Intelligent Target Selection for Direct Marketing

Sjoerd van Geloven

25 April 2002

Preface

This report is the description of the thesis project "Intelligent Target Selection in Direct Marketing", the final project of the Electrical Engineering curriculum given at the Faculty Information Technology and Systems (ITS) at Delft University of Technology (DUT).

This thesis originated of a mutual interest of the Control Laboratory of the ITS faculty and the Department of Computer Science of the Economics Faculty of the Erasmus University in Rotterdam in the use of Artificial Intelligence (AI) in Direct Marketing.

I would like to thank Robert van Amerongen and Robert Babuska, from the Control group, for giving me the opportunity to work on this interesting project that is not a typical graduation project for Electrical Engineering students. Furthermore, I am grateful to both for their critical comments and the latter for his support during the implementation phase.
From the Erasmus University, I would like to thank Willem-Max van der Bergh, Jan van den Berg and Uzay Kaymak for their guidance and support through this project. Especially the latter, who provided the project description and provided me with both a lot of insight information on the subject of target selection and the fuzzy algorithm implementation.

Sjoerd van Geloven
25 April 2002

List of Tables

List of Figures

Contents

Chapter 1

Introduction

Direct marketing is the process of approaching customers with the intention to sell products or services. Because the typical response is low, it is worthwhile to know which customers are interested in the offer in mind. In this way, resources can be directed to the profitable customers. It is, however, very difficult to predict in advance, which persons are interested in a specific product or service. This problem is known as "target selection" in direct marketing, in which the targets are the prospects who will respond.
Most companies have a lot of information about their customers. The purchase history and zip code information are usually present in a client database. Zip code information contains the typical characteristics of an individual living in a certain zip code district. A characteristic, e.g. age, of clients stored in such a database is called a variable, attribute or feature. These features can be used to describe customers which hopefully results in a certain likelihood of response.

A general approach to tackle the difficulties in predicting the response of prospects is Knowledge Discovery in Data. For the part in this process dealing with the mining of clients resulting in a certain likelihood of response, several algorithms have been used over the years. The relation between the attributes and the variable describing the response is expected to be non-linear, because of the fact that humans tend to base their decisions on both knowledge and emotion. We do not have the availability of these factors, but we do have knowledge of some characteristics of the prospects. We hope that the features used in a certain configuration are good predictors but also know that human decision making does not always depend on rational thinking. So, the assumption is that the relationship between the features and the target variable has a non linear character. The degree of this non-linearity is not known. This is the reason why many target selection algorithms with various complexity structures are used during the last decades. Five of them, increasing in complexity, are subject of study in this thesis.

Another goal of this project is to investigate whether the algorithms (three statistical ones and two algorithms based on artificial intelligence) used for the purpose of target selection in direct marketing could help each other in order to capture the assumed high complexity and ultimately obtain better results. More specifically, can the strength of an algorithm overcome a weakness of another?

For example, can the feature subset denoted as significant by one algorithm prove better results for a second? The intention is to build a system which is more intelligent than a stand-alone algorithm in order to capture assumed high non-linearity but still can act as autonomously as possible. This last demand is added because the intervention of domain experts is time consuming and expensive. The underlying thought for this operation is that every gain in targeting the most valuable customers directly results in extra revenue, and eventually increases profits.
Before the questions above can be answered, the algorithms suited for target selection have to be studied and compared. Another interesting question is whether different types of variables (binary, categorical, continuous or other) and scaling (transforming the feature values) have impact on target selection. These effects are studied for each algorithm or configuration.

So, the main goal of this thesis is to investigate which configuration of one or more algorithms performs best for the purpose of target selection in direct marketing. Furthermore, the scaling and different type of variable treatments of this configuration are explored.

The outline of this thesis is as follows: in chapter 2 the application domain of direct marketing is further explored and the approach of target selection by means of Knowledge Discovery in Data explained. Chapter 3 describes the five algorithms used for target selection in this thesis by their mathematical background. Not all algorithms can be applied in a straight-forward way: some algorithm specific variables need to be set. These parameter choices and additional implementation issues are also covered in chapter 3. In chapter 4 the search for complexity fit starts. After introducing some performance measures used for comparing the different configurations, all algorithms are optimized for the purpose of target selection. The methodology is explained by a real-life direct marketing target selection problem. This data set is introduced and suggestions for optimal target selection are given for this example. The chapter ends with fertile combinations of algorithms: the configurations which can cope with the highest complexity. Results of the stand-alone algorithms and the combinations on this real-life data set are presented in chapter 5, together with a comparison to previous obtained results on this data set. Finally, in chapter 6 conclusions are drawn and recommendations are made for further research.

Chapter 2

Target Selection in Direct Marketing

"A general rule of thumb in marketing is that approximately 20 percent of a company's customers account for about 80 percent of its business." ... "It is not worthwhile for a company to offer generous promotions to the other 80 percent. The more these loyal customers are retained, the more data can be gathered about them." [1]
From this fragment it may be concluded that it is very important to know which customers would be interested in certain offers.
This chapter tries to give insight in the way target selection in direct marketing can be applied and first of all, what direct marketing is. The application domain, direct marketing, is explored in section 2.1. Section 2.2 provides an overview of the Knowledge Discovery in Data cycle, a general method very well suitable for target selection. Finally, in section 2.3 gain charts are explained; the graphs used in this thesis to compare results.

2.1 Direct Marketing

Direct marketing is a powerful tool for companies to increase profit, especially when one knows his customers well. A definition of direct marketing is: "Direct marketing is the science of arresting the human intelligence long enough to take money off it" [3]. This sounds cynical, but true. However, a formal definition needs more than a laugh. The Direct Marketing Association, the industry's official trade association, defines direct marketing as "an interactive system of marketing which uses one or more advertising media to effect a measurable response or transaction at any location." In a general marketing activity, there are three matters to be analyzed: the *customers*, the *competitors* and the *company* marketing the product. In this thesis, the focus is on the customers. How can the most valuable customers or prospects be targeted?

The advantage of direct marketing compared to other marketing activities is that it seeks a direct response from pre-identified prospects. It allows companies to direct their energies toward those individuals who are most likely to respond. This selectivity is necessary to enable the use of high cost – high

impact marketing programs such as personal selling in the business-to-business environment [4]. A second fundamental characteristic of direct marketing is that it generates individual response by known customers. This enables companies to construct detailed customer purchase histories, which tend to be far more useful than traditional predictors based on geodemographics and psychographics [4].

The growth of direct marketing can be explained by several changes in the social environment [2] [5]. First of all, the composition of households changed. Nowadays a large number of women joins the labor force and there are more double-income households then ever. Secondly a re-evaluation of leisure time is taking place. Next to this, credit cards offer convenience and the required security for payments. Another enabler is the information and communication technology: high computing power allows computations that never could be done by humans, larger amounts of data can be stored is and new advertising media like the Internet, e-mail and WAP (Wireless Application Protocol, used in Telecommunications for data transport) are introduced. Last but not least new life styles and consumer values are incorporated in society. Direct marketing provides opportunities for identifying, measuring and reaching these groups as market segments.

Generally, there are two ways to identify those customers who are most likely to respond to offerings. Either you have a customer database yourself, or you acquire a list from, so-called, list brokers. Apart from the list, there are some other issues to take into account. In direct marketing, five questions are often considered [3]:

1. Who am I trying to reach?
2. What do I want them to do?
3. Why should they do it?
4. Where should I reach them?
5. When should I reach them?

Who? This question corresponds to target selection. At first, the target market has to be defined. To whom does a company want to make an offer? Businesses or consumers? Existing customers or new? And the most important consideration: which prospects? The second question is whether the company has sufficient data on the subjects defined as target marker or that extra information is obliged? List brokers sell lists with, for example, zip code characteristics. Furthermore, it is obvious that one would like to address those customers, who are likely to respond. But specific campaign information can still be important because of the fact that the results of the target selection are not known yet. Demographic issues can play an important role: for instance if the company sells music records by mail and they have sponsored a concert, and the target selection gives all the customers who like the kind of music performed during the concert, it is not the smartest thing to invite all of them because not every music fan will travel hundreds of kilometers to see the concert. In this example, it would have been wise to only include those customers in the database,

who live in the surroundings of the location of the concert. This pre-selection can be seen as a form of feature selection which is dealt with in section 2.2.1. Now that the customer data base is built, the targets from this data base can be selected. This corresponds to the question: "Which customers are the targets of this campaign?" A general approach for the purpose of target selection (including the selection or building of the data base) is presented in section 2.2.

What? The proposition one would like to make has to be defined here. What is one selling and how should the recipient of the communication react? One of the first things to decide is whether the response should be one or two-stage; does one want merely an expression of interest or selling products or services immediately. One stage is sometimes called mail order, where as two-stage is referred to as direct mail advertising [2]. Next, the kind of response has to be determined; filling in a coupon, picking up the phone and dialing a (toll-free) number, go to a web-site or in another way. Make this completely clear, and go through the logistics.

