that assigns all cases to the majority class) and the zero error rate (of a perfect model that classify all cases correctly) (cf. GaTa97, P. 9). Any model should have more prediction power than default model, while for various reasons, a perfect model with zero prediction error rate does not exist in practical applications.

The true error rate is defined as the error rate of a model that tested on the true distribution of cases in the underlying population. The true error rate of a model can not be seen before the model is applied on large number of real cases. Waiting to observe the actual outcomes of real cases is of course not the practicable method to evaluate classification model at the time of model building. The true error rate of models can be estimated empirically. The major question is then how to estimate the true error rate empirically.

The error rate of a classifier on the sample cases that were used for the training of the classifier (train data) is called the apparent error rate (see Figure 3.1.1/1). Apparent error rate is not a reliable estimate of the true error rate. The larger the apparent error rate, the more the classifier fits the outcomes of the training data. However, a classifier that fits well because it reproduces the idiosyncrasies of the training data may be expected to give rather poor predictions over the future cases. This phenomenon is called 'over-fitting'. In general, apparent error rates tend to be biased optimistically. The true error rate is almost invariably higher than the apparent error rate (cf. WeKu91, P. 24). Much more believable estimate of the true error rate is the error rate of a classifier when it classifies cases that are not used for the training of the classifier.

The method to estimate true error rate of a classifier is called 'holdout method', or 'train and test method'. The basic idea is this: a sample of data (train data) is given to enable a classifier to be built. Then the classifier is tested on a second independent sample (test data) whose true classifications are known but are unseen to the classifier.
The error rate of a classifier on the test data is called the **test sample error rate**. It is statistically proved that when the test sample size reaches 1000, the test sample error rate is already an extremely accurate estimate of the true error rate (cf. WeKu91, P. 30). Therefore, the test sample error rate is used as the estimation of the true error rate (see Figure 3.1.1/1).

![Figure 3.1.1/1: Three kinds of error rates](image)

### 3.1.2 Sampling of train and test data

In order to make sure the test error rate is a reliable estimation of the true error rate, the cases in the train data set and test data set should be randomly taken from the underlying population. In addition, the cases in the two sample sets should be independent. Independent samples mean that there is no relationship among them other than that they are samples from the same population (cf. WeKu91, P. 28). Theoretically, to ensure that the samples are independent, they might be gathered at different times or by different investigators. Nevertheless, the common problem faced by the model builders is the lack of past examples. Therefore, the traditional way to sampling train and test data is: train and test data are both randomly taken from all available cases.

#### 3.1.2.1 Sampling for the adjustment of model structures

Sometimes, in order to get the best performance from an algorithm, the structure of the model or some parameters should be tuned, for example, the amount of pruning in a decision tree, the number of hidden nodes in the multi-layer perceptron neural network. To do this a separate test data set is needed to compare the performance of different model structures and chose the optimal one. Therefore, three samples are selected from the available cases: one for building the model, one for choosing the optimal structure and
parameters, and one for testing the final model (see Figure 3.1.2.1/1). In order to increase the number of cases for training the final model, sometimes after the optimal structure and parameters are chosen, the first two samples are used as the whole train data to build the final model (cf. Mich94, P. 110). One principle should be strictly kept: the sample for testing final model should not be used by the tuning of model structures in order to make sure that they are completely unseen when they are used to evaluate the final model.

![Figure 3.1.2.1/1: Three samples for training and testing](image)

3.1.2.2 Sampling for population drift

The accurate error rate estimation by the test sample is based on the assumption that the cases in the test sample are representative for the future real population on which the classification model will be applied. For real-world credit scoring problems, most populations change over time. The test sample error rate may not be the accurate estimation of the true error rate for the future changed population. This changing of population, which is also called population drift, violates the major assumption of the error rate estimation from the test sample.

Even if a model's test sample error rate is low to a satisfying level, it may not perform well later for the changed population. When the model is built for a drifting population, the
Criterions of the model evaluation

It is more important to rebuild the model, since the model becomes invalid as the population changes. If models cannot be rebuilt as frequently as the changing of population, the model building should not overfit but rather slightly underfit to the sample data.

To get a reliable model for a drifting population, an effective way to take test sample from available cases was proposed by Weiss/Indurkhya (cf. WeIn98, P. 43). As shown in Figure 3.1.2.2/1, if the available cases have a time attribute, i.e. they can be ordered by time, the training cases are taken from a period prior to a specified time, and the test cases are taken from the following period. The length of the time periods depends on the knowledge of the application. In this way the cases for the train and test samples are no longer viewed as random cases from the same population, a separation between the train and test samples is fixed. Because the population may change with time, we have a closer simulation of the task of prediction: train from the current period to predict for a future period.

Through sampling the test data from another time period the over-fitting problem is avoided in some measure. The built model can be thought to slightly underfit to the data.

![Figure 3.1.2.2/1: Train/Test procedure for nonstationary population (cf. WeIn98, P. 44)](image)

3.1.3 The size of train and test data

Generally speaking, the larger the train data the better the classifier, although the effects begin to diminish once a certain volume of train data is exceeded. And at the other hand, the larger are the test data, the more accurate is the error rate estimation. In the situations that a vast supply of data is not available, there is a dilemma here: to get a good classifier, we want to use as much of the data as possible for training; to get a good error estimate, we want to
use as much of it as possible for testing. Traditionally, a fixed percentage of cases are used for training and the remainder for testing. With insufficient cases, classifiers can not be effectively built, so the majority of cases are usually used for training. The usual proportions are approximately a 2/3, 1/3 split (cf. WeKu91, P. 30).

Usually we use the test sample error rate as the estimation of the true error rate and comparing the performance of two classifiers with their test sample error rates. If the difference of test sample error rates of classifier A and classifier B is obtained, is that difference significant, in other words, is the difference reliable or only due to the occasion of the particular test data? The answer depends on how many cases are used as test data. In fact, it can be statistically proved how many test data are enough in order to get reliable estimation of the true error rate.

Suppose there are \( n \) cases in the test data set, the error rate is estimated as:

\[
\hat{P}_{err} = \frac{\text{number of incorrectly classified cases}}{n}
\]

The random variable \( X \) is defined as:

\[
X_i = 1, \text{if the case is incorrectly classified}, \quad X_i = 0, \text{if the case is correctly classified}, \quad i = 1, 2, ... n.
\]

Suppose the probability of a case will be incorrectly classified is \( P_{err} \). The mean and variance of \( X \) is easy to obtain:

\[
\mu_{X_i} = l(P_{err}) + 0(1 - P_{err}) = P_{err}
\]

\[
\text{Var}_{X_i} = E[X_i]^2 - (E[X_i])^2 = P_{err}(1 - P_{err})
\]

It is easy to see that:

\[
\hat{P}_{err} = \frac{\sum_{i=1}^{n} X_i}{n}
\]

Due to the central limit theorem (cf. MiAr95, P. 327), for large \( n \), \( \hat{P}_{err} \) is approximately normally distributed with the mean and the variance:

\[
\mu = P_{err}
\]

\[
\text{Var} = \frac{P_{err}(1 - P_{err})}{n}
\]

Suppose two classifiers A and B are tested respectively over two independently drawn test data sets with \( n_A \) and \( n_B \) cases, then \( \hat{P}_{Aerr} \) and \( \hat{P}_{Berr} \) are two independently normally distributed
random variables with the means $P_{Aerr}$, $P_{Berr}$ and variances $P_{Aerr}(1 - P_{Aerr})/n_A$, $P_{Berr}(1 - P_{Berr})/n_B$ respectively.

Since the difference of two normal random variables is normal, the statistic $\hat{P}_{Aerr} - \hat{P}_{Berr}$ is at least approximately normally distributed with the mean $\mu_{\hat{P}_{Aerr} - \hat{P}_{Berr}}$ and variance $\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}}$.

Furthermore a standard normal random variable is obtained:

$$\frac{(\hat{P}_{Aerr} - \hat{P}_{Berr}) - \mu_{\hat{P}_{Aerr} - \hat{P}_{Berr}}}{\sqrt{\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}}}}$$

Our null hypothesis is that the mean is equal to zero:

$$H_0 : \mu_{\hat{P}_{Aerr} - \hat{P}_{Berr}} = 0,$$

which means there is no difference between the performances of the two models. To test such a hypothesis an approximately standard normal test statistic is (cf. WeIn98, P. 29):

$$Z = \frac{(\hat{P}_{Aerr} - \hat{P}_{Berr})}{\sqrt{\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}}}}$$

which is a normally distributed variable.

Since $\hat{P}_{Aerr}$ and $\hat{P}_{Berr}$ are independent, the variance are calculated as:

$$\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}} = \text{Var}_{\hat{P}_{Aerr}} + \text{Var}_{\hat{P}_{Berr}} = P_{Aerr}(1 - P_{Aerr})/n_A + P_{Berr}(1 - P_{Berr})/n_B .$$

Replacing $P_{Aerr}$ and $P_{Berr}$ with their unbiased estimators $\hat{P}_{Aerr}$ and $\hat{P}_{Berr}$ (cf. MiAr95, P. 327), we have the statistic:

$$Z = \frac{(\hat{P}_{Aerr} - \hat{P}_{Berr})}{\sqrt{P_{Aerr}(1 - P_{Aerr})/n_A + P_{Berr}(1 - P_{Berr})/n_B}}$$

If $\hat{P}_{Aerr} = \frac{1}{n} \sum_{i=1}^{n} X_{A_i}$ and $\hat{P}_{Berr} = \frac{1}{n} \sum_{i=1}^{n} X_{B_i}$ are obtained by using only one test set, i.e. they are dependent, the analysis requires a paired comparison. In this case, the joint variance $\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}}$ is estimated by a case-by-case computation.

$$\text{Var}_{\hat{P}_{Aerr} - \hat{P}_{Berr}} = \frac{1}{n} \text{Var}_{X_{A_i} - X_{B_i}}$$

Therefore, the Z-value is:
Criterions of the model evaluation

\[ Z = \frac{(\hat{P}_{err}^A - \hat{P}_{err}^B)}{\frac{1}{n} \sqrt{\sum_{i=1}^{n} ((X_A^i - X_B^i) - (\hat{P}_{err}^A - \hat{P}_{err}^B))^2}} \]

where \( X_A \) and \( X_B \) are the prediction for \( i \)th case in the test data by classifier \( A \) and \( B \) respectively, as defined by Formula 3.1.3/2.

The value of \( Z \) is compared with a critical value at the designated significance, generally 5% or 1% is used in practice, therefore the critical value is about equal to 2 (cf. WiFr00, P. 131; WeIn98, P. 29). If \( Z \) is larger than 2, then the null hypothesis \( H_0 : \mu_{P_{err}^A} = \mu_{P_{err}^B} = 0 \) is rejected, which means the difference between two classifiers' error rates is significant.

From the calculation of \( Z \), we can see that when larger test sample (larger \( n_A \), \( n_B \), and \( n \)) are used, larger value of test statistics \( Z \) is obtained, smaller differences are considered to be significant. Here we give an example to show the relationship between the size of test sample and the significance of the differences.

Suppose, we have one test data set with \( N \) cases. Two classifiers A and B are tested on it and have test sample error rates \( \hat{P}_{err}^A \) and \( \hat{P}_{err}^B \) respectively. If \( \Delta \hat{P} = \hat{P}_{err}^A - \hat{P}_{err}^B = 0.01 \), is this difference significant? To answer this question, the \( Z \) value should be calculated with Formula 3.1.3/14. We can transform the calculation as:

\[ Z = \frac{(\Delta \hat{P})}{\sqrt{\frac{a(1+\Delta \hat{P})^2 + b(1-\Delta \hat{P})^2 + (N-a-b)\Delta \hat{P}^2}{\sqrt{(a+b) + 2(b-a) \Delta \hat{P} + N\Delta \hat{P}^2}}}} \]

where 'a' is the number of cases that classified correctly by A, and incorrectly by B. 'b' is the number in the inverse situation.