Why? Although an enterprise might address its offer to the ones that are likely to respond, an attractive offer could help them to overcome the last doubt. This is where the unique selling proposition comes in.

Where? If there is only one address per customer available, there is not much to choose from. But in the professional market there can be a difference in sending to the office or home address. And in most cases even if the address of the company is known, it is often not entirely clear which person to address. This is due to the fact that people rotate in a company and even switch jobs outside a company. So in some cases it seems that one can better use a job title rather than a name.

When? Timing is one of the few parameters that can be completely controlled. Timing can make the difference between a successful campaign and one that fails. One thing to consider is the day of the week. Monday morning and Friday afternoon are not the best times to pick, for the business customers. In the consumer market the weekend seems to work, because most households have full-time responsibilities during the week nowadays.

The author of [2] leaves the "where" part out (a proper list must come up with the right addresses), combines the "why" and "what" and adds a "how". The how-question must come up with the most ideal way of communication. Postal service, e-mail or another communication element. Next to this, the author of [2] gives a relative importance in generating response. The right person (or list) is the most important factor, followed by a good offer and timing. The communication element has the least significance, which probably is the reason why the author of [3] has not mentioned this question in the given questionnaire.

As mentioned above, the "who question" corresponds to target selection in direct marketing: which prospects does one has to target? Target selection can be carried out following the Knowledge Discovery in Data cycle. What this is and how to do this are subjects covered in the next section.

2.2 Target Selection and KDD

This section provides an overview of the Knowledge Discovery in Data cycle: a general method very well suitable for target selection.

The KDD (Knowledge Discovery in Data) process is interactive and iterative, involving numerous steps with many decisions being made by the user. The intention, however, is to construct the process as autonomously as possible.
The preparing stage sometimes is referred to as the most important part of KDD. Results of a test mailing or previous mailings will structure the input of this stage. A test mailing is a mailing to a small representative part of a population. If this stage is not treated in a proper way the saying Garbage In Garbage Out will take over the best intentions. Depending on the nature of the data an algorithm for the data mining task has to be chosen. Finally the results in the form of models or rules have to be interpreted. Asking the help of a domain expert can be of great assistance. In the end the whole KDD cycle will end in a go or no go decision in case of direct marketing offers. So, we find three stages in KDD: preparing, data mining and interpretation.
KDD is made up of nine basic steps [10].

1. Developing an understanding of the application domain. Collect relevant prior knowledge and identify the customers' goal of the KDD process.
2. Selection: create a target data set.
3. Preprocessing: remove noise as far as possible, decide on strategies for handling missing values. This step is also called data cleaning.
4. Transformation: data reduction and projection. Find useful features to represent the data depending on the goal of the task. Use dimensionality reduction to reduce the effective number of variables under consideration.
5. Matching the goals of the KDD process to a particular data mining method.
6. Choosing the data mining algorithm(s).
7. Data mining: searching for patterns of interest in a particular representational form.
8. Interpretation of patterns found, and possibly return to any of the steps 1-7.
9. Consolidating the discovered knowledge: incorporating this knowledge in another system for further action. This system can be the use of the knowledge to predict the profitable prospects, or just reporting the results to the interested parties. This step also includes a validation with previously obtained results.

Preparing is step 2 to 4, data mining can be divided into step 5 to 7. If the first and the last step are left out, the KDD cycle is complete. The KDD process can involve significant iterations and may contain loops between any two steps. In the international literature, most effort is focused on the actual data mining step. The other steps are, however, equally important for a successful application of KDD in practice.

In the next subsections these three most important stages, preparing, data mining and interpretation will be addressed individually.

2.2.1 Preparing

The preparing stage consists of three steps: selection, preprocessing and transformation.

Selection

Selection in the phrase "target selection" addresses the entire procedure resulting in a selection of the prospects who are most likely to respond from a certain population. The selection of a test population which will act as "train" samples for the different configurations, is meant here. In the case of direct marketing, the selection will either involve previous mailings or a test mailing. A test mailing must address a representative subgroup of the entire population. This can be achieved by a random mailing or by addressing a subgroup, which has representative characteristics like a student population. Representative means that the selection has the same characteristics as the whole population, and that the size is not too small compared to the population (usually a few percent). One has to keep in mind that domain expertise can be of great help during the whole KDD cycle. Relatively expensive offers can better be not tested on a student population because of their limited cash.

Preprocessing

In this step cleaning and removal of noise are the basic operations. Data cleansing is the process of ensuring that all values in data set are consistent and correctly recorded. This definition has two important implications. In the process of gathering the data one has to make sure that this is done in a proper way. One has to check and recheck the sources. If the data is put in a database by data entries, a thorough check is needed in order to eliminate typing errors. Even if the information in a database is reliable, one has to cope with noise and missing values.

Starting with the noise, it is difficult to give a proper definition of noise. Noise can be slipped in the data by the procedure used to construct the data set. There can exist large correlations between features, if, for instance, they represent more or less the same characteristic. Another type of noise is referred to as "conflicting rows", which is the name of the appearance of exactly the same feature values and different response values. Although an ideal target selection configuration should be able to deal with this type of noise, beforehand it is not clear what the effect will be. So in every KDD process, the impact of these two types of noise has to be estimated and accordingly proper precautions have to be taken.

In case of missing values, the first thing to consider is the trade-off between the time spent on modeling missing values and the potential benefit. There are several ways to handle missing values in data fields:

- Remove all the columns or rows in which missing values are located. Obviously, this solution will only work for a very small number of missing values, a large amount of available data or a large number of responses.

- Assign the average, mean or modal value. This is simple, but can peak the distribution, which can become a huge problem when the feature plays an important role in the decision-making process. In case of a lot of missing values in the crucial attribute, this solution can give the wrong impressions. Therefore this solution will in general only be useful in case of a small number of missing values.

- Distribute the data using the probability distribution of the no missing records. Still relatively easy to apply, but if the assigned variable is important, errors will occur in the data mining results.

- Segment the data using the distribution of another variable, and assign segment averages to missing records in each segment. The results of this operation heavily depend on the correlation between the variables.

- Combine the previous two methods. Difficult to implement, and the gain again depends on the degree of correlation between the attributes.

- Build a classification model and impute the missing values. This is the best method of all, but very time consuming.

- The authors of [7] give a totally different approach: they introduce two fuzzy quantifiers, which indicate the degree of preference for a feature as a function of the number of unknown data records for that feature. One is labeled 'few missing values', and the other 'most variables allowed'. Thus, by combining both quantifiers, it is possible to determine the optimal percentage of missing values to be left in a data set. In case of a small data set, a high percentage could cause significant problems. This approach resembles the first one, but it does the removal more intelligently.

A final comment on the missing values is that the way they are handled, may depend on the algorithms that one is going to use for data mining. It is possible to handle the missing values and mine the data in the same computational procedure.

Transformation

Transformation is the last step before the actual data mining stage can be entered. Depending on the algorithms to be used in this stage, the data has to be set into a desired format. Next to this, feature selection is necessary in the case of too many attributes. This data reduction in the vertical sense depends on the goal of the KDD. Useful features have to be selected, and irrelevant removed. Finally, as mentioned earlier, different types of variables can give rise to problems. Depending on the algorithm used during the actual data mining step, some formats have to be transformed or removed. These three different types of transformation will be addressed individually.

Type	examples		
binary	0	1	0
continuous	2	$\sqrt{5}$	12
ordinal	Monday	Tuesday	Thursday
categorical	0-5	6-10	11-15
nominal	red	blue	yellow

Table 2.1: Different types of variables

Scaling

Depending on the mining algorithm in mind scaling can be a convenient transformation. When one has to cope with outliers, for example, a log-transformation can bring the values on a more narrow scale. Even if there are no outliers present, scaling could help to bring the values in a closer range, such that some features will not have a greater impact than others, just because of their larger scale.

Feature selection

Feature selection can be done in three different ways when applying the KDD cycle. First of all, one could think of a pre-selection based on specific campaign information about location or age, as mentioned in the "who question" in the previous section. This type of feature selection is a bit dangerous or futile because proper feature selection whether internal or external should also exclude these individuals. Internal and external is the distinction made between the other two types of feature selection. External feature selection is a selection based on some kind of relation found or given. This selection is nothing more than excluding a certain set of features before the actual mining stage. Internal feature selection, however, occurs as a loop over step 4 to 8. This means that features are excluded in a iterative way.

Different types of variables

Some algorithms can not cope with all types of variables. In order to give this statement more body, a short overview of the most common formats is given in table 2.1. Some mining algorithms treat all values as continuous. The problem faced with ordinal features is that there does exist a certain order (Tuesday follows Monday) but no value is dominant over (greater than) another. When nominal values are inhibited in a data set, the problem is even worse: there is no order and no dominance. Although ordinal and nominal can be transformed to continuous values, this procedure is arbitrarily: the nominal examples of table 2.1 could be transformed in 1-2-3, but 2-3-1 is a another possibility. So, if these types of variables occur in a data set great caution should be taken before transforming them to continuous attributes because new non-existing relations between the values are added.

2.2.2 Data mining

The goals of knowledge discovery are defined by the intended use of the system. The author of [10] distinguishes two types of goals: verification, where the system is limited to validate the user's hypothesis, and discovery where the system finds new patterns autonomously. The discovery goal can be subdivided into prediction, where the system finds patterns in order to predict the future behavior of entities, description, where the purpose is to present the patterns found to a user in a human-understandable form, or a combination of the two. Although the boundaries between description and prediction are not always sharp, the distinction can be useful in understanding to overall discovery goal. Various combinations of descriptive and predictive characteristics are listed in the following list of primary data mining methods [10]:

- Classification: learning a function that maps a data item into one of the predefined classes.
- Regression: learning a function, which maps a data item to a real-valued prediction variable and the discovery of parameters in the functional relationships between variables.
- Clustering: identifying a finite set of clusters to describe the data. (Closely related is the method of probability density estimation, which consists of techniques for estimating the joint multi-variate probability density function of all the variables.)
- Summarization: finding a compact description for a subset of data, e.g. the derivation of a summary or association rules and the use of multivariate visualization techniques.
- Change and deviation detection: discovering the most significant changes in the data from previously measured or normative values.

The next step is to construct specific algorithms to implement the methods mentioned above. According to [10], three components can be identified in each data mining algorithm: model representation, model evaluation and search.

Model representation

Model representation is the language used to describe discoverable patterns. More precisely, the category of models is defined here. If the representation is too limited, then no amount of training time or examples will produce an accurate model for the data. It is important that a data analyst fully comprehends the representational assumptions, which may be inherent in a particular method. It is equally important that an algorithm designer clearly states which representational assumptions are being made in a particular set of algorithms. More powerful representational models increase the danger in over fitting the training data, resulting in reduced prediction accuracy on unseen data.

Model evaluation criteria

Model evaluation criteria are quantitative or qualitative statements (or fit functions) of how well a particular pattern meets the goals of the KDD process. For

prediction models, computing the prediction accuracy on some test set can do this. Descriptive models can be evaluated along the dimensions of predictive accuracy, novelty, utility, and understandability of the fitted model.

Search method

The search method consists of two components: structure search and parameter search. Once the model representation and the model evaluation criteria are fixed, then the data mining task has been reduced to purely an optimization task: choose a specific structure from the selected family, which optimizes the evaluation criteria and find the parameters. In the parameter search the algorithm must search for the parameters, which optimize the model evaluation criteria given observed data and a fixed model representation. Structure search occurs as a loop over the parameter search method: the model representation is changed so that a family of models is considered.

2.2.3 Interpretation

The authors of [11] developed a theory regarding the value of extracted patterns. Their starting point is a general, but severe definition of KDD: extracting interesting patterns from raw data. Where there is some agreement on what patterns are in the literature, the question of how to interpret the "interesting" leads to disjoined discussions. A few characteristics are given: patterns could be interesting on the basis of their confidence, support, information content, unexpectedness or actionability. The latter means the ability of the pattern to suggest concrete and profitable action by the decision makers, and sounds promising. Their statement is: "a pattern in the data is interesting only to the extent in which it can be used in the decision making process of the enterprise to increase utility." Just finding patterns is not enough in the competitive environment that industries find themselves in these days. There must be an ability to respond and act on the patterns, ultimately turning the data in information, the information into action, and the action into profit.
Their point of view is, combined with classical linear programming, that an activity is interesting if the function describing this activity has a highly non-linear cross-term. Only then could data mining prove proper results. Depending on the nature of this non-linear behavior they propose a degree of interestingness. Because of their starting point of looking at data mining as an activity by a revenue-maximizing enterprise examining ways to exploit information it has on its customers, this could be a promising approach for the direct marketing application.

Although the perspective described above has interesting components, it lies outside the scope of this thesis. We aim at targeting prospects, and we do not have the intention to investigate the economic or financial consequences for a certain enterprise. The interpretation in this thesis heavily depend on the format the results are shown in. One of these formats is the subject of the next section.

2.3 Gain charts

Gain chart analysis is a general approach, described in [2]. By equating marginal costs and marginal returns one is able to determine which prospect should receive a mailing in order to maximize the expected profit. Gain chart analysis is based on a two-stage procedure. In the first stage, the response of a previous mailing in a particular target population (which can be a test mailing) is analyzed. The characteristics of the prospects that influence the response are identified, and their impact is quantified. This makes it possible to assign an index of prosperity. The likelihood to respond to a future mailing is calculated for each member of the population. In the next step, the members of the population are ordered by this index, from high to low. Prospects with similar index values can then be divided in groups of equal size and the average response per group is calculated. Then, those prospects are selected for the future mailing for whom the average group returns exceed expected costs of the mailing.

In this thesis, many results will be given in the form of a gain chart. The ultimate goal is to get the gain chart to leave and continue as vertical as possible, which means that more valuable prospects are identified. In other words, the goal is to try to increase the skewness of the gain chart. A example shows how gain charts are interpreted. If we consider the example gain chart of figure 2.1, the dashed line would represent an ideal gain chart: by mailing less then one-tenth of the total customer group all the respondents are found. The other extreme is when no target selection is used, but just a random mailing. The resulting gain chart for this case is the dotted one. No gains are obtained by a random mailing. The typical gain chart (solid curve) lies somewhere between the two extremes. In this particular case, by mailing 20 % of the total group around 50 % of the respondents are "caught". Another observation is that there are no respondents in the last 17 % of the total group.

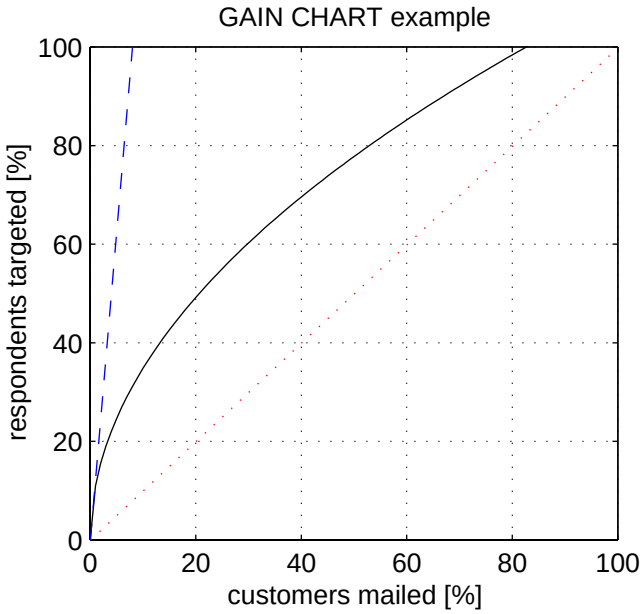

Figure 2.1: Gain chart explanation

Chapter 3

Mining algorithms

There are many algorithms available for the purpose of Target Selection. Although not one universal algorithm exists, some do better than others. The core of the decision process is a response model whose purpose is to assess the purchase propensity of each customer in the list prior to the mailing, as a function of the customers attributes and purchase history. A variety of approaches have been developed in the database marketing industry to model response, traditional human-driven segmentation methods involving RFM (recency, frequency and monetary) variables [19], tree-structured "automatic" segmentation models such as Automatic Interaction Detection (AID) [14] [17] [18], Chi Square AID (Chaid) [8], Classification and Regression Trees (Cart) [13] and C4.5 [24], linear statistical models such as linear [30], multiple regression [23] and discriminant analysis [25], non-linear discrete choice logit [9] [20] and probit [22] models. Linear regression is one of the algorithms which are subject of study. Although the relationship between the dependent and independent variables is not expected to be linear, this algorithm can prove reasonable results and has been chosen as a benchmark. Chaid and logit are also used in this project. These two are the most commonly applied algorithms in response modeling, and have proven good results [16].
Recent developments in artificial intelligence (AI) have triggered the use of AI-based methods to model response. Of these methods, artificial neural networks [21] and fuzzy modeling [7] are two examples which will be investigated, because we hope that these algorithms can deal with the assumed high complexity.