We denote \( R = (a + b) / N \), then \( a + b = RN \), \(-RN = b - a = RN \). After some transformations of Formula 3.1.3/15 we have:

\[ \frac{N(\Delta \hat{P})}{\sqrt{RN + 2RN\Delta \hat{P} + N\Delta \hat{P}^2}} \leq Z \leq \frac{N(\Delta \hat{P})}{\sqrt{RN - 2RN\Delta \hat{P} + N\Delta \hat{P}^2}} \]

After some further transformations we get:

\[ Z \geq 2, \text{ when } |\Delta \hat{P}| \geq \frac{4R + 2\sqrt{R^2 - R(4 - N)}}{N - 4} \]
That means, if we set $Z$ to be 2, i.e. the significant level between 1% and 5%, the minimum significant difference of error rates is the function of $R$ and $N$. Figure 3.1.3/1 shows the minimum significant difference of two error rates for $N = 1000, 2000, 4000$ and $8000$.

Suppose less than 20% of the cases in test data are classified differently by two classifiers\(^2\) ($R = (a+b)/N \leq 20\%)$. It shows that if $N = 1000$, the difference of two error rates larger than 2.9% is significant. If $N = 8000$, the difference larger than 1% is significant\(^3\). In other words, when larger test data set is used, smaller difference is considered to be significant.

![Figure 3.1.3/1: Test sample sizes and statistic significance](image)

For example, suppose the performances of several classifiers are ordered by their test sample error rates, which have differences larger than 1% If the size of the test sample is larger than 8000, then the performance differences are believable. If the test sample is only 1000, this ordering of performances can not be trusted, i.e., if we test them on another test set, the relative ordering of these models might be changed, because they may be due to the particular occasion of the small test sample. In this case, other methods should be used to estimate true error rate, such as cross validation or bootstrap.

\(^2\) If $\hat{P}_{\text{Aerr}} + \hat{P}_{\text{Berr}} \leq 20\%$, then this supposition is realistic. Since $\hat{P}_{\text{Aerr}}$ and $\hat{P}_{\text{Berr}}$ are the percentages of misclassified cases by each classifier respectively, there are less than the percentage of $\hat{P}_{\text{Aerr}} + \hat{P}_{\text{Berr}}$ cases that are possibly classified differently by A and B. In other words, the maximum percentage of differently classified cases by A and B is $\hat{P}_{\text{Aerr}} + \hat{P}_{\text{Berr}}$.

\(^3\) However, the difference less than 1% may be or may not be significant according to the percentage of cases that are actually classified differently by two classifiers. If two classifiers classify cases similarly, only 5% of the cases are classified differently, then 0.5% difference is significant.
Therefore, the larger the test data, the more reliable is the evaluation of classifiers. Usually, large data sets are available for the practical application of data mining. Large data sets make both the model building and model evaluation based on a more reliable foundation. It is what the practical data mining applications are different from experiments done by researchers in laboratory with small data set.

### 3.2 Practicability criterions

When we consider choosing one from multiple classifiers as our final credit scoring model, the important criterion to be considered would be the classification accuracy. It is the direct and basic measure of model performance. The best model is the model that can predict the class of a new case most accurately.

Nevertheless, classification accuracy is not the only selection criterion in the credit scoring problem. Classification models derived from the data are not used in isolation but in conjunction with the practical decision making, often in context of the setting of credit decision rules. Therefore, the choice of models depends not only on the classification accuracy, but also on other criterions with respect to the practicability of models. Some other criterions of the model evaluation include:

- **The speed of the classification model.**
  This issue is in terms of computational time. The required time for both the training and the applying of the classification model should be considered. The speed of applying a classification model to a new case is important, since an instant decision is much more appealing to a potential borrower than having to wait for a longer time (cf. HaHe97, P. 535). With the drifting of the population, most models fail to be robust for long time. The time used to train a model is therefore important since model should be revised frequently. An extreme example is the k-nearest-neighbor method. The training is very fast (since no model is constructed, the classification occurs at the time of application), but the application of the method is slow (since all the cases in the train data should be used for the classification).

- **The transparency and interpretability of the classification model.**
  The transparency or interpretability of models has important meanings when the models need to be explained to the credit analysts. Sometimes the scores generated by an automatic model may be used as the reference for the final decision by experienced credit analysts. The understandability and reasonability of models should be acceptable to the credit analysts. Transparent models are those that are conceptually understood by the credit decision makers. One important desired character of the models is that there is a transparent linkage between input variables and the output so that one can see the
impact of each input variables on the output. An example of transparent models is the linear discriminant analysis, in which output is a linear function of each input variable. By contrast, models of neural networks act as black boxes, they are opaque and not able to provide simple clues about the basis for their classifications. The problem of the incomprehensibility of the neural networks hinders the acceptance by credit analysts in practice, especially, when the credit grantors need to explain to their customers why their credit applicants are refused (cf. Dink95, P. 54). The further efforts should be taken to explain why the models have reached their conclusion, such as sensitivity analysis.

The simplicity of the classification model.

When comparing the interpretability of different model methods, it may be a subjective matter which method is more understandable than the other. For example, someone may argue that a linear discriminant function is more transparent, others may argue that a decision tree is more transparent. However, it is reasonable to pursue the simplest possible model for a given method. In other word, for a given method, the simpler the model, the easier it is to be comprehended (cf. LiMo98, P, 98). For example, in the view of a credit analyst, a more understandable and succinct decision tree might be preferred than a complex decision tree. The Simplicity is closely related to the speed and the interpretability of models. If a model is simpler, then it is sometimes also easier to understand and faster to be applied.

The selection of classification models is a decision with multiple criterions. We pursue a more reliable classifier with higher accuracy, a faster classifier as well as an understandable classifier with simpler structure. When these intentions are contradicted with each other, we must tradeoff one for another. Usually, the classification accuracy is the foremost concern. But it is not the only concern. The importance of each criterion varies with different credit decision problems.

In some situations, when the credit decision rule is drawn from a classification model, the rule should be reasonably justified, and the justification should be transparent and interpretable to both the credit grantor and the credit taker. Both sides may require an explanation why the application is accepted or rejected.

In other situations, the speed of credit decisions is more important. For example, in order to reduce the attrition rate (the rate that customers change to other lenders) of an online-offered credit product, the speed of credit decisions is of same importance as accuracy, and the interpretability of models may be less important.
4 Methods of the model evaluation

4.1 Confusion matrix and two types of errors

A classification model gives a predicted class for every case in the test data set, the actual class of each of them is known. The confusion matrix provides a convenient way to compare the frequencies of actual versus predicted class for the test data set.

Two-class problem is the most common situation in credit scoring. The credit customers are usually classified into two classes: 'good' and 'bad' (or 'accepted' and 'rejected', 'default' and 'non-default'). The format of confusion matrix for two-class problem is shown in Table 4.1/1.

The notations are defined as:
- \(a\): the number of 'bad' customers that are predicted correctly.
- \(b\): the number of 'good' customers that are predicted incorrectly.
- \(c\): the number of 'bad' customers that are predicted incorrectly.
- \(d\): the number of 'good' customers that are predicted correctly.

<table>
<thead>
<tr>
<th>predicted class</th>
<th>actual class</th>
<th>number</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>bad</td>
<td>bad</td>
</tr>
<tr>
<td>bad</td>
<td>a</td>
<td>c</td>
<td>(c/(a+c)=a)</td>
</tr>
<tr>
<td>good</td>
<td>b</td>
<td>d</td>
<td>(b/(b + d) = \beta)</td>
</tr>
</tbody>
</table>

*Table 4.1/1: Confusion matrix for two-class problem*

A single-error rate summarizes the overall performance of a classification model is:

\[
\text{Total error rate} = \frac{(b+c)}{(a+b+c+d)}
\]

**Formula 4.1/1**

For credit scoring problem, two types of error rate should be considered (cf. Joos98, P. 65):

Type I error rate = \(\frac{c}{(a+c)}\)

Type II error rate = \(\frac{b}{(b + d)}\)

Type I error rate is also called a rate or credit risk, it is the rate of 'bad' customers being categorized as 'good'. When this happens, the mis-classified 'bad' customers will become default. Therefore, if a credit institution has a high a rate, which means the credit granting policy is too generous, the institution is exposed to credit risk.

Type II error rate is also called \(\beta\) rate or commercial risk, it is the rate of 'good' customers being categorized as 'bad'. When this happens, the mis-classified 'good' customers are
refused, the credit institution has therefore opportunity cost caused by the loss of 'good' customers. If a credit institution has a high $\beta$ rate for a long time, which means it takes a longtime restrictive credit granting policy, it may lose its share in the market (cf. Boge67, P, 113). The credit institution is therefore exposed to commercial risk.

For simplicity, the total error rate given by Formula 4.1/1 is often used as the criterion to compare different classification models. The assumption behind that is: two types of errors are of same importance. The importance of the two types of errors is obviously different in the credit scoring problem, because the correct prediction of 'bad' risks is more important due to the associated higher cost of mis-classifying a 'bad' risk. The total error rate is not a proper criterion to measure the performance of a classification model.

4.2 The tradeoff of two types of error rates

4.2.1 ROC Curve

Some classification models produce continuous scores as final results. For two classes’ problems, there is an important graphical technique for evaluating classification models that produce continuous scores --- ROC curve$^4$.

The scores produced by a classifier can be considered as the probability of a case belongs to a particular class. Suppose the cases are in the order: increasing scores correspond increasing of probabilities that the cases belong to class 'bad'. Thus the scores order the cases in the test data set. A threshold is to be determined in order to give a prediction of the class of a case. With the decreasing of the threshold value more cases are classified as 'bad'. That means, in general, with the decreasing of the threshold value Type I error rate would decrease and Type II error rate would increase, the same when vise verse. By varying the threshold value we have different values of Type I error rate and Type II error rate. ROC curve can be used to show the tradeoff of two errors at different threshold values.

Figure 4.2.1/1 shows an example of a ROC curve. The X axis of a ROC curve represents Type II error rate; Y axis represents 1-Type I error rate$^5$.

---

$^4$ ROC stands for "receiver operating characteristic" or "relative operating characteristic", which is used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel (cf. WiFr00, P. 141; Spac89, P. 160).

$^5$ A variation of ROC curve is used by other researchers, in which the X axis represents Type I error rate; Y axis represents Type II error rate (cf. Utho96, P. 86; Baet98, P. 14).
A ROC curve for the perfect classifier, which orders all 'bad' cases before 'good' cases, is the curve follows the two axes. It would classify 100% 'bad' cases into class 'bad' and 0% 'good' cases into class 'bad' for some value of the threshold. A classifier with an ROC curve which follows the 45° line would be useless. It would classify the same proportion of the 'bad' cases and 'good' cases into the class 'bad' at each value of the threshold; it would not separate the classes at all. Real-life classifiers produce ROC curves which lie between these two extremes.

By comparing ROC curves one can study the difference in classification accuracy between two classifiers. For a given Type I error rate the curve which has smaller Type II error rate will be superior. Similarly, for a given Type II error rate, the curve which has smaller Type I error rate will be better. The higher curve that is nearer to the perfect classifier has more accuracy. If and only if two curves touch or cross at a point will they give identical performance (at the threshold values corresponding to the point of intersection). All ROC curves intersect at the bottom left and top right corners of the square. Sometimes the curve for one classifier is superior to that for another at all values of the threshold; one curve is higher than the other throughout the diagram. A measure is given by the area under the ROC curve (denoted as AUC). The curve that has a larger AUC is better than the one that has a smaller AUC (cf. Hand97, P.133).