In this chapter, the theoretical background of the five models will be covered together with some additional implementation issues and commends on how to set up the optimal configuration. Starting with linear regression in section 3.1, the logit model follows in section 3.2, Chaid in section 3.3, the Neural Networks in section 3.4, and finally the Fuzzy modeling in section 3.5. Each section ends with a "conclusion" in which we address the individual strengths and weaknesses of the technique and then there are some direct comparison of techniques in terms of:

- Clarity and explicability. The transparency of the model is discussed here: can the decisions made by the model be explained?
- Implementation and integration

- Data requirements (including comments on the variable types)
- Accuracy of model
- Construction of model

Finally, in section 3.6 the algorithms are shortly compared and general comments are made about their differences based on the individual strengths and weaknesses sections.

3.1 Linear regression

The first method used is linear regression, which belongs to the family of regression models. Regression models view observations on certain events as the outcome of a random experiment. The outcomes of the experiment are assigned unique numeric values. The assignment is one-to-one; each outcome gets one value, and no distinct outcomes receive the same value. This outcome variable, Y, is a random variable because until the experiment is performed, it is uncertain what value Y will take. Probabilities are associated with outcomes to quantify this uncertainty. Thus, the probability that Y takes a particularly value, y, might be denoted $\text{Prob}(Y = y)$. So, in target selection if an individual responds, this is noted as $(Y = 1)$, if not $(Y = 0)$. The regression approach believes that a set of factors, gathered in a vector $\boldsymbol{x}$, explains the decision, so that

$$\begin{aligned} \text{Prob}(\boldsymbol{Y = 1}) &= \boldsymbol{F(\beta^T x)} \\ \text{Prob}(\boldsymbol{Y = 0}) &= \boldsymbol{1 - F(\beta^T x).} \end{aligned} \tag{3.1}$$

The set of parameters $\boldsymbol{\beta}$ reflect the impact of the changes in $\boldsymbol{x}$ on the probability. The response (discrete values 0 and 1) in linear regression is described in the linear regression equation:

$$\boldsymbol{y = \beta^T x + u} \tag{3.2}$$

In this equation, $\boldsymbol{u}$ represents the error term, of which the expected value is zero. Put in other words, the error term can take values other than zero, but on average it equals zero. Assuming that we have the availability of $\boldsymbol{y}$ and $\boldsymbol{x}$ in a data set, we can calculate an estimate $\boldsymbol{b}$ for $\boldsymbol{\beta}$. Now we can give a prediction for $\boldsymbol{y}$, or likelihood of response:

$$\boldsymbol{\widehat{y} = b^T x} \tag{3.3}$$

The error made in this prediction, is called a residual and is defined by:

$$\boldsymbol{e = y - \widehat{y}} \tag{3.4}$$

The prediction or estimation is usually done by ordinary least square (OLS) estimation. The basic idea of OLS is to choose those $\boldsymbol{b_i}$'s to minimize the sum of squared residuals:

$$\sum_{\boldsymbol{1}}^{\boldsymbol{n}} \boldsymbol{e_i^2}, \tag{3.5}$$

in which $\boldsymbol{n}$ represents the number of customers. It can be proven that minimizing equation 3.5 results in the following estimate for $\boldsymbol{\beta}$:

$$\boldsymbol{b = (x^T x)^{-1} x^T y} \tag{3.6}$$

When a test set comes up with an estimate for β, equation 3.3 can be used to predict a likelihood of response for the prospects whose characteristics x are known.
Now that the mathematical framework is described, the assumptions structuring linear regression are looked into. These Gauss-Markov assumptions guarantee that the estimates will have good properties [30].

3.1.1 Linear regression assumptions

1. $E(u) = 0$, the expected value of the errors is zero
2. The independent variables are non-random (and have finite variances)
3. The independent variables are linearly independent. Failure of this assumption is called multicollinearity (singularity).
4. $E(u^2 = \sigma^2)$, the disturbances u_i are homoscedastic: the variance of disturbance is the same for each observation.
5. $E(u_i u_j) = 0 \; \forall \; i \neq j$, disturbances associated with different observations are uncorrelated.

If the assumptions 1 to 3 are satisfied, then the OLS estimator b will be unbiased: $E(b) = \beta$. Unbiasedness means that if we draw many different examples, the average value of the OLS estimator based on each sample will be the true parameter value β. If all 5 assumptions hold, the variance of the OLS estimator equals $\text{Var}(b) = \sigma^2 (x^T x)^{-1}$. If the independent variables are highly correlated, the matrix $x^T x$ will become nearly singular and the elements of $(x^T x)^{-1}$ will be large, indicating that the estimates of β may be imprecise.

The assumptions mentioned above have some important effects on the configuration in which linear regression will perform best.

3.1.2 Feature selection and correlation

Because the following feature selection procedure is also based on linear relationships, it is mentioned here.
The overall aim is to keep the number of attributes as low as possible in order to reduce the model complexity. A real-life data set can contains hundreds of variables, which have to be reduced to a number around or lower than ten. This value is arbitrarily, but the less features the better.
We recall the correlation coefficient:

$$\rho(x_k, y) = \frac{\frac{1}{n} \sum_i (x_{ik} - \bar{x}_k)(y_i - \bar{y})}{\sigma_{x_k} \sigma_y}. \tag{3.7}$$

If we use these correlation coefficients of the variables, the following selection reduces the number of features (the model complexity). Only select those attributes that have an high absolute correlation with the the dependent attribute compared to the other attributes. If we use a threshold of at least two times

larger, features x_k are included if they satisfy the condition given in equation 3.8.

$$|\rho(x_k, y)| > \frac{2}{K} \sum_k |\rho(x_k, y)|. \tag{3.8}$$

3.1.3 Strengths and weaknesses

Probably the most important constraint of linear regression is that is not able to directly[1] account for non-linear relations. On the other hand, the main advantage is the ease of constructing and understanding the model. Another advantage is that the dependent variable can take any value. All dependent attributes are treated as continuous. Although F-tests can be applied, linear regression does not have straight-forward feature selection capabilities.

3.2 Logit

The second method, which will be used in modeling the individuals, is known as logistic regression or logit analysis, logit in short. Since that the dependent variable is binary: either there is a response or not, this form of regression is referred to as binominal logistic regression.
The following mathematical background can be found in [6].

In the logit model the right-hand side of the regression equation is

$$F(\beta^T x) = \frac{\exp(\beta^T x)}{1 + \exp(\beta^T x)}. \tag{3.9}$$

The estimation of a logit model is usually based on the method of maximum likelihood. Each observation is treated as a single draw from a Bernoulli distribution (binominal with one draw). We call the estimate for β b. The model with success probability $F(b^T x)$ and independent observations leads to the joint probability or likelihood function:

$$L = \text{Prob}(Y_1 = y_1, Y_2 = y_2, ..., Y_n = y_n) = \prod_{y_i=0} (1 - F(b^T x)) \prod_{y_i=1} (F(b^T x)) \tag{3.10}$$

Taking the logs, we obtain the log-likelihood, G:

$$G = \ln L = \sum_i \Big(y_i \ln F(b^T x) + (1 - y_i) \ln \big(1 - F(b^T x)\big) \Big) \tag{3.11}$$

The first-order conditions for maximization require setting the gradient to zero:

$$g = \frac{\partial G}{\partial b} = \sum_i (y_i - F(b^T x_i) x_i = 0 \tag{3.12}$$

The second derivatives for the logit model are based on:

$$H = \frac{\partial^2 G}{\partial b \partial b^T} = -\sum_i F(b^T x_i)(1 - F(b^T x_i)) x_i x_i^T \tag{3.13}$$

[1] Transformations such as a log transformation can be done if the attributes have logarithmic distributions (non-linear).

The Hessian is always negative definite, so the log-likelihood is globally concave. It will usually converge to the maximum of the log-likelihood in just a few iterations unless the data are especially badly conditioned. The asymptotic covariance matrix for the maximum likelihood estimator can be estimated by using the inverse of the Hessian evaluated at the maximum likelihood estimates.

The initial coefficients are compute by ordinary least squares estimation: then the log-likelihood is maximized with respect to the beta-vector. This can be done by applying Newton's method, which implies the following iteration (number of iterations is denoted by t):

$$\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \boldsymbol{H}^{-1}\boldsymbol{g}_t \tag{3.14}$$

The estimate for the beta vector is updated until a certain convergence criterion is met or a maximum number of iterations has been exceeded.

If the dependent variable has more than two classes, one could apply a multinominal logit. More about this subject and other binary choice models, can be found in [9] and [6].

3.2.1 Implementation and verification

The logit algorithm is implemented according to the procedure of 3.2. In this section a simple example is used to show that the algorithm works correctly.