Sometimes, however, one curve is higher in some interval and another is higher in other intervals. The AUC criterion will not help us identify which is superior. In this case one can look to see which curve is higher within the range which is thought to be realistic for the problem at hand (cf. Hand97, P.134). In Figure 4.2.1/2, curve A is superior to B and C.
Whether curve B or curve C is superior depends on which range of threshold values are chosen.

\[ \text{Type II error rate} = 1 - \text{Type I error rate} \]

\begin{table}
\centering
\begin{tabular}{|c|c|c|}
\hline
A: & B: & C: \\
\hline
\end{tabular}
\end{table}

\textbf{Figure 4.2.1/2: Comparing ROC curves}

\subsection*{4.2.2 Cost function}

An optimal classification model is the one which minimizes the expected future loss when it is used to classify the credit customers. Under the two classes assumption the amount of loss will depend on the costs of two kinds of misclassification. The cost function is like follows (cf. Joos98, P. 65):

\[ \text{Expected cost} = P_g C_{\text{TypeI}} \text{ Type I error rate} + P_b C_{\text{TypeII}} \text{ Type II error rate}, \]  
\text{Formula 4.2.2/1}

where \( P_g, P_b \) = population proportion 'good' and 'bad' cases

\[ C_{\text{TypeI}}, C_{\text{TypeII}} = \text{Cost of Type I and Type II error} \]

The costs of building classification models are same for all cases. Without loss of generality, we can take this constant cost to be zero. We assume further that cases which are correctly classified incur no additional costs (cf. AdHa99, P. 1139). The cost function implies another assumption: there is no difference in the costs for each individual case. This assumption is usually violated in reality. Credits in reality may have different amount of principle or securities. However, the difference in the individual costs can be ignored since scoring models are usually built separately for different credit markets: large corporate loan, middle corporate loan, small business loan or a particular consumer credit product. Moreover, as will be explained in chapter 4.2.3, the costs of misclassification are often assumed to be a range of values, not a fixed value. And the determination of a threshold value may be flexibly according to different credit cases (see chapter 4.2.3). Therefore, although being not exactly agreed with the reality, this assumption is adopted commonly by researchers and practitioners.
The choice of $C_{TypeI}$ and $C_{TypeII}$ has a major effect on the evaluation of a classification model. However, the computation of the costs is often the most difficult problem in reality. The reason is the factors that affect the costs are difficult to be quantified.

For example, in the applicant scoring problem, $C_{TypeI}$ is the cost of granting a credit to a 'bad' customer. $C_{TypeII}$ is the opportunity costs of rejecting an application of a 'good' customer. Some components of $C_{TypeI}$ and $C_{TypeII}$ include:

<table>
<thead>
<tr>
<th>$C_{TypeI}$</th>
<th>+ loss of principal money</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- any income received during the process, such as interest income of the secured property</td>
</tr>
<tr>
<td></td>
<td>- value of secured property at the time of liquidation</td>
</tr>
<tr>
<td></td>
<td>+ other costs such as legal costs, increased administrative costs, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$C_{TypeII}$</th>
<th>+ loss of interest income with the 'good' customer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+ loss/ -profit on alternative customers</td>
</tr>
</tbody>
</table>

Table 4.2.2/1: Components of Type I error cost and Type II error cost

The second component of $C_{TypeII}$ is based on the assumption that the lender will not keep the unloaned money idle and will lend it to another customer. The profit on the alternative customer decrease $C_{TypeII}$, the loss on the alternative customer increase $C_{TypeII}$ (cf. Natu97, P. 289).

The practical calculation of misclassification costs may be more complicated. In practice, credit decision and management is a complex process, in which many factors involved that affect the calculation of the misclassification costs. Some complex examples that may happen in practice are:

- When credits are granted for more than one year, the costs should be calculated dynamically in multi-periods to reflect the inflow and outflow of money in different years (cf. Natu97, P. 289).

- When a customer is rejected according to the result of the scoring model, it is reexamined by credit analysts and credit is granted to this customer, perhaps with new conditions. The cost of reexamination and the probability of the correct decision by credit analysts should be considered in the cost function (cf. Utho97, P. 68).

- When a customer fails to pay up the debt on time, but it is paid off finally through collection. The cost of collection should be considered.

- When a behavior scoring model is built, which assesses the degree of risks of existing creditors, the aim is to identify existing risky creditors and to take proper actions to them in order to avoid possible loss. Therefore, the costs of making wrong decisions would

---

6 + denotes this component increases the cost, - denotes this component decreases the cost.
include the cost of taking actions on 'good' customers and the loss that results from not taking actions on 'bad' customers.

There are still other complex situations that add difficulties and uncertainties to the calculation of the costs of misclassification. Due to the difficulties and uncertainties, usually the credit institutions cannot give precise estimates of the cost of misclassification. Moreover, the relative misclassification costs are likely to change over time as the economy evolves. Therefore, although it is possible that a range of likely costs can be given, it is usually improbable that exact value of costs can be given (cf. AdHa00, P. 307).

4.2.3 The determination of the threshold value

The advantage of ROC curves is that they visualize the possible tradeoffs between Type I error rate and Type II error rate for classification models and allow the end users to choose the appropriate threshold value. But this gives the model users an additional task: how to choose the optimal threshold value?

Even if the models are used flexibly without fixed threshold value, a possible range of threshold values still need to be determined. When we use ROC curves to compare classification models, the area under ROC (AUC) is worthless in the situation that two ROC curves are crossed with each other (see Figure 4.2.1/2). To solve this problem, the likely values of threshold must be determined in order to know which segment of ROC curves are to be compared.

A ROC curve can give the optimal threshold value or a range of possible threshold values when the cost function can be determined.

Given a classification model that outputs scores for each case in the test set, theoretically, if the costs of the two different errors and the population proportion of 'good' and 'bad' cases are definite, the optimal threshold point between 'good' and 'bad' cases can be determined through minimizing the expected cost.

Without changing its role as comparison criterion, the expected cost (see Formula 4.2.2/1) can be re-expressed as:

\[
\text{cost criterion} = 1 + \left( \frac{C_{\text{TypeII}} \cdot P_g}{C_{\text{TypeI}} \cdot P_b} \right) \times \text{Type II error rate} - (1 - \text{Type I error rate}).
\]

Formula 4.2.3/1

We denote:

\[
CR = \frac{C_{\text{TypeI}}}{C_{\text{TypeII}}},
\]

Formula 4.2.3/2

\[
PR = \frac{P_g}{P_b}.
\]

Formula 4.2.3/3
then the function of Formula 4.2.3/1 is a group of parallel lines in the space of ROC curves (see Figure 4.2.3/1) which have the slope:

\[ S = \frac{C_{\text{Type II}} P_g}{C_{\text{Type I}} P_b} = \frac{PR}{CR}. \]  

\[ \text{Formula 4.2.3/4} \]

The line that is tangent with the ROC curve achieves the minimum cost. The optimal threshold value that minimizes the cost function can be found at the tangent point of the ROC curve and the cost line (point T), and the Type I error rate, Type II error rate at this threshold value can be determined.

As indicated in chapter 4.2.2, precise costs \( C_{\text{Type I}} \) and \( C_{\text{Type II}} \) may not be known in a given real application. However, only the cost ratio \( CR \) is relevant with the slope of the cost line, and although it is difficult to quantify \( C_{\text{Type I}} \) and \( C_{\text{Type II}} \), the ratio of them are often naturally to be acquired from domain experts (cf. AdHa99, P. 1141). The possible range of the cost ratio \( CR \) (it must be greater than 1) may be given by the experts in the application domain. For example, an expert may be able to say that Type I error is between 10 and 40 times as serious as Type II error, i.e. the cost ratio \( CR \) lies in the interval \([10, 40]\). Thus, if the 'good'-'bad' proportion (PR) in the total population\(^7\) is estimated as 10, a range of possible value of \( S \) can be determined as \([1, 1/4]\). Figure 4.2.3/2 shows the two minimum cost lines corresponding to \( S_1 = 1 \) and \( S_2 = 1/4 \) and the corresponding optimal threshold values \( T_1 \) and \( T_2 \).

\(^7\) In some researches, the numbers of 'good' and 'bad' cases are derived from the training sample (cf. Schu92), which assumes implicitly that the proportions of good-bad cases in the training sample reflect those in the population (cf. Mich94, P. 177).
Methods of the model evaluation

The threshold value decides which credit cases are classified as 'good', which are classified as 'bad'. If one threshold value $T$ can be determined by choosing fixed cost ratio and population proportion, then the credit cases with the score greater than $T$ are classified as 'bad', other as 'good'. If two threshold values $T_1$, $T_2$ are determined by a range of cost ratios, the cases are divided into three segments (cf. Hofm90, P. 943): segment 1: score $< T_1$, segment2: $T_1 < \text{score} < T_2$, and segment 3: score $> T_2$. The credit cases in the first segment are 'good', in the third segment are 'bad'. Others are ambiguous and would be analyzed and decided manually by a credit expert.

Although exact cost function is used in academic research in order to determine the optimal threshold value, in practical application the determination of a threshold value may be more flexible. Varying threshold values may be used for practical credit decisions. Here are some possible practical ways of selecting flexible threshold values:

- The threshold value can be determined indirectly by determining the number of applicants that should be accepted or determining the maximum allowable probability of default (cf. RoGl94, P. 592).

- Different threshold values may be used for different credit cases. For example, threshold values may be selected according to the amount of the credit and its securities' value, or according to the sector of industry to which the credit taker belongs (cf. Feid92, P. 216).

- The selection of a threshold value can also be affected by the changed risk attitude of credit decision makers. The cost ratio is sometimes regarded as a subjective factor and depends on the risk behavior of the decision maker and his or her attitude about the cost of different type of error (cf. Joos98, P. 65).

- Changed factors in the macroeconomic environment may also change the selection of threshold values. In different economical conditions, the credit policy may be more or less stringent, the selected threshold value may be higher or lower.
Methods of the model evaluation

From the discussions above we know that the determinations of the cost function as well as of the optimal threshold value are sometimes uncertain in practice. In the following example study, precise estimations of costs are not given, we only assume some values of the slope of cost function $S$ to determine the threshold values.

4.3 Incremental analysis

4.3.1 Learning curve

Learning curve is a graphical tool to show the changing performance of a series of classification models. The changing trend is illustrated in a curve.

Possible factors that can change continuously and result in the continuous changing in the performance of models are:

- The size of the train sample. It can be denoted by the number or the percentage of the cases in the sample. For example, 100, 250, 500, 1000, 1500, 2000 cases, or 10%, 25%, 50%, 80%, 100% of all cases are used as the train sample.

- The model parameters. Some model parameters represent the structure of models. For example, the size of decision trees; the number of hidden nodes in neural networks.

- The number of input variables. Sometimes a feature selection algorithm orders the independent variables according to a particular criterion. In this order they are added one by one to the subset of input variables. A series of models can be trained with these subsets of input variables that have increasing number of predictors.

In a learning curve the performance of models is usually denoted by the total error rate. With the help of a learning curve, the changing trend of the prediction accuracy caused by a particular changing factor can be observed intuitively.

Following three sections introduce the method of using learning curves to analyze the changing of the classification accuracy due to the change of train sample sizes and model structures.

4.3.2 Incremental case analysis

Incremental case analysis is the way to show the changing error rates of a series of classification models generated by using train samples with increasing sizes (cf. Weln98, P. 173).

The goal of incremental case analysis is to find the least sample size to train model. If the same quality of models can be generated by training with fewer cases, then the smaller train sample will be used. The smaller train sample can speed up learning, especially for computation-extensive methods such as neural networks, the speedup is dramatic. For some
real world application, when large number of examples are available, which is usually the case in credit scoring due to the mass customers, it may be not necessary to use all the cases to find the best solution if the model with fewer cases can induce exhaustively the concepts in the data set.

Incremental case analysis uses learning curves to monitor the incremental change in errors and illustrate the potential for additional gains in performance when the sample size is increased. Usually the test error rate will decrease as the number of train cases increase. However, the degree and speed of the decrease vary greatly with different data sets. For some data sets that imply more complex concepts, the error rate may be decreased significantly and continuously when increasing training cases are used; for others data sets with simple concepts, the error rate may decrease at first and stop decreasing once a certain amount of training cases is exceeded.

4.3.3 Incremental complexity analysis

Incremental complexity analysis is the way to show the changing error rates of a series of classification models generated with different model sizes (cf. Weln98, P.176). Model size is here the synonym of complexity or the capacity of models. For example, the larger the decision tree, the more complex it is; the more the number of hidden nodes in a neural network, the more complex it is. Model complexity can be adjusted by tuning model parameters.

The goal of incremental complexity analysis is to find the right size solution. As shown in Figure 4.3.3/1, the optimal model size is the one that neither under-fits nor over-fits the data. If the model is too small, it will not fit the data enough well with its limited generalization power. If the model is too large, it will over-fit the data and lose generalization power by incorporating too many accidents or outliers in the training data not shared by other data sets (cf. GaTa97, P. 11).

Therefore, with the increasing of model sizes the test error rate decreases at first, at some point the model becomes too large, producing over-fitting which results in the increasing of error rates.
4.3.4 The basic phenomenology of incremental analysis

For some algorithms, the sample size and the model size have this relation: the larger the sample size, the more complexity the found model, i.e. as the number of train sample cases increases, the model size tends to increase. Therefore, the incremental case analysis should assume a fixed model size; the incremental complexity analysis should assume a fixed sample size. Figure 4.3.4/1 and Figure 4.3.4/2 describe the basic phenomenology of incremental analysis with learning curves. For a fixed model size, as the training data increase, the test error rate decreases and get to an asymptotic value when the sample size is large enough. For a given sample size, as the model size increases, the test error rate decreases at first, after reaches an optimal model size it begins to increase.

Figure 4.3.3/1: Test error tradeoff in terms of model size (cf. GaTa97, P. 10)

Figure 4.3.4/1: Learning curve for incremental case analysis
The following phenomena may happen when an incremental analysis is carried out. Although not always being inviolate, they are observed in many applications (cf. Weln98, P. 180):

- When the size of the train sample is increased, test error rate decreases significantly and no evidence of reaching an asymptotic value, it may indicate that the concept in the underlying population has not been fully extracted, the model can be improved by using more cases.

- When the size of the train sample is increased, test error rate stops decreasing for a fixed model size, it may indicate the model size should be increased to increase the learning ability of the model.

- When the size of the train sample is increased, test error rate stops decreasing for any increased model size, significant gains in performance may be not achieved by either increasing the train sample size or increasing the model size. The model is reached its full ability to extract knowledge from the data set at hand.

Using incremental analysis helps answer a key question for mining large data set: how many useful concepts can be extracted from the data with the least learning cost in terms of computation time. Suppose two solutions are close in classification accuracy, the less-complex solution that learned from a smaller sample has some advantages: more explanatory, less variable performance on new data and more rapid to be trained or applied to new data. Through the incremental analysis, simpler, faster and more robust classification models can be found.
5 An example of the credit scoring model evaluation

In this chapter, different methods are used to develop scoring models for a real world credit risk assessment problem. Five classification algorithms are evaluated and compared:

- Linear discriminant analysis (LDA)
- Logistic regression (LR)
- K-nearest-neighbors (k-NN)
- Model tree algorithm --- (M5)\textsuperscript{8}
- Multi-layer perceptron neural network (MLP)

The details of these classification algorithms were provided in Liuy02, Chapter 4. They are representative for three professional backgrounds: statistics, machine learning and neural networks. The process of evaluating the generated classification models are described, which utilizes various model evaluation criterions and methods introduced earlier in this paper.

5.1 Problem definition and data preparation

5.1.1 Definition of the objective of scoring models and the risk classes

The data are obtained from a German credit insurance company. The kind of product provided by this insurance company is as this: A company supplies goods or services to its clients. The clients do not pay for the received goods or services immediately but will pay later according to the contracts. The supplier asks for the insurance company to insure the payment of its clients. The granted insurance is related with the creditworthiness of the clients of the suppliers. These clients are called the risk partners of the insurance company.

The business problem is how to decide the insolvent probability of the risk partners and to predict their future payment behavior using their available credit information.

The company's current risk assessment system\textsuperscript{9} collects different resources of credit information. The main information is business information which comes from a credit

---

\textsuperscript{8} M5 is an algorithm that can produce model trees. The algorithm generates a conventional decision tree with linear regression models at each of its leaves. M5 has been developed by Quinlan (cf. Quin92, P. 343); The implementation of the algorithm used in this paper is described in WaWi97 (cf. WaWi97, P. 1). Model trees produce continuous numeric values. They can be applied to classification problems by treating the continuous values as the approximated probability of belonging to a class (cf. Fran97, P. 63).

\textsuperscript{9} The current risk assessment system consists of several components, such as knowledge-based systems and other methods to assess different information concerning the credit worthiness of the risk partner. The system is described in HeSc99 and Müll96.
information agency. The system assigns a rating to every collected credit information of a risk partner. The ratings are in four levels: A, B, C, D (see Table 5.1.1/1).

<table>
<thead>
<tr>
<th>A</th>
<th>Very good</th>
<th>Companies that have above-average good creditworthiness and very low insolvent probability.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Rather good</td>
<td>Companies that have good or satisfied creditworthiness and low insolvent probability.</td>
</tr>
<tr>
<td>C</td>
<td>problemmatic</td>
<td>Companies that are problematic and have high risk of insolvency.</td>
</tr>
<tr>
<td>D</td>
<td>Very bad</td>
<td>Companies that can not be insured.</td>
</tr>
</tbody>
</table>

Table 5.1.1/1: The ratings of companies given by the current system (cf. Müll96, P. 81)

The domain experts believe that companies with rating A/B have rather good creditworthiness and positive credit decisions can usually be made on them, while companies with rating D are very risky companies that can not be insured. Companies with rating C are those ambiguous but suspiciously problematic. Although C companies are problematic, positive credit decisions should be made on some of them and negative on others. A quite large proportions of companies are rated as C, therefore, the focus and the difficulty of the credit decisions exist in the identification of the risk level of each C companies.

The first idea is whether the companies can be classified into several clusters with the unsupervised classification techniques (clustering algorithms) in order that each cluster represents a certain level of default probability. The clustering algorithms can group the similar credit cases by comparing the similarity of different credit cases. The data requirement of clustering algorithms is easy to be satisfied, since the class membership of cases in the data set need not to be provided. However, due to the fact that the learning is unsupervised, the results of clustering algorithms are often difficult to be controlled.

As an example, simple k means clustering algorithm is applied to all the available data. The number of clusters are denoted before applying the algorithm. If the cases are classified into four clusters and three clusters, the frequencies of cases from different clusters and different ratings are given in the cross tables (see Table 5.1.1/2 and Table 5.1.1/3).

One problem of clustering algorithms is the explanation of the results. Whether the clusters represent different risk levels is unknown. Although a quite large percent of D companies are clustered into Cluster 3 and 4, there are still some D cases are classified into Cluster 1 and 2. It is not reasonable to conclude that cases in cluster 1 have lower credit risk and in Cluster 3 and 4 have higher credit risk.

Another problem of clustering methods exists in the evaluation of the clustering results. The evaluation criterions of clustering results measure the similarity within each cluster and the
dissimilarity between different clusters. The definition of similarity are various with various algorithms. Using different clustering algorithms with different algorithms’ parameters, different results can be obtained. Which clustering result should be used can not be decided. No supervisor during the building of the models means no criterions for the evaluation of the usefulness of models.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>% within Rating</td>
<td>67.0%</td>
<td>33.0%</td>
<td>100.0%</td>
<td>67.0%</td>
<td>33.0%</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>20712</td>
<td>17163</td>
<td>10454</td>
<td></td>
<td>48329</td>
</tr>
<tr>
<td>% within Rating</td>
<td>42.9%</td>
<td>35.5%</td>
<td>21.6%</td>
<td>42.9%</td>
<td>35.5%</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>3382</td>
<td>3117</td>
<td>4869</td>
<td></td>
<td>11366</td>
</tr>
<tr>
<td>% within Rating</td>
<td>29.8%</td>
<td>27.4%</td>
<td>42.8%</td>
<td>29.8%</td>
<td>27.4%</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>1224</td>
<td>1205</td>
<td>3238</td>
<td></td>
<td>5667</td>
</tr>
<tr>
<td>% within Rating</td>
<td>21.6%</td>
<td>21.3%</td>
<td>57.1%</td>
<td>21.6%</td>
<td>21.3%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>38.7%</td>
<td>32.9%</td>
<td>28.4%</td>
<td>100.0%</td>
<td>65367</td>
</tr>
</tbody>
</table>

Table 5.1.1/2: Crosstabulation of three clusters and ratings

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>2</td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>% within Rating</td>
<td>67.0%</td>
<td>33.0%</td>
<td>100.0%</td>
<td>67.0%</td>
<td>33.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>20516</td>
<td>17351</td>
<td>10460</td>
<td>2</td>
<td></td>
<td>48329</td>
</tr>
<tr>
<td>% within Rating</td>
<td>42.5%</td>
<td>35.9%</td>
<td>21.6%</td>
<td>42.5%</td>
<td>35.9%</td>
<td>21.6%</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>3376</td>
<td>2940</td>
<td>4917</td>
<td>135</td>
<td></td>
<td>11368</td>
</tr>
<tr>
<td>% within Rating</td>
<td>29.7%</td>
<td>25.9%</td>
<td>43.3%</td>
<td>29.7%</td>
<td>25.9%</td>
<td>43.3%</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>805</td>
<td>111</td>
<td>2586</td>
<td>2165</td>
<td></td>
<td>5667</td>
</tr>
<tr>
<td>% within Rating</td>
<td>14.2%</td>
<td>2.0%</td>
<td>45.6%</td>
<td>14.2%</td>
<td>2.0%</td>
<td>45.6%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>37.8%</td>
<td>31.2%</td>
<td>27.5%</td>
<td>3.5%</td>
<td>100.0%</td>
<td>65367</td>
</tr>
</tbody>
</table>

Table 5.1.1/3: Crosstabulation of four clusters and ratings

Experiments with the clustering algorithms showed that unsupervised classification methods can not give a satisfying solution for the problem at hand. Therefore, the supervised classification methods that can build scoring models are chosen as the way of solution. To build scoring models, the available cases should have known classes. The known classes of the cases in the data set are important, since the known classes of the train cases are the supervisor for the building of the models and the known classes of the test cases constitute the measure of error rates to evaluate the generated models.
The usual way to build scoring models is using data with two classes ("good" and "bad" cases). The algorithms used in this study generate continuous scores that represent the probabilities of belonging to a class. The results are not binary good/bad models but are described with continuous scores. The input data are classified into two classes, while the outputs are continuous scores (see Figure 5.1.1/1).

Because the actual payment behaviors of the available cases is not known, the classification of "good" cases and "bad" cases are decided according to the ratings given by the current risk assessment system of the insurance company. If we divide the cases in the data set into A/B ("very good" and "rather good") and D ("very bad") two classes, the generated models are trivial and no use in reality. For example, the generated decision tree model consists of only one root node: "ZAHLWEISE" ≤ 43 and > 43. The reason is that the D companies are easy to be identified. Having only one input variable “ZAHLWEISE”\(^{11}\), the D companies are already identified with nearly 100 percent accurate rate (see Figure 5.1.1/2).