The Matlab implementation of the logit algorithm used can be found in appendix **??**, and is called logit.m. A simple example is used to test the algorithm. The example comes from [6], and is listed in table 3.1. This particular data set

Table 3.1: Spector and Mazzeo data

gpa	tuce	psi	grade	gpa	tuce	psi	grade
2.66	20	0	0	3.28	24	0	0
4.00	21	0	1	2.76	17	0	0
3.03	25	0	0	2.63	20	0	0
3.57	23	0	0	3.53	26	0	0
2.75	25	0	0	3.12	23	1	0
2.06	22	1	0	2.89	14	1	0
3.54	24	1	1	3.39	17	1	1
3.65	21	1	1	3.10	21	1	0
2.89	22	0	0	2.92	12	0	0
2.86	17	0	0	2.87	21	0	0
3.92	29	0	1	3.32	23	0	0
3.26	25	0	1	2.74	19	0	0
2.83	19	0	0	3.16	25	1	1
3.62	28	1	1	3.51	26	1	0
2.83	27	1	1	2.67	24	1	0
4.00	23	1	1	2.39	19	1	1

was used to analyze the effectiveness of a new teaching method. The dependent

variable is "grade", an indicator whether students' grades on examination improved after exposure to "psi" (exposed or not), a new teaching method. The other variables are "gpa", the grade point average; and "tuce", the score on a pre-test which indicates entering knowledge of the material. The implemented logit algorithm showed similar results: the coefficients and t-ratios matched perfectly even in the third decimal. In table 3.2 the coefficients obtained by logit and the known coefficients are listed for this Spector and Mazzeo data set.

Table 3.2: Coefficients and t-ratios compared

	logit Coeff	Greene Coeff	logit t-ratio	Greene t-ratio
beta	-13.021347	-13.021	-2.640538	-2.640
gpa	2.826113	2.826	2.237723	2.238
tuce	0.095158	0.095	0.672235	0.672
psi	2.378688	2.379	2.234425	2.234

The only two parameters which can (default values are inhibited) be adjusted by the user are the maximum number of iterations or the convergence criterion. Testing the algorithm resulted in the observation that 6 iterations are enough for good results: even if the default convergence criterion ($\mathbf{1 \times 10^{-6}}$) is not met, results do not change significantly when more iterations are made.
Although the algorithm itself proves good results stand-alone, in this stage it can not be used as a target selection algorithm. Some adjustments and additional operations have to be applied before logit works as an proper target selection method. These issues will be covered in the remainder of this section, resulting in a "optimized" target selection algorithm.

3.2.2 Scaling

Scaling seems to be an plausible technique for logit, because in the implementation the t-test is used to look whether a calculated coefficient significantly differs from zero. If no scaling is applied, features with a large value scale will tend to be more significant than attributes with a modest scale. Therefore, all the variables are scaled to zero mean and variance one, a common transformation.

3.2.3 Feature selection

The last remark on the logit configuration, probably the most important issue, is how the feature selection can be done? Feature selection is the process of eliminating features until only significant features, the best predictors, are left. This can be done in the following way:
First, 100 random selections are made from the data set. On each of this selection the logit algorithm is performed. Then, sub-selections are chosen based on the significant features denoted by the previous run. In other words, the least important attributes are eliminated from the data set, based on the t-test of significance. This is repeated until the resulting data sets can no more be reduced, the features in the data set are all significant. Every selection assigns his own set of significant predictors. The significant features of each selection

are tabled, and for each feature is calculated by how many of the 100 selections it was found to be significant.

T-test

In order to determine which features have more impact on the dependent then other, a t-test can be used. Basically, the t-test on the logit coefficients tests whether a coefficient differs from zero, and expresses this with a degree of significance, the so-called t-statistic. At the 95 % significance level the t-ratio for the estimate of a coefficient is 1.96. The mathematical background of the t-test can be found in [6].

3.2.4 Strengths and weaknesses

Logit is known as a hard benchmark to beat: usually logit finds itself among the best performing algorithms in target selection environments [16]. However, logit does not automatically account for interaction effects [29]. Logit does not assume a linear relationship between the dependent and the features, but it does assume a linear relationship between the "logit" (see equation 3.9) of the features and the dependent. It may handle nonlinear effects even when exponential and explicit interaction and power terms can be added as extra features, but logit does not account for these effects automatically.
Another problem that can occur is known as multicollinearity. This means that the features are linear functions of each other. To the extent one variable is a linear function of other variables, the reliability of the estimate of the coefficient for this variable decreases [29].
Logit treats all feature values as continuous, just as linear regression. One requirement regarding the data is due to the use of maximum likelihood estimation: this procedure implies that the reliability of estimates decline when there are few cases for each observed combination of the variables. So, large samples increase the reliability.

3.3 Chaid

Chaid belongs to the set of decision trees. It is used for classification and prediction purposes. It has been successfully used to identify target groups of customers for direct mail for many years.

A decision tree represents a series of questions or rules, based on independent variables, shown as a path through the tree. Oddly, decision trees are shown going down from the root tree node towards the leaf tree nodes. A decision tree can be built using an algorithm that splits records into groups where the probability of the outcome differs for each group based on values of the independent variables. Chaid was first introduced in [8] as an extension from a technique called Automatic Interaction Detection (AID). Chaid uses the chi squared test to determine whether to branch further and if so which independent variables to use. Hence it's name Chi Squared Automatic Interaction Detection (CHAID). It was developed to identify interactions for inclusion into regression models. Chaid easily copes with interactions, which can cause other modeling techniques

difficulty. Interactions are combinations of independent variables that affect the outcome. For instance, profitability may be at the same level for low transactions in combination with high balance credit customers as for high transaction with low balance credit customers; in this case of modeling profitability, these two independent variables (transactions and balance) should not be considered in isolation.

The Chaid algorithm splits records into groups with the same probability of the outcome, based on values of independent variables. The algorithm starts at a root tree node, dividing into child tree nodes until leaf tree nodes terminate branching. Branching may be binary, ternary or more. The splits are determined using the chi squared test. This test is undertaken on a cross-tabulation between outcome and each of the independent variables. A cross-tabulation is a kind of frequency table: for each category of the predictor and each value of the dependent, the individuals with the same category value in combination with the same dependent value are counted and their sum is put into the cross-tabulation. The result of the test is a "p-value". The p-value is the probability that the relationship is spurious, in statistical jargon this is the probability that the Null Hypothesis is correct. The p-values for each cross-tabulation of all the independent variables are then ranked, and if the best (the smallest value) is below a specific threshold then that independent variable is chosen to split the root tree node. This testing and splitting is continued for each tree node, building a tree. As the branches get longer there are fewer independent variables available because the rest has already been used to further up that branch. The splitting stops when the best p-value is not below the specific threshold. The leaf tree nodes of the tree are tree nodes that did not have any splits with p-values below the specific threshold or all independent variables are used. This outlines a purely automated approach to building a tree. The best trees are built when a model builder checks each split and makes rational decisions (using background domain knowledge) as to the appropriateness of splitting on a particular variable at a specific point. The model builder can spot splits using independent variables that raises questions as to quality, hence avoiding problems of building an invalid tree, or model, on poor input data. A model builder may also decide to stop splitting at a higher level than the automated approach would stop, in order to produce a simpler model.

The step-wise procedure for Chaid as stated in [8] is as follows:

1. For each predictor in turn, cross-tabulate the categories of the predictor with the categories of the dependent variable and do steps 2 and 3.

2. Find the pairs of categories of the predictor (only considering allowable pairs as determined by the type of predictor) whose 2 by 2 sub-table is least significantly different. If this significance does not reach a critical value, merge the two categories, consider this merger as a single compound category, and repeat this step.

3. For each compound category consisting of three or more of the original categories, find the most significant binary split (constraint by the type of the predictor) into which the merger may be resolved. If this significance

is beyond a critical value, implement the split and return to step 2. [2]

4. Calculate the significance of each optimally merged predictor, and isolate the most significant one. If this significance is greater than a criterion value, subdivide the data according to the (merged) categories of the chosen predictor.

5. For each partition of the data that has not yet been analyzed, return to step 1. This step may be modified by excluding from further analysis partitions with a small number of observations.

3.3.1 The chi square test

The first step in computing the chi square test of independence is to compute the expected frequency for each cell in the cross-tabulation under the assumption that the null hypothesis is true. To put it in other words: calculate the expected cell frequencies assuming that the compared features are independent. The general formula for expected cell frequencies is:

$$\boldsymbol{E_{ij} = \frac{T_i T_j}{N}} \tag{3.15}$$

where $\boldsymbol{E_{ij}}$ is the expected frequency for the cell in the $\boldsymbol{i}$-th row and $\boldsymbol{j}$-th column, $\boldsymbol{T_i}$ is the total number of subjects in the $\boldsymbol{i}$-th row, $\boldsymbol{T_j}$ is the total number of subjects in the $\boldsymbol{j}$-th column, and $\boldsymbol{N}$ is the total number of subjects in the whole table. The formula for chi square test of independence is:

$$\boldsymbol{\chi^2 = \sum_{ij} \frac{(E_{ij} - a_{ij})^2}{E_{ij}}} \tag{3.16}$$

where $\boldsymbol{a_{ij}}$ is the value of the cross-tabulation in the $\boldsymbol{i}$-th row and $\boldsymbol{j}$-th column .