---

\(^{10}\) It is a common problem in practice that the available cases with actual classes are not enough for model building, especially for the newly established system for which old customers’ data do not exist (cf. Heno83, P. 158). In this situation, one alternative solution is the manual classification. The credit customers are evaluated by experts and classified into different risk levels (cf. Krah98, P. 109). Although the trained models with these cases does not strictly conform to the assumption of predictive scoring model (The assumption of predictive scoring model is learning prediction rules from past cases with known payment behaviors), this kind of solution can be realized in practice when cases with true classes are not available.

\(^{11}\) The meaning of the variable “ZAHLWEISE” refers to Appendix A. Other variables are given in chapter 5.1.2.
The class boundaries between A and B as well as between B and C companies are overlapped. Since there is only a very small number of A companies in the data set (3 cases in the sample), the division between A and B will not be considered. In the following analysis, the available cases are classified into A/B ("very good" and "rather good") and C/D ("suspiciously bad" and "very bad") two classes (for simplicity, termed as "good" and "bad" two classes).

It will be illustrated in the analysis in chapter 5.4 that the models resulted from this class' definition (learned from such a data set) give a useful solution:

1) At one hand, the models can find the patterns of A/B companies and the patterns of D companies. The "rather good" and "very bad" companies will be identified with high accuracy.

2) At the other hand, the models can find the bad companies and the good companies in the C group. As shown above, the models' results are continuous scores, the scoring models will not only give a binary good/bad decision on a case. The problem of deciding the credit standing of the C companies is solved as this: the C companies with higher scores are those that have the similar patterns as the D companies, and those with lower scores have the similar patterns as the A/B companies. Thus, the credit decisions on the C companies can be made according to the scores derived from the models.

5.1.2 Preparation of the data

The underlying data base holds credit information for 145978 risk partners from 01.01.2000 to 02.03.2001. Excluding those cases that have not been rated or have too many missing values, we get 65367 available cases. Among them there are 48332 'good' partners (with ratings A and B) and 17035 'bad' partners (with ratings C and D).
The available cases are divided into two time segments: cases from 01.01.2000 to 02.11.2000, cases from 03.11.2000 to 02.03.2001. Train samples are selected from the first segment of time, while test samples are selected from the second segment of time. This way of sample selection insure that the train and test sample are independent with each other.

An incremental analysis is firstly carried out. Then final models are built with the optimal structure and the proper size of train sample. This evaluation process is described in details in Chapter 5.2. Two test samples are needed in this process, because the test data for evaluating the final models should be completely unseen at the time of the model building (see Chapter 3.1.2.1). A train and a test data are randomly selected from the first time segment for incremental analysis. A second test data is randomly selected from the second time segment to test the performance of the final models. Three data samples are drawn from the population (see Figure 5.1.2/1).

Due to the low prevalence of 'bad' cases in the available cases, a random selection will result in a sample that with low proportion of 'bad' cases. It was found that the priori probabilities for 'good' and 'bad' cases in the train data set have influence on some model methods. It was argued by other researchers (cf. Lohr94, P. 151; Joos98, P. 60) and also validated by preliminary mining experiences with this data set that low number of 'bad' cases leads to inaccurate parameter estimations in the models. The good result can be achieved when using balancing (cf. FrHo98, P. 189). Balancing means using the same number of 'good' and 'bad' cases in the train data set.
'bad' cases in the train sample. Figure 5.1.2/1 illustrates the process of sampling three balanced samples.

The credit information for these risk partners are the business information from the credit information agency 'Die Vereine Creditreform'. After necessary transformations of the original data and recreations of some new variables, 14 input variables are used as predictors for the model building. They are shown in Table 5.1.2/1.12

The categorical variables are transformed into binary variables for the algorithms that require numerical input variables (LDA, LR, MLP and M5). The input variables are normalized for MLP so that the values are between -1 and 1. The normalization of input variables can improve the performance of a MLP neural network (cf. Krau93, P. 144).

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAMMKAP</td>
<td>numeric</td>
<td>Share capital</td>
</tr>
<tr>
<td>LUMSATZ/</td>
<td>numeric</td>
<td>Newest annual sales/ Number of employees.</td>
</tr>
<tr>
<td>MITARBEITER</td>
<td>numeric</td>
<td>Number of years since the year of LUMSATZ</td>
</tr>
<tr>
<td>JAHRNUM</td>
<td>numeric</td>
<td>Change of the annual sales from second year to the third year in percentage</td>
</tr>
<tr>
<td>UM23</td>
<td>numeric</td>
<td>Change of the annual sales from first year to the second year in percentage</td>
</tr>
<tr>
<td>UM12</td>
<td>numeric</td>
<td></td>
</tr>
<tr>
<td>IS_BELAST</td>
<td>categorical</td>
<td>Whether the real estate is mortgaged</td>
</tr>
<tr>
<td>BELAST</td>
<td>numeric</td>
<td>The percentage of the mortgaged real estate</td>
</tr>
<tr>
<td>ZAHLWEISE</td>
<td>numeric</td>
<td>Index of mode of payment</td>
</tr>
<tr>
<td>BONITAETINDEX</td>
<td>numeric</td>
<td>Index of creditworthiness</td>
</tr>
<tr>
<td>GRUNDJAHR</td>
<td>numeric</td>
<td>Number of years since the risk partner is established</td>
</tr>
<tr>
<td>AUFTRAGE</td>
<td>numeric</td>
<td>Index of order-book situation.</td>
</tr>
<tr>
<td>ENTWICKLUNG</td>
<td>numeric</td>
<td>Index of company development</td>
</tr>
<tr>
<td>RECHSTFORM</td>
<td>categorical</td>
<td>Legal form</td>
</tr>
<tr>
<td>BRANCH</td>
<td>categorical</td>
<td>Sector of industry</td>
</tr>
</tbody>
</table>

Table 5.1.2/1: List of input variables

5.2 The process of the evaluation

The process of evaluation consists of two steps: incremental analysis and comparison analysis.

---

12 The meanings of variables 'ZAHLWEISE', 'AUFTRAGE', 'ENTWICKLUNG', and 'BONITAETINDEX' refer to Appendix A.
5.2.1 Incremental analysis

Using learning curves to do incremental case analysis and incremental complexity analysis helps to analyze the ability of models on extracting knowledge from the available data sample.

Six percentages of the train sample are used for incremental case analysis: 10%, 20%, 33%, 50%, 67% and 100%. For each percentage, the cases are randomly selected but the 'bad'- 'good' proportions in each of them are kept on 1:1. Each algorithm is trained with these six sub-train-samples and tested on the test sample 1.

In the incremental complexity analysis the following parameters that change the model structures will be considered.

- For the LDA and LR algorithms, no parameters need to be tuned. Model options are set to be their default values.
- For the model tree algorithm M5, parameter F (the pruning factor) is to be decided. Trees with different complexities are generated by varying this parameter. Different size of model trees can be obtained. When F is set to be larger, simpler model trees will be generated, vice verse. Model trees are trained for \( F = 0, 0.5, 1, 2, \ldots, 12 \).
- For K-nearest-neighbors method, the number of the nearest neighbors \( k \) is adjusted. Performance are evaluated for \( k = 1, 3, 5, \ldots, 31 \).
- For multi-layer propagation network, the network structure should be determined by the number of hidden layers and the number of hidden nodes in each layer. Only one hidden layer is used in this analysis\(^\text{13}\). Models are built for \( H \) (the number of hidden nodes) = 1, 2, \ldots, 12\(^\text{14}\).

Models are trained with different sample sizes, different algorithms and different model parameters. One optimal model structure and size of train sample is chosen for each algorithms based on the analysis of performances on the test sample 1 with the help of learning curves. Figure 5.2.1/1 shows the process of the incremental analysis.

\(^{13}\) It has been proved by many researchers that one hidden layer structure is complex enough for many real world problems (cf. Hech91, P. 131; Baet95, P. 22-23; Krus99, P. 170).

\(^{14}\) MLP models have other parameters, learning rate (set to be 0.3), momentum (set to be 0.2), time of iterations (set to be 3000). 20% of the train sample are used as validation set. The training will continue until it is observed that the error on the validation set has been consistently getting worse (20 times), or if the time of iterations is reached. Moreover, if the network diverges from the answer the software will automatically reset the network with a lower learning rate and begin training again.
5.2.2 Comparison analysis

Using the optimal model structure obtained from the incremental analysis described in chapter 5.2.1, optimal models for the five algorithms are built with proper sizes of train sample. Each model is tested with the test sample 2. The overall process of comparison analysis shown in Figure 5.2.2/1 will create following results.

1. Output of models: The classification outputs of each model are firstly derived. Outputs of models include the true class (C), the predicted class (c), and the predicted score (p) for each case in the test sample 2.

2. ROC: The values of points in ROC curves are then obtained. They are Type I error rate ($E_1$) and Type II error rate ($E_2$) for every possible threshold value (see Chapter 4.2.1).

3. AUC: The values of AUC (area under ROC curve) are calculated. (see Chapter 4.2.1).
4. Models' error rates: If the slope of cost functions $S$ is supposed, the threshold values are decided for each model. The corresponding Type I error rate ($E_1$), Type II error rate ($E_2$) can be obtained (see Chapter 4.2.3).

![Figure 5.2.2/1: Process of the comparison analysis](image-url)
5.3 Results of the evaluation

5.3.1 Results of the incremental analysis

5.3.1.1 Linear discriminant analysis and logistic regression

The incremental case analysis shown in Figure 5.3.1.1/1 and Figure 5.3.1.1/2 indicates that after 33% of the train sample cases are used, further increase of train data can not increase classification accuracy significantly.

The conclusion of the analysis is: The model performance cannot be improved by using more training cases.
5.3.1.2 Neural networks

The incremental case analysis of MLP (see Figure 5.3.1.2/1) indicates that for some network structures, the error rate is not decreased monotonously when the size of train sample is increased. However, the general trend agrees with the expected phenomena: decreasing error rates with increasing size of train samples.

The conclusion of the analysis is: 1. The optimal model structure is the network with 3 hidden nodes. 2. Due to the unstability of neural networks, the error rates is not always monotonously decreased when increasing train data are used. From the trend of the learning
curves it can be predicted that significant improvement in accuracy will not occur by using more cases than the complete train sample.

5.3.1.3 K-nearest-neighbors

![Graph of total error rate vs train sample size for different k values.](image)

**Figure 5.3.1.3/1: Incremental case analysis of k-nearest-neighbors**

The incremental case analysis of k-NN illustrated in Figure 5.3.1.3/1 proves the expected phenomena: the error rate is decreased with the increased sizes of the train sample. To show the further decreasing of error rates, twice of the train sample are reselected from the available cases and used to train models (see points of 200%).

![Graph of total error rate vs k for different data points.](image)

**Figure 5.3.1.3/2: Incremental complexity analysis of k-nearest-neighbors**

The incremental complexity analysis of k-NN is illustrated in Figure 5.3.1.3/2. The trends of lines show that as the parameter k is increased, the error is decreased at first. However, when k is larger than 7, increasing of k can not decrease error rates but rather increases error rates slightly. This result conforms to the expected shape of learning curve for incremental complexity analysis (see Chapter 4.3.4, Figure 4.3.4/2).

The conclusion of the analysis is: 1. The model performance can be improved by using more training cases. 2. The model parameter k should be set to 7.
5.3.1.4 Decision trees

The model complexity of model trees is decided by the pruning factor (parameter $F$ of the algorithm M5). If $F$ is set to 0, the largest tree is obtained. The error rates with different extents of pruning are shown in Figure 5.3.1.4/1. From the overall trend, it can be seen that as the tree grows (pruning factor is getting smaller), the error rate decreases at first, after reaches a lowest value it stops decreasing. When the pruning factor is smaller than 1 the error rates increases slightly.

Figure 5.3.1.4/2 shows the error rates of model trees obtained by using different percentage of train sample. The error rates decrease significantly when the train sample size increases from 10% to 33%. The decrease in the error rates is still observed (although not very significant) after 67% of the cases in the train sample are used. However, when the size of train sample increases to 200%, there is no evidence of decreasing in error rates.

The conclusion of the analysis is: 1. The model performance can not be improved significantly by using more training cases. 2. The simplest model tree using 100% train sample that has the lowest error rate is constructed with $F = 11$. 
5.3.2 Results of the comparison analysis

Based on the analysis in chapter 5.3.1, the proper model structures for each algorithm are obtained. To compare the performance of different classification algorithms, the following 5 models are trained:

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Model structures</th>
<th>Train sample sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>/</td>
<td>3300 cases</td>
</tr>
<tr>
<td>LR</td>
<td>/</td>
<td>3300 cases</td>
</tr>
<tr>
<td>M5</td>
<td>F = 11</td>
<td>14000 cases</td>
</tr>
<tr>
<td>MLP</td>
<td>H = 3</td>
<td>14000 cases</td>
</tr>
<tr>
<td>k-NN</td>
<td>k = 7</td>
<td>26430 cases</td>
</tr>
</tbody>
</table>

*Figure 5.3.2/1: The model structures and train sample sizes of five models*

For LDA and LR, only 33% of the train sample are used, since more train cases cannot improve the model performance. For M5 and MLP, models are trained with the merged data set of the train sample and the test sample 1 (7000 'bad' and 7000 'good' cases). For k-NN, because the incremental cases analysis showed the potential of improvements in accuracy with larger train samples, models are trained with a reselected larger sample with 13215 'bad' and 13215 'good' cases.

Five models are tested on the test sample 2. Their performances are compared with respect to the classification accuracy and other applicability criterions including speed and transparency.

5.3.2.1 Classification accuracy

Figure 5.3.2.1/1 shows the ROC curves of the five classifiers.

Since k-NN do not produce continuous scores, a score is computed for each case:

Score for i_th case = \( b_i / k \),

where k is the number of nearest neighbors, \( b_i \) is the number of 'bad' cases in the k nearest neighbors of i_th case. The score can be explained as the local probability of the case belonging to 'bad' class. Since the scores calculated in this way are not continuous, there are ties between discrete points in the ROC curve for the k-NN model.
An example of the credit scoring model evaluation

The values of AUC (area under the curve) are given in Table 5.3.2.1/1.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5</td>
<td>0.974</td>
</tr>
<tr>
<td>MLP</td>
<td>0.949</td>
</tr>
<tr>
<td>k-NN</td>
<td>0.941</td>
</tr>
<tr>
<td>LR</td>
<td>0.937</td>
</tr>
<tr>
<td>LDA</td>
<td>0.931</td>
</tr>
</tbody>
</table>

Table 5.3.2.1/1: AUC for five algorithms

According to the discussions of deciding the threshold value in Chapter 4.2.3, if the slope of the cost functions $S$ or a possible range of $S$ can be estimated, the optimal threshold value that minimizes the costs of error classification can be determined. For our problem at hand, we just divided the cases according to their ratings. The actual payment behaviors are not known, the costs of different types of errors can not be given in this situation.

To compare the five algorithms a value of $S$ is supposed. For example, suppose $S = 1/4$, the threshold values for each algorithms are shown in the ROC curves. The corresponding Type I error rates ($E_1$), Type II error rates ($E_2$) are shown in Table 5.3.2.1/2.

---

15 The values of AUC are calculated by SPSS.
An example of the credit scoring model evaluation

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>E1</th>
<th>E2</th>
<th>Total error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5</td>
<td>5.0%</td>
<td>5.6%</td>
<td>5.3%</td>
</tr>
<tr>
<td>MLP</td>
<td>9.8%</td>
<td>14.4%</td>
<td>12.1%</td>
</tr>
<tr>
<td>kNN</td>
<td>10.8%</td>
<td>14.2%</td>
<td>12.5%</td>
</tr>
<tr>
<td>LR</td>
<td>10.0%</td>
<td>24.7%</td>
<td>17.4%</td>
</tr>
<tr>
<td>LDA</td>
<td>10.2%</td>
<td>25.8%</td>
<td>18.0%</td>
</tr>
</tbody>
</table>

Table 5.3.2.1/2: Optimal models for $S = 1/4$

The best model is the model tree M5. MLP and k-NN are the second best models. LDA and LR are the worst ones. The significance of differences in error rates between each pair of algorithms is not statistically tested. But it can be concluded that the differences among M5, MLP (kNN), and LR (LDA) are significant, while the differences between MLP and kNN, between LR and LDA are small.

5.3.2.2 Practicability criterions

The performance of models can be evaluated with other two important practicability criterions: the speed of models in terms of training time and applying time as well as the transparency of the final model (see Chapter 3.2).

The comparison result of the models' speed is shown in Table 5.3.2.2/1.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
</tr>
<tr>
<td>LDA</td>
<td>○</td>
</tr>
<tr>
<td>LR</td>
<td>○</td>
</tr>
<tr>
<td>M5</td>
<td>○</td>
</tr>
<tr>
<td>MLP</td>
<td>●</td>
</tr>
<tr>
<td>k-NN</td>
<td>●</td>
</tr>
</tbody>
</table>

○ in seconds ● More than one minute ■ More than 10 minutes

Table 5.3.2.2/1: The comparison of the speed of five models

16 Although significance testing is not carried out, according to the discussion in chapter 3.1.3, the significance of difference between classification accuracy can be roughly observed. According to the graphic method of significant test (see Figure 3.1.3/1), with the 7640 cases test sample, if the sum of two error rates is less than 20%, then the difference more than 1.04% is definitely significant. However, the difference less than 1.04% may be or may not be significant according to how many cases are classified actually differently by two classifiers.
LDA and LR is the most rapid model in the terms of training time, it needs almost no time to build the linear models (in several seconds). The training for M5 model trees is a little slower but still can be finished in several minutes. The most time-consuming model is neural network method, training the final MLP network takes more than 30 minutes. It worth remarking that although k-NN method needs no time at all for training, but to find the proper parameter k, models should be tested on large test sample. This testing procedure is rather time-consuming, takes several minutes to several ten minutes, depending on the size of the train and the test sample as well as the value of parameter k.

In the sense of applying models to new cases, all algorithms are at the same level of speed except k-NN, which is slower due to the fact that classification procedure occurs at the applying time. If 26430 cases are used as the train sample, it will take more than one minute to classify only one case.

The transparency of a classification model means whether the results of model can be explained and whether the explanation is logical:

- The transparency of MLP and k-NN

  The models of MLP neural network and k-nearest-neighbors methods are opaque. The relation between input variables and the final classification result cannot be seen. The influence of each variable on the final scores can not be observed directly.

  To solve the difficulty in explaining the variables' influence on output, sensitivity analysis is usually used. The sensitivities of each variable can be calculated by varying the value of one input variable while fixing all other input variables and measuring the magnitude of the impact on the output. Thus an index can be determined for each input variable that measures its relative effect on the final scores (cf. Schu91, P. 34; West00, P. 1148). One can also use a sensitivity curve of a variable to show its influence on the final score. The sensitivity curve of an input variable is defined as the curve of the scores if the variable runs through its value-field on the test data set and all other input variables remain unchanged on their average values (cf. FrHo98, P. 190).

  A problem of sensitivity analysis arises when input variables are categorical. Since categorical variables have not average values, sensitivity curves should be drawn for each value of a categorical variable. If there are more than one categorical input variables, sensitivity curves should be drawn for each possible combination of the categorical variables.

- The transparency of LDA and LR

  Both of these two algorithms produce scores with a linear function of input variables. The sign of a variable's coefficient in the function represents the influence of this variable on
the score: either increasing or decreasing the final score. The signs of some variables' coefficients in LDA and LR models are listed in Table 5.3.2.2/2.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Sign</th>
<th>Coefficients for LDA</th>
<th>Coefficients for LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAHRNUM</td>
<td>-</td>
<td>-0.1685</td>
<td>-0.1798</td>
</tr>
<tr>
<td>STAMMKAP</td>
<td>+</td>
<td>6.2429e-10</td>
<td>2.594e-9</td>
</tr>
<tr>
<td>UM12</td>
<td>-</td>
<td>-0.0005</td>
<td>-2.7516e-3</td>
</tr>
<tr>
<td>BELAST</td>
<td>+</td>
<td>0.0003</td>
<td>3.106e-3</td>
</tr>
<tr>
<td>ZAHLWEISE</td>
<td>+</td>
<td>0.9261</td>
<td>4.1714</td>
</tr>
<tr>
<td>AUFTRAGE</td>
<td>+</td>
<td>0.0197</td>
<td>0.1095</td>
</tr>
<tr>
<td>GRUNDAHR</td>
<td>-</td>
<td>-0.0030</td>
<td>-8.6163e-3</td>
</tr>
<tr>
<td>LUMSATZ/MITARBEITER</td>
<td>-</td>
<td>-1.3590e-9</td>
<td>-1.3514e-9</td>
</tr>
</tbody>
</table>

Table 5.3.2.2/2: Variables' coefficients for LDA and LR functions

A higher score corresponds to the higher probability of the case belongs to 'bad' class. Therefore, according to the sign of these coefficients, the influence of input variables can be explained. For example, ZAHLWEISE is the index of mode of payment (see Appendix A.1). The sign of coefficient of 'ZAHLWEISE' is logical, which means the larger is 'ZAHLWEISE', the higher is the probability of the case belongs to 'bad' class. Similarly, the signs of coefficients of variable 'UM12', 'AUFTRAGE' (see Appendix A.2) and LUMSATZ/MITARBEITER are also logical.

The LDA and LR methods are sensitive to highly correlated input variables. If some input variables are highly correlated with each other, the coefficients of variables would be biased and not explained logically (cf. Feid92, P.113; Nieh87, P. 109). For example, if 'BONITAETINDEX' and 'ZAHLWEISE' both are used as input variables, one of their coefficients is positive, the other is negative. This is not logical since both of them should have positive coefficients. Due to this reason, the input variables 'BONITAETINDEX' (see Appendix A.4), 'ENTWICKLUNG' (see Appendix A.3), 'UM23' and 'IS_BELAST' are removed from the train sample for LDA and LR models. Although this results in a little decrease in the prediction accuracy (decrease of 0.4% for both LDA and LR), the derived models are more explainable.

The explanation of influences of other input variables is difficult. LDA and LR assume linear relationships between final scores (the insolvent probability) and an input variable. However, this assumption is not realistic for some input variables.
For example, GRUNDJAHR is the age of the company. The insolvent probability is usually not the linear function of GRUNDJAHR. The relation between GRUNDJAHR and the insolvent probability may depend on different values of other variables. For example, when 'ZAHLWEISE >2' and when 'ZAHLWEISE<=2', the influence of GRUNDJAHR on the insolvent probability may be different. In fact, it is one of the shortcomings of LDA and LR models that they cannot cope with interactions among input variables. An interaction occurs when the relationship between the output variable and an input variable changes for differing values of another input variable. One possible solution is to add new combined variables, for example, the product of two variables. Such a way might increase the prediction power of models. However, the transparency of models decreases since these combined variables are not interpretable.