The degrees of freedom ($\boldsymbol{df}$) are equal to $\boldsymbol{(R-1)(C-1)}$, where $\boldsymbol{R}$ is the number of rows and $\boldsymbol{C}$ the number of columns. Now, a chi square table can be used to determine the probability for the calculated $\boldsymbol{\chi^2}$ and $\boldsymbol{df}$.

3.3.2 Significance of the predictors

Step 4 of the Chaid algorithm requires a test of significance of the reduced contingency table. If there has been no reduction of the original contingency table, a chi square test can be used.This test is conditional on the number of categories of the predictor, otherwise it must be viewed as conservative.
If categories have been merged in a predictor, a Bonferroni multiplier can be used according to [8]. This is needed to account for the number of ways a $\boldsymbol{c}$ category predictor of a given type can be reduced to $\boldsymbol{r}$ groups ($\boldsymbol{1 \leq r \leq c}$). The formulae for calculating these multipliers for the three types of predictor allowed by Chaid are:

[2] This third step does not specify how to find the required binary split. Using direct search, finding an optimal binary split for nominal variables requires time that is exponential in the number of categories.

1. Monotonic predictors. As in AID, a monotonic predictor is one whose categories lie on an ordinal scale. This implies that only contiguous categories may be grouped together. In this case the Bonferroni multiplier is equal to the binomial coefficient [8]:

$$B_{mon} = \begin{pmatrix} c-1 \\ r-1 \end{pmatrix} \tag{3.17}$$

2. Free predictors. Again as in the conventional AID, a free predictor is one whose categories are purely nominal. This implies that any grouping of categories is permissible. In [8], the multiplier is derived as:

$$B_{free} = \sum_{i=0}^{r-1} (-1)^i \frac{(r-i)^c}{i!(r-i)!} \tag{3.18}$$

3. Floating predictors. In some practical cases, the categories of a predictor lie on an ordinal scale with the exception of a single category that either does not belong with the rest, or whose position on the ordinal scale is unknown (like infinity or not-a-number). This is defined as the floating category, and the predictor is called a floating predictor. Again, the Bonferroni multiplier is derived in [8]:

$$B_{float} = \frac{r-1+r(c-r)}{c-1} B_{mon} \tag{3.19}$$

3.3.3 Implementation

Chaid is not implemented in Matlab. SPSS Chaid has been used instead. The reason is the lack of sufficient information on some critical decision points in the implementation. More information on SPSS Chaid can be found in [32].

3.3.4 Strengths and weaknesses

The form of a Chaid tree is intuitive, it can be expressed as a set of explicit rules in English. This means that the business user can confirm the rationale of the model and if necessary, modify the tree or direct it's architecture from their own experience or their background domain knowledge. Input data quality problems can be spotted, hence problems of building an invalid model on poor input data can be avoided. Also the most important predictors (or independent variables) can easily be identified and understood.
A Chaid model can be used in conjunction with more complex models. For instance, a Chaid model could identify who may be at risk of leaving and then a more complex profit model could be used to determine whether the customer is worth keeping.
Chaid models can handle categorical (like marital status) and banded continuous independent variables (like income). Continuous independent variables, like income, must be banded into categorical-like classes prior to use in Chaid. In particular if the independent variables are categorical with high cardinality (implicit "containing" relationships) Chaid should perform even better. A Chaid model can automatically prevents over-fitting and handle missing data. Chaid

does not require much computational power.
Chaid needs rather large volumes of data to ensure that the number of observations in the leaf tree nodes large enough to be significant. Continuous variables must be banded.

3.4 Neural Networks

Neural Networks (NNs) are biologically inspired models which try to mimic the performance of the network of neurons, or nerve cells, in the human brain. Expressed mathematically, a NN is made up of a collection of processing units (neurons, nodes), connected by means of branches, each characterized by a weight representing the strength of the connection between the neurons. On the first try, since the NN is still untrained, the input neuron will send a current of initial strength to the output neurons, as determined by the initial conditions. But as more and more cases are present, the NN will eventually learn to weight each signal appropriately. Then, given a set of new observations, these weights can be used to predict the resulting output.

Neural Networks are often mentioned in many theoretical and practical journals as a promising and effective alternative to conventional response modeling in database marketing for targeting audiences through mail promotions. NNs employ the results of a previous mailing campaign, for which it is known who responded with an order, and who declined, to train a network and come up with a set of "weights", each representing the strength of connection between any two linked nodes, or neurons, in the network. These weights are then used to score the customers in the data set and rank them according to their likelihood of purchase; usually, the higher the NN score, the higher the propensity to purchase.

The neurons are usually grouped in several layers. One input layer, one or more "hidden" layers and an output layer. One example of a Neural Network used in this thesis is the so-called one-layer supervised Feedforward-Backpropagation Neural Network.

3.4.1 The feedforward-backpropagation NN

Supervised means that the NN gets both the input and known output values. *Feedforward*, because the only direction for information flow is from the input layer to the output layer. In this way, from the inputs the outputs of the first layer are calculated. These form the input of the second hidden layer, and this goes on until the output layer has been reached. The input layer neurons do not perform any computations, but merely distribute the inputs x_i ($i = 1 : p$) to the weights w_{ij}^h of the hidden layer (j represents the number of neurons in the layer). The weighted inputs are add up for each neuron (resulting in a value z_j) and passed through a non-linear function σ, which is called the activation function. The value of this function $v_j = \sigma(z_j)$ is the output of the neuron. There exist several activation functions. The three activation functions most commonly used are: Purelin, Tansig and Logsig. The transformations are defined as follows:

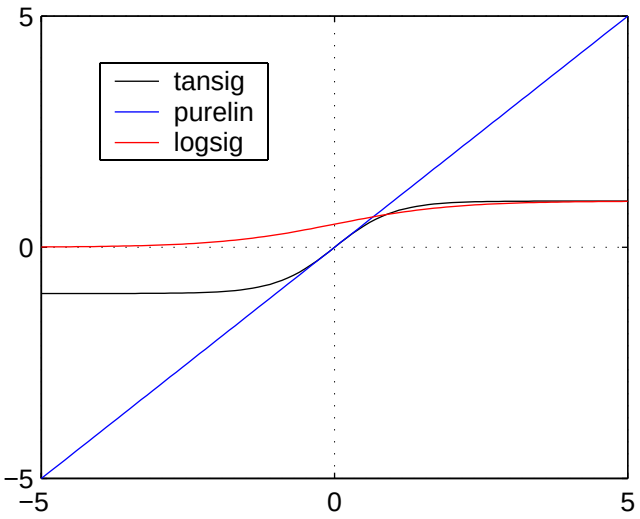

Figure 3.1: Activation functions

$$purelin(n) = n \tag{3.20}$$

$$tansig(n) = \frac{2}{1 + \exp(-2n)} - 1 \tag{3.21}$$

$$logsig(n) = \frac{1}{1 + \exp(-n)} \tag{3.22}$$

In figure 3.1 the operations are made visual for n from -5 to 5: purelin does not perform any transformation, tansig transform the values outside the -1 to 1 range to these limits and through the logsig "filter" only frequencies between 0 and 1 pass, the rest is made 0 or 1 depending on the sign (positive or negative). Now the output has to be calculated [12] [3]:

$$y_l = \sigma^o(\sum_{j=1}^{l} w_{jl}^o v_j) \tag{3.23}$$

Backpropagation, because the weights are adjusted by minimizing the squared errors in the other direction. The known output (supervised learning) is compared with the output of the output layer. The difference of these two values is called the error:

$$e_l = d_l - y_l \tag{3.24}$$

where d represents the desired output. This error is used to adjust the weights of the NN, starting with those in the output layer, assuming that the activation function of the output layer is linear;

$$w_{jl}^o = w_{jl}^o + \alpha v_j e_l \tag{3.25}$$

[3] No threshold is used in this particular procedure.

Then we adjust the hidden layer weights, mathematically expressed as:

$$w_{ij}^h = w_{ij}^h + \alpha x_i \sigma_j^T(z_j) \sum_l e_l w_{jl}^o \tag{3.26}$$

Before the training can start, the weights have to be randomly initialized. The updating of the weights is repeated until the error has reached a criterion value. One run is called an epoch.

3.4.2 Implementation

The neural network toolbox version 3.0.1 (R11) is used. The function "newff" builds a feed-forward back-propagation network. As training function is chosen for "trainlm": the Levenberg-Marquardt back-propagation training.

3.4.3 Strengths and weaknesses

NNs can be applied to both directed (supervised) and undirected (unsupervised) data mining. NNs can handle both categorical and continuous independent variables without banding.
NNs can produce a model even in situations that are very complex because an NN produces non-linear models.
The independent variables must be converted into the range from 0 to 1, this is done using "transformations" which can be inaccurate when the independent variables are skewed with a few outliers.
The output from an NN is usually continuous which may be difficult to transform to a discrete categorical outcome.
Several parameters must be set up, for instance the number of hidden layers and the number of nodes per hidden layer. These parameters affect the model built. The results of small differences in these parameters can be the difference between a very predictive model and a poor model. So, experience in building NNs is almost a demand for a fair configuration.
The results of an NN cannot be explained, it is a "black-box", a set of weights with no inherent meaning. Recently, some explanation of an NN may be obtained by using additional techniques to visualise the networks and to produce rules from prototypes (using sensitivity analysis). Explicability is a legal requirement for credit product application models in the US, this means that NNs cannot be used to build credit risk models. The lack of clarity means that unfair prejudice cannot be ruled out from the credit decision.
NNs can produce models that are sub-optimal.
To build an NN model requires an experienced statistician or expert NN user to ensure that the model is not over-fitted.
This method can be very time-consuming because of the number of re-presentations of the data that is required during training. Also if there is a large number of predictive variables, then the time taken to find a solution are further lengthened. The skill and effort required to build an NN plus the time involved means that this technique is costly.

3.5 Fuzzy modeling

The fuzzy modeling algorithm used in this thesis is proposed and fully described in [7]. In this section a short review is given.

Although the model presented has an procedure to deal with missing values (see section 2.2), pre-processing tasks are not further mentioned in this section. The model follows the two-fold analysis of direct marketing campaigns: first feature selection must determine the variables that are relevant for the specific target selection problem. Second, rules [4] for selecting the customers should be determined, given the relevant features.

3.5.1 Feature selection

In this fuzzy model, feature selection is done using gain charts from individual models made for each variable. This means that each variable is modeled using fuzzy c-means clustering, which allows cluster merging and variable sized cluster prototypes, and gain analysis, leading to additive improvement of the gain charts per feature and to the total model as models of individual features are added to the total model in a decision tree like manner. First the clients ($\boldsymbol{k = 1 : N}$) are ranked per feature ($\boldsymbol{j = 1 : n}$). By assigning a fuzzy Respondent Density (RD) for each cluster $\boldsymbol{M_j}$ of each variable $\boldsymbol{j}$, which represents the ratio of the total membership of the positive respondents in the cluster to the total membership of all clients $\boldsymbol{N}$ in the cluster.

$$\boldsymbol{RD_{ij} = \frac{\sum_{k=1}^{N} \mu_{ij}(x_{kj}) y_k}{\sum_{k=1}^{N} \mu_{ij}(x_{kj})}, 1 \leq i \leq M_j, j = 1, 2, ..., n} \tag{3.27}$$

In equation 3.27 $\boldsymbol{\mu_{ij}(x_{kj})}$ represents the membership of the $\boldsymbol{k}$'th client in the $\boldsymbol{i}$'th cluster of feature $\boldsymbol{j}$. These RD values are then used to compute a score (SC) for each client for each feature.

$$\boldsymbol{SC_{jk} = \sum_{i=1}^{M_j} \mu_{ij}(x_{kj}) RD_{ij}, 1 \leq k \leq N} \tag{3.28}$$

The score SC for each client for each feature is made up of a weighted sum over all clusters of the product of the membership value for each cluster, the value of the client for the particular feature and the RD value for the cluster. By doing so, the higher the SC score, the more plausible that the particular client is a respondent, according to the feature examined. Next, gain charts are calculated for each feature. Each gain chart is assigned a score SX which reflects its targeting efficiency. Feature selection can now be done in an iterative manner, the feature with the highest SX score, e.g. the most favorable gain chart, is the most important feature according to this methodology. The cluster centers and the corresponding RD values for the selected feature are recorded as part of the final model. Now the clients whose SC scores for the most important feature

[4] Rules are preferred, because domain experts can verify them with their experience. Some algorithms (like NNs) do not come up with explicit rules but do result in a selection model that can assign an index of prosperity to a individual (see section 3.6 for more information on this subject).

are in fact the highest SC scores compared to all SC score for the other features, are removed from the data set.
In this way, the clients that are best described by this feature are identified. This process continues until a maximum depth is met, in other words a (user specified) maximum number of important features is reached. Other stopping criteria are possible: for example, the process stops when no more respondents are left in the data set (all respondents are described by the selected features).

3.5.2 Target selection

Now that the subset with the most attractive features is found, the individual models of each feature are combined in a final model. The ranking and selection of the clients is done by averaging the results from each feature. The target selection score for a client is the sum of the SC scores of the feature subset divided by the number of features in this set.

3.5.3 Strengths and weaknesses

The fuzzy modeling technique makes it possible to capture some of the structures in the data for which one normally would have to consider the interactions between the features [7].
The fuzzy modeling approach provides an advantage to many other AI techniques, since the model can be represented in a rule-based form that analyst can verify against their own knowledge and experience.
Because of the rather complex structure of the fuzzy modeling technique, many calculations have to be made. This is the reason why the algorithm is quite time consuming compared to a method like Chaid.

3.6 Summary of target selection algorithms

In this section we shortly review all five target selection algorithms and discuss their differences and suitability.

The first remark addresses the availability of sufficient data. Although all target selection algorithms need representative train samples, Chaid requires more samples than the other four algorithms. If there are not enough discriminating attributes, Chaid can better be used for feature selection and another method to score the prospects. This is because in the case of a few leaf nodes, Chaid assigns just a few different scores for all prospects. On the other hand, a Chaid tree is relatively easy to build, and available in several commercial software packages. It is, however, difficult to implement Chaid yourself, not because of the difficult structure but because of the lack of relevant background information on several specific choices.
Neural networks are quite laborious: the number of neurons in the hidden layer has to be determined by trail and error. Furthermore, the time to train a neural network can be quite long, especially in case of a large number of attributes and the network is a blackbox; no linguistic rules can be deducted like the Chaid and fuzzy modeling algorithm do have this capability. The fuzzy modeling technique has another advantage: it handles the feature selection and client scoring

in the same computational procedure. One disadvantage is that the training requires a lot of time compared to Chaid and the regression techniques. Logit and linear regression are fast procedures, but because the procedures involve matrix inversion, not all data sets can be presented, especially if they are bad conditioned. So, these algorithms are in particular suitable for the task of client scoring where another algorithm should determine the relevant features.

Recapitulating, based on the differences between the models *feature selection* can be best done by Chaid and the fuzzy modeling algorithm. The feature selection procedure described in section 3.1.2 is very easy and based on linear relationships. For the purpose of *client scoring*, linear and logistic regression, neural networks and the fuzzy modeling algorithm are suited best. The latter is best used without external feature selection because this can demolish the specific structures which the fuzzy modeling technique tries to find.

Chapter 4

Methodology

In this chapter the previous two chapters are used in order to derive the methodology to really conduct some "intelligent" target selection. This chapter can be roughly divided in three main parts. The first part is the introduction of the data set and its treatment as application domain. In this way, the abstract theory of the KDD cycle is employed in a real business problem. By doing so, we believe that the different steps are explained in a clearer way to the reader. The second part is an explanation of different numerical performance measures which will be used to compare the different configurations. The measures eventually used, are derivatives of the gain chart analysis (section 2.3). The last and most important part is the practical filling of the KDD cycle of chapter 2: the methodology to come up with the different configurations for intelligent target selection in direct marketing.

4.1 Data set and direct marketing

The data set used in this thesis work is a real-life data set used in a contest and very representative for all issues concerned in deploying target selection in direct marketing. The original assignment and description of this data set follows first. The rest of this section tries to give the answers to the five questions, typical for direct marketing campaigns, as mentioned in chapter 2.

4.1.1 Original TIC assignment

"Direct mailings to a company's potential customers, many of them see this as "junk mail", can be a very effective way for them to market a product or a service. However, as we all know, much of this junk mail is really of no interest to the people that receive it. Most of it ends up thrown away, not only wasting the money that the company spent on it, but also filling up landfill waste sites or needing to be recycled. If the company had a better understanding of who their potential customers were, they would know more accurately who to send it to, so some of this waste and expense could be reduced. So here is our challenge:

Can you predict who would be interested in buying a caravan insurance policy and give an explanation why?