Other solution is to divide the continuous values of the variable into several intervals through discretization. It is expected that there is a linear relation with the insolvent probability in each intervals of the variable's value. However, the reasonable discretization is not easy to find.

- The transparency of the model tree

The transparency of model tree is very high. The final model tree derived by M5 algorithm is given in Appendix C. The model tree consists of two parts: a tree structure and the linear models as leaves. The model tree method combines the advantages of decision tree methods and regression methods. The model trees keep the attractive character of decision trees that a set of understandable rules can be drawn from the trees. The tree structure provides a solution to deal with the interaction among input variables. With the regression functions at each leaves model trees can produce continuous scores, thus the shortcoming of decision tree methods, which give only binary classifications, is overcome.

5.4 The comparison of the scoring models with the rating system

Since the cases have ratings in four levels A, B, C, D, which are given by the current rating system. The scoring model generated by the M5 model tree algorithm is taken as an example to show how the cases with different ratings are classified by the scoring models.

We apply the finally derived model tree to all the available cases in the data set. Figure 5.4/1, Figure 5.4/2, Figure 5.4/3 are the distribution of the scores of A/B, C and D companies.
Figure 5.4/1: The distribution of the scores of A/B companies

Figure 5.4/2: The distribution of the scores of C companies

Figure 5.4/3: The distribution of the scores of D companies
An example of the credit scoring model evaluation

It is clearly shown in the figures that the most of A/B companies have scores lower than 0.3 and the D companies have scores higher than 0.9, while the scores of C companies are scattered in the range of the scores.

The attractive character of the scoring model is the continuous scores assigned to each case. The risk levels of each case are measured in a continuous scale. In practical application, several-classes can be constructed through dividing the scores into several segments. Credit decisions are made flexibly by changing the division values of scores. A flexible way of using the scores produced by the model is as follows:

With two threshold values $T_1$ and $T_2$, the cases can be divided into three classes: class 1 (score<$T_1$), class 2 ($T_1$<=$score$<$T_2$), and class 3 (score=$T_2$). The following cross tables (See Table 5.4/1) show the distribution of cases in the ratings and score classes with different $T_1$ and $T_2$.

<table>
<thead>
<tr>
<th>Ratings</th>
<th>Score classes (T1=0.3 T2=0.97)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>Count</td>
<td>2</td>
</tr>
<tr>
<td>%</td>
<td>66,7%</td>
<td>33,3%</td>
</tr>
<tr>
<td>B</td>
<td>Count</td>
<td>46905</td>
</tr>
<tr>
<td>%</td>
<td>97,1%</td>
<td>2,4%</td>
</tr>
<tr>
<td>C</td>
<td>Count</td>
<td>1248</td>
</tr>
<tr>
<td>%</td>
<td>11,0%</td>
<td>19,2%</td>
</tr>
<tr>
<td>D</td>
<td>Count</td>
<td>1</td>
</tr>
<tr>
<td>%</td>
<td>0,0%</td>
<td>100,0%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>48156</td>
</tr>
<tr>
<td>%</td>
<td>73,7%</td>
<td>5,1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ratings</th>
<th>Score classes (T1=0.3 T2=0.985)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>Count</td>
<td>2</td>
</tr>
<tr>
<td>%</td>
<td>66,7%</td>
<td>33,3%</td>
</tr>
<tr>
<td>B</td>
<td>Count</td>
<td>46905</td>
</tr>
<tr>
<td>%</td>
<td>97,1%</td>
<td>2,8%</td>
</tr>
<tr>
<td>C</td>
<td>Count</td>
<td>1248</td>
</tr>
<tr>
<td>%</td>
<td>11,0%</td>
<td>33,8%</td>
</tr>
<tr>
<td>D</td>
<td>Count</td>
<td>1</td>
</tr>
<tr>
<td>%</td>
<td>0,0%</td>
<td>100,0%</td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>48156</td>
</tr>
<tr>
<td>%</td>
<td>73,7%</td>
<td>7,9%</td>
</tr>
</tbody>
</table>
It is shown that most of A/B companies are classified into the first class, and most of D companies are classified into the third class. That means the most of A/B companies and D companies can be identified by the scoring model.

The C companies, whose risk levels are difficult to be decided, are classified into three classes with different percentages. The C companies in class 3 have the similar patterns as D companies, while the C companies in class 1 have the similar patterns as A/B companies. The C companies that classified into class 2 are ambiguous. The credit decision with them should be made individually by the experts.

The threshold values T1, T2 can be decided flexibly by the users. If a larger T2 is used then more C companies are classified into class 2 and need to be individually reviewed by experts.

<table>
<thead>
<tr>
<th>Rating</th>
<th>A</th>
<th>Count</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>%</td>
<td>66.7%</td>
<td>33.3%</td>
<td>100.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>%</td>
<td>97.1%</td>
<td>2.9%</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>%</td>
<td>11.0%</td>
<td>71.7%</td>
<td>17.3%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>Count</td>
<td>48156</td>
<td>9568</td>
<td>7643</td>
<td>65367</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4/1: Crosstabulation of ratings and score classes
6 Summary and conclusion

This paper focused on the problem of the evaluation of classification models for credit scoring. The reason of the model evaluation is to find the optimal solution from various classification models generated in an iterated and complex model building process.

Classification accuracy is the basic criterion to measure the performance of a model. Classification error rates can be estimated from the test sample. In order to get an accurate estimation, the sample of test data has some requirements: independent from training data, representative for the underlying population, and large enough.

Because credit scoring model should be effectively applied to real world problems, other criterions for evaluating practical scoring models are also important under some circumstances: the speed, the transparency and the simplicity of the models.

Usually credit scoring models classify credit customers into two classes: 'good' and 'bad'. The two types of error rates are used to assess the performance of models. There are different evaluation methods to describe these two types of errors. The most simple and direct method is confusion matrix. ROC curve is the graphic representation of the tradeoff between two types of errors. If the cost function, which weighs the two kinds of errors, is available, the optimal threshold value on the ROC curve can be determined and the cost minimization model is obtained. Some practical ways of selecting threshold values are discussed.

With the help of a learning curve, incremental analysis can be carried out in order to get the proper model structures and train sample size. There are some basic phenomena in the incremental cases analysis and incremental complexity analysis.

An example illustrated the process of using five algorithms to build credit scoring models, then evaluating and comparing them. Decision tree method M5 has a higher classification accuracy than other methods. The second accurate model is produced by the multi-layer perceptron neural network method, but its outputs can not be understood intuitively.

Finally, the scores obtained from models are compared with the ratings given by the current system. A flexible way of using the scores produced by scoring models is explained.
Appendices

Appendix A. Explanations of some variables in the data set

The variables listed here can help to understand the explanations of the models in this paper. The information comes from the German credit information agency 'Die Vereine Creditreform'. To keep their original meanings the explanations listed here are in German.

A.1 Variable ‘ZAHLWEISE’

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Zahlungserfahrungen liegen noch nicht vor</td>
</tr>
<tr>
<td>02</td>
<td>bisher keine ausreichenden Erfahrungen, Beanstandungen liegen nicht vor</td>
</tr>
<tr>
<td>11</td>
<td>Skontoausnutzung</td>
</tr>
<tr>
<td>12</td>
<td>pünktlich, meist Skonto</td>
</tr>
<tr>
<td>13</td>
<td>zumeist Skonto, teils Zielinanspruchnahme bis zu 30 Tagen</td>
</tr>
<tr>
<td>21</td>
<td>innerhalb vereinbarter Ziele</td>
</tr>
<tr>
<td>22</td>
<td>innerhalb vereinbarter Ziele von 30 bis 60 Tagen</td>
</tr>
<tr>
<td>23</td>
<td>innerhalb vereinbarter Ziele von 60 bis 90 Tagen</td>
</tr>
<tr>
<td>29</td>
<td>soweit bekannt, pünktlich</td>
</tr>
<tr>
<td>31</td>
<td>ohne Beanstandungen bei gelegentlichen Zielüberschreitungen</td>
</tr>
<tr>
<td>32</td>
<td>meist innerhalb vereinbarter Ziele, teils auch länger</td>
</tr>
<tr>
<td>33</td>
<td>bisher ohne Beanstandungen, in letzter Zeit gelegentlich Zahlungserinnerungen</td>
</tr>
<tr>
<td>39</td>
<td>ohne Beanstandungen</td>
</tr>
<tr>
<td>41</td>
<td>Zielüberschreitung bis zu 30 Tagen</td>
</tr>
<tr>
<td>42</td>
<td>teils innerhalb vereinbarter Ziele, teils mit Zielüberschreitung</td>
</tr>
<tr>
<td>43</td>
<td>langsam mit Zielüberschreitungen, mehrfach Zahlungserinnerungen</td>
</tr>
<tr>
<td>44</td>
<td>langsam mit Zielüberschreitungen, Wechselprolongationen</td>
</tr>
<tr>
<td>51</td>
<td>erhebliche Zielüberschreitungen bis zu 3 Monaten</td>
</tr>
<tr>
<td>52</td>
<td>erhebliche Zielüberschreitungen von mehr als 3 Monaten</td>
</tr>
<tr>
<td>53</td>
<td>schleppend, mehrfach Mahnung, auch Scheckrückgaben</td>
</tr>
<tr>
<td>54</td>
<td>wird beanstandet, termingebundene Verpflichtungen wurden nicht eingelöst</td>
</tr>
<tr>
<td>55</td>
<td>langsam und schleppend, Creditreform-Inkasso-Dienst wurde eingeschaltet</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>61</td>
<td>Haftanordnung zur Abgabe der eidesstattlichen Versicherung</td>
</tr>
<tr>
<td>62</td>
<td>Abgabe der eidesstattlichen Versicherung über die Vermögensverhältnisse</td>
</tr>
<tr>
<td>63</td>
<td>Antrag auf Eröffnung des gerichtlichen Vergleichsverfahrens</td>
</tr>
<tr>
<td>64</td>
<td>Eröffnung des gerichtlichen Vergleichsverfahrens</td>
</tr>
<tr>
<td>65</td>
<td>Antrag auf Eröffnung des Konkursverfahrens</td>
</tr>
<tr>
<td>66</td>
<td>Eröffnung des Konkursverfahrens</td>
</tr>
<tr>
<td>67</td>
<td>Konkursverfahren mangels Masse abgelehnt</td>
</tr>
<tr>
<td>68</td>
<td>Eröffnung des Vorverfahrens</td>
</tr>
</tbody>
</table>

### A.2 Variable 'AUFRAGE'

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 sehr gute Auftragslage</td>
</tr>
<tr>
<td></td>
<td>11 sehr guter Geschäftsgang</td>
</tr>
<tr>
<td>2</td>
<td>20 gute Auftragslage</td>
</tr>
<tr>
<td></td>
<td>21 guter Geschäftsgang</td>
</tr>
<tr>
<td>3</td>
<td>30 zufriedenstellende Auftragslage</td>
</tr>
<tr>
<td></td>
<td>31 zufriedenstellender Geschäftsgang</td>
</tr>
<tr>
<td>4</td>
<td>40 rückläufige Auftragslage</td>
</tr>
<tr>
<td></td>
<td>41 rückläufiger Geschäftsgang</td>
</tr>
<tr>
<td>5</td>
<td>50 mäßige Auftragslage</td>
</tr>
<tr>
<td></td>
<td>51 mäßiger Geschäftsgang</td>
</tr>
<tr>
<td>6</td>
<td>60 schlechte Auftragslage</td>
</tr>
<tr>
<td></td>
<td>61 schlechter Geschäftsgang</td>
</tr>
</tbody>
</table>

### A.3 Variable 'ENTWICKLUNG'

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01 Geschäftsentscheidung bleibt abzuwarten</td>
</tr>
<tr>
<td>1</td>
<td>10 expansive Unternehmensentwicklung</td>
</tr>
<tr>
<td></td>
<td>11 expansive Geschäftsentscheidung</td>
</tr>
<tr>
<td>2</td>
<td>20 positive Unternehmensentwicklung</td>
</tr>
<tr>
<td></td>
<td>21 positive Geschäftsentscheidung</td>
</tr>
<tr>
<td>3</td>
<td>30 konstante Unternehmensentwicklung</td>
</tr>
<tr>
<td></td>
<td>31 konstante Geschäftsentscheidung</td>
</tr>
</tbody>
</table>
### Appendices

| 4 | 40 stagnierende Unternehmensentwicklung |
| 41 stagnierende Geschäftsentwicklung |
| 5 | 50 rückläufige Unternehmensentwicklung |
| 51 rückläufige Geschäftsentwicklung |
| 6 | 60 stark rückläufige Unternehmensentwicklung |
| 61 stark rückläufige Geschäftsentwicklung |

#### A.4 Variable ‘BONITAETINDEX’

‘BONITAETINDEX’ is the Creditreform credit-standing index. It is given in the form of a three-digit figure (from 100 to 600). The first digit corresponds to a ranking aligned to the German system of school grades (where 1 is very good and 6 is fail); the second and third digits permit further differentiation (more information see CreditR02).

#### Appendix B. Brief summary of software and toolsets used in this study

Weka (Decision tree M5, Neural networks and k-NN): Weka is a software workbench which integrates many different Machine learning methods as well as some useful preprocessing methods (more information see WiFr00).

SPSS (Logistic regression and Linear discriminant analysis): SPSS is an integrated statistic tool. It provides statistical classification algorithms as well as tools for descriptive analysis, sampling, and graphics.

#### Appendix C. The derived model tree

**Tree structure:**

```
ZAHLWEISE <= 29.5:
  | BONITAETINDEX <= 224:
  | ZAHLWEISE <= 12.5:
  | | GRUNDJAHR <= 11.5:
  | | | GRUNDJAHR <= 4.5:
  | | | | GRUNDJAHR <= 2.5: LM1
  | | | | GRUNDJAHR > 2.5: LM2
  | | | GRUNDJAHR > 4.5: LM3
  | | GRUNDJAHR > 11.5: LM4
  | ZAHLWEISE > 12.5:
  | | UM23 <= -2.5:
  | | | UM12 <= -8.5: LM5
  | | | UM12 > -8.5: LM6
  | | | UM23 > -2.5:
  | | | | GRUNDJAHR <= 8.5:
  | | | | | GRUNDJAHR <= 4.5:
  | | | | | | GRUNDJAHR <= 2.5: LM7
  | | | | | | GRUNDJAHR > 2.5: LM8
  | | | | | GRUNDJAHR > 4.5: LM9
```
| | | | GRUNDJAHR > 8.5 : LM10 |
| | | | BONITAETINDEX > 224 : |
| | | | UM12 <= -2.5 : |
| | | | UM12 <= -9.5 : |
| | | | UM23 <= -9.5 : LM11 |
| | | | UM23 > -9.5 : LM12 |
| | | | UM12 > -9.5 : LM13 |
| | | | UM12 > -2.5 : |
| | | | GRUNDJAHR <= 4.5 : |
| | | | GRUNDJAHR <= 2.5 : LM14 |
| | | | GRUNDJAHR > 2.5 : LM15 |
| | | | GRUNDJAHR > 4.5 : |
| | | | | BONITAETINDEX <= 256 : LM16 |
| | | | | BONITAETINDEX > 256 : |
| | | | | GRUNDJAHR <= 13.5 : LM17 |
| | | | | GRUNDJAHR > 13.5 : |
| | | | | | ENTWICKLUNG <= 3.5 : LM18 |
| | | | | | ENTWICKLUNG > 3.5 : LM19 |
| | | | | BONITAETINDEX <= 278 : |
| | | | | BONITAETINDEX > 278 : |
| | | | | | ENTWICKLUNG <= 3.5 : LM20 |
| | | | | | ENTWICKLUNG > 3.5 : LM21 |
| | | | ZAHLWEISE > 29.5 : |
| | | | ZAHLWEISE <= 41.5 : LM22 |
| | | | ZAHLWEISE > 41.5 : LM23 |

Linear models at the leaves:

LM1: Score = 0.873 - 0.00158V1 - 0.0011V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.0848V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM2: Score = 0.458 - 0.00158V1 - 0.0011V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.102V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM3: Score = 0.0349 - 0.00158V1 - 0.0011V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.00367V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM4: Score = 9.43e-4 - 5.5e-4V1 - 0.0011V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 1.66e-5V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM5: Score = 0.712 - 1.31e-4V1 - 5.83e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.00367V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM6: Score = 0.0349 - 0.00158V1 - 0.0011V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.00367V12 + 0.00128V13 - 0.00274V14 + 0.00166V15 + 7.59e-5V16.

LM7: Score = 0.927 - 1.31e-4V1 - 5.83e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.051V12 + 0.00118V13 - 0.00237V14 + 0.00133V15 + 7.59e-5V16.

LM8: Score = 0.216 - 1.31e-4V1 - 5.83e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.0341V12 + 0.00118V13 - 0.00237V14 + 0.00133V15 + 7.59e-5V16.

LM9: Score = 0.107 - 1.31e-4V1 - 5.83e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.00237V12 + 0.00118V13 - 0.00237V14 + 0.00133V15 + 7.59e-5V16.

LM10: Score = 0.0788 - 1.31e-4V1 - 5.83e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 6.49e-6V6 + 1.51e-4V8 + 2.75e-5V9 + 7.46e-5V10 + 2.32e-6V11 - 0.109V12 + 0.00118V13 - 0.00237V14 + 0.00133V15 + 7.59e-5V16.

LM11: Score = 0.903 - 1.89e-4V1 - 5.03e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 3.24e-6V6 + 8.65e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.00161V10 + 9.63e-5V11 - 1.65e-6V12 + 4e-4V13 - 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM12: Score = 0.632 - 1.89e-4V1 - 5.03e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 3.24e-6V6 + 8.65e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.00161V10 + 9.63e-5V11 - 1.65e-6V12 + 4e-4V13 - 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM13: Score = 0.194 - 1.89e-4V1 - 5.03e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 7.49e-6V6 + 8.65e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.00161V10 + 9.63e-5V11 - 1.65e-6V12 + 4e-4V13 - 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM14: Score = 0.88 - 0.00201V1 - 5.03e-4V2 - 1.1e-4V3 + 1.11e-4V4 - 4.34e-6V6 + 7.49e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.00213V10 + 1.03e-4V11 - 0.0258V12 + 4e-4V13 - 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.
LM15: Score = 0.132 - 0.00201V1 - 5.03e-4V2 - 1.11e-4V4 - 4.34e-6V6 + 7.49e-6V7 + 7.81e-5V8 - 2.75e-5V9 + 0.00213V10 + 1.03e-4V11 - 0.0144V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM16: Score = -0.0756 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 - 4.34e-6V6 + 7.81e-5V8 + 2.75e-5V9 + 0.0492V10 + 8.23e-5V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM17: Score = 0.0597 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 + 3.71e-9V5 - 4.34e-6V6 + 3.66e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.00532V10 + 1.16e-6V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM18: Score = -0.0424 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 + 1.16e-8V5 - 4.34e-6V6 + 3.66e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.0364V10 + 1.16e-6V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM19: Score = 0.218 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 + 1.16e-8V5 - 4.34e-6V6 + 3.66e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.0756V10 + 1.16e-6V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM20: Score = 0.0747 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 - 4.34e-6V6 + 3.66e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.0273V10 + 1.16e-4V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM21: Score = 0.46 - 6.85e-4V1 - 5.03e-4V2 - 1.11e-4V4 - 4.34e-6V6 + 3.66e-6V7 + 7.81e-5V8 + 2.75e-5V9 + 0.0517V10 + 1.16e-4V11 - 1.65e-6V12 + 4e-4V13 + 7.86e-4V14 + 5.3e-4V15 + 7.59e-5V16.

LM22: Score = 0.984 - 8.01e-5V1 - 1.59e-4V3 + 1.59e-4V4 - 1.41e-4V7 + 1.16e-6V8 + 3.95e-5V9 - 5.81e-6V11 - 2.3e-6V12 + 1.09e-4V16.

LM23: Score = 0.997 - 8.01e-5V1 - 1.59e-4V3 + 1.59e-4V4 - 4.2e-7V7 + 1.15e-4V8 + 3.95e-5V9 - 5.81e-6V11 - 2.3e-6V12 + 1.09e-4V16.

V1: JAHRNUM
V2: RECHTSFORM = 77,46,78,24,76,79,0
V3: RECHTSFORM = 46,78,24,76,79,0
V4: RECHTSFORM = 24,76,79,0
V5: STAMMKAP
V6: UM12
V7: UM23
V8: ZAHLWEISE
V9: AUFTRAGE
V10: ENTWICKLUNG
V11: BONITAETINDEX
V12: GRUNDJAHR
V14: BRANCH = A,I,H,F,M,Q
V15: BRANCH = I,H,F,M,Q
V16: BRANCH = H,F,M,Q
Literature


AdHa99 Adams, N. M./Hand, D. J.: Comparing classifiers when the misallocation costs are uncertain. In: Pattern Recognition, 32(1999), P. 1139-1147.


Hofm90 Hofmann, V. H. J.: Die Anwendung des CART-Verfahrens zur statistischen
Bonitätsanalyse von Konsumentenkrediten. In: Zeitschrift für Betriebswirtschaft,
60(1990), 9, 941-962.

Joos98 Joos, P./Vanhoof, K./Ooghe, H./Sierens, N.: Credit classification: a comparison of logit
models and decision trees. In: Gholamreza Nakhaeizadeh et al. (eds.), Application of
machine learning and data mining in finance: European Conference on Machine

Krah98 Krahl, D./Windheuser, U./Zick, F.K.: Data Mining: Einsatz in der Praxis. Addison-

Krau93 Krause, C.: Kreditwürdigkeitsprüfung mit Neuronalen Netzen. IDW-Verlag GmbH,
Düsseldorf, 1993.

Krus99 Kruse, A.: Antragsprüfung und Kartenüberwachung von privaten Kreditkartenkunden mit

LiMo98 Liu, H./Motoda, H.: Feature selection for knowledge discovery and Data Mining. Kluwer

Liuy02 Liu, Y.: An framework of data mining application process for credit scoring. Research
paper, Institute of Information Systems, University of Goettingen, Nr. 02/2002,
Göttingen.

Lohr94 Lohrbach, T.: Einsatz von Künstlichen Neuronalen Netzen für ausgewählte
betriebswirtschaftliche Aufgabenstellungen und Vergleich mit konventionellen

MiAr95 Milton, J. S./Arnold, J. C.: Introduction to probability and statistics: Principles and
1995.

Mich94 Michie, D./Spiegelhalter, D. J./Taylor, C. C.: Machine Learning, Neural and Statistical

Müll96 Müller, J.: DV-gestützte Systeme zur Kreditwürdigkeitsprüfung bei

NaTu97 Nayak, G. N. / Turvey, C. G.: Credit risk assessment and the opportunity costs of loan

Nieh87 Hiehaus, H.: Früherkennung von Unternehmenskrisen. IDW-Verlag GmbH, Düsseldorf,
1987.


In: Proceedings of the sixth international workshop on machine learning. Morgan

Schu91 Schumann, M.: Neuronale Netze zur Entscheidungsunterstützung in der
Betriebswirtschaft, in: Biethahn, J./Bloech, J./Bogaschewski, R./Hoppe, U. (Hrsg.),