The competition consists of two tasks:

1. Predict which customers are potentially interested in a caravan insurance policy.

2. Describe the actual or potential customers; and possibly explain why these customers buy a caravan policy.

Participants need to provide a solution for both tasks. We want you to predict whether a customer is interested in a caravan insurance policy from other data about the customer. Information about customers consists of 86 variables and includes product usage data and socio-demographic data derived from zip area codes. The data was supplied by the Dutch data mining company and is based on a real world business problem. The training set contains over 5000 descriptions of customers, including the information of whether or not they have a caravan insurance policy. A score set contains 4000 customers which you will classify. A portion of your performance will be judged on your accuracy of classification of this set. For the prediction task, the underlying problem is to the find the subset of customers with a probability of having a caravan insurance policy above some boundary probability. The known policyholders can then be removed and the rest receives a mailing. The boundary depends on the costs and benefits such as of the costs of mailing and benefit of selling insurance policies. To approximate this problem, we want you to find the set of 800 customers in the score set that contains the most caravan policy owners. For each solution submitted, the number of actual policyholders will be counted and this gives the score of a solution. Only the indexes of the selected records need to be provided, assuming that the first record has index number 1 (e.g. 1,7,24,...,3980,4000). The purpose of the description task is to give a clear insight to why customers have a caravan insurance policy and how these customers are different from other customers. Descriptions can be based on regression equations, decision trees, neural network weights, linguistic descriptions, evolutionary programs, graphical representations or any other form. The descriptions and accompanying interpretation must be comprehensible, useful and actionable for a marketing professional with no prior knowledge of computational learning technology. Since the value of a description is inherently subjective, submitted descriptions will be evaluated by experts in insurance marketing and data mining."

The distinction between the train (tic data set) and final validation (score set) set is very important to keep in mind.

4.1.2 The direct marketing questions

Although the answers to the questions are somewhat straight-forward, they are instructive and provide a complete overview of the method presented in this thesis. The five questions to be considered for a successful direct marketing campaign are (see section 2.1):

1. Who am I trying to reach?
   Individuals interested in a caravan insurance.

2. What do I want them to do?
   It is obvious that the prospects should buy a caravan insurance. To lower the threshold, the intended reaction of the prospects must be made as easy as possible. If telephone numbers are available, one could consider phoning the prospects a few days after sending the offer to ask them about their interests in caravan insurance policies and their opinion on the offer.

3. Why should they do it?
   This is not completely clear. Nothing is mentioned about an attractive offer in the assignment. One could think of discount packages with, for instance, car insurances.

4. Where should I reach them?
   Because the target market is the consumer market, home addresses are most likely to be used in this particular campaign.

5. When should I reach them?
   One thing to consider here is the time of the season. Spring could work, because people are planning their summer holidays and might be considering the purchase of a caravan.

4.2 Performance measures

In this section, the model evaluation criteria are defined. In order to make statements about the performance of different target selection configurations, several means of comparison can be used. First of all, scatter matrices can provide insight in the performance. A scatter matrix for response modeling is a 2 by 2 matrix in which the rows represent the predicted response $\hat{\boldsymbol{y}}$ and the columns the actual response $\boldsymbol{y}$. In table 4.1 an example of a scatter matrix is

Table 4.1: Example scatter matrix

SCATTER	$\boldsymbol{y = 0}$	$\boldsymbol{y = 1}$	total
$\hat{\boldsymbol{y}} = \mathbf{0}$	654	23	677
$\hat{\boldsymbol{y}} = \mathbf{1}$	286	37	323
total	940	60	1000

given. From this table can be seen that 37 out of 60 respondent are successfully classified, which is equal to 61.6 %. This percentage is denoted as $\boldsymbol{P_1}$. Equally, the value for $\boldsymbol{P_0}$ in this example is 69.6 %. Although scatter matrices can provide insight in the percentages of successful classifications, there are two disadvantages of using this measure to compare different configurations. It is not clear which percentage or what combination of $\boldsymbol{P_0}$ and $\boldsymbol{P_1}$ to use. On the one hand a high value of $\boldsymbol{P_1}$ increases the number of respondents, but this usually causes $\boldsymbol{P_0}$ to decrease resulting in a higher cost per order. On the other hand, a large value for $\boldsymbol{P_0}$ increases customer annoyance, which is difficult to express in terms of profit loss but still present. Hence, there is a trade-off between $\boldsymbol{P_0}$ and $\boldsymbol{P_1}$ which can not be numerically expressed.

4.2.1 Numerical Gain Chart Value

Using gain charts (section 2.3) for comparing different target selection configurations is difficult when the curves are much alike. Several numerical measures can be defined, a surface measure which would represent the area between the gain chart curve and the 45 degree reference line (the dotted line in figure 2.1), or a measure which represents the percentage respondents targeted at a certain fixed percentage of the total group mailed. One could think of a certain combination of the two mentioned. Let $\boldsymbol{P_{20}}$ denote the percentage of respondents at the 20 percent level of the total group, in the same way: $\boldsymbol{P_{50}}$ represents the percentage of respondents when 50 percent of the total groups is mailed. The measure used is denoted by Numerical Gain Chart Value (NGCV) and is defined by:

$$\text{NGCV} = \boldsymbol{\alpha}(\boldsymbol{P_{20}} - \mathbf{0.2}) + \boldsymbol{\beta}(\boldsymbol{P_{50}} - \mathbf{0.5}) \tag{4.1}$$

The choice for the weight factors $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is somewhat arbitrarily, the idea behind it is that the importance of a high skewness of the gain chart is expressed by the ratio of the two coefficients. So, $\boldsymbol{\alpha}$ should be larger than $\boldsymbol{\beta}$. The absolute values are not that important: NGCV is merely a measure to compare several configurations, the value itself does not have a direct meaning: it is made up of two weighted gain values of response percentages to random selection. In this line of thought, $\boldsymbol{\alpha}$ can be set to 0.7 and $\boldsymbol{\beta}$ to 0.3.

4.2.2 CoIL Performance Measure

The performance measure used in the CoIL challenge was the number of respondents in the first 20 % of the score set after the prospects are ordered by a configuration in terms of the likelihood of response. This number is denoted by CoIL Performance Measure (CPM). We shall use this performance measure in this thesis because results can be easily compared to the results of the CoIL contestants.

4.3 Target Selection by KDD

This section provides the filling of the Knowledge Discovery in Data cycle as presented in section 2.2. The intention is to construct the process as autonomously as possible. This constraint is added because we want to develop some kind of independent agent who can do the target selection for us and does not need interference of domain experts who are in general very expensive to consult. We will now fill all the gaps in the nine basic steps.

1. Developing an understanding of the application domain.

   Domain expertise is gained in section 4.1.2 and the customers' goal of this KDD process is the purchase of a caravan insurance policy.

2. Selection: create a target data set.

   The target data set used in this thesis work, is the training set, mentioned in section 4.1. It contains 5822 client records, of 86 attributes each. The first 43 features are zip code characteristics and the rest are features

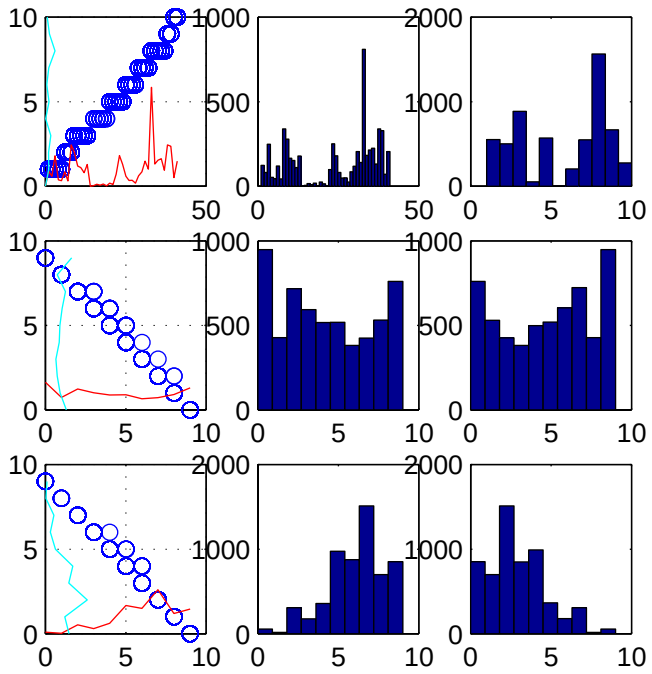

Figure 4.1: Correlations of feature pairs 1 and 5, 30 and 31, 35 and 36

related to products owned by the prospects. All 86 features present in the ticdata set are listed by number in tables A.1 and A.2. Their labels, together with a short description are given. The domain refers to a specific type of category, shown in several tables which also can be found in Appendix A.
For example, if an individual in the training set has value 6 for the fifth feature, this means that this individual according to his zip code characteristics is retired and religious. A value of 7 for feature 32 means that 76 to 88 percent of the households living in the same the zip code area as the prospect, owns 1 car. Finally, a value of 4 for feature 59 represents a contribution between 200 and 500 Dutch guilders paid by this prospect annually for all his fire policies.

3. Preprocessing.

   Because this particular data set does not have any missing values, the occurrence of noise is the only thing to investigate. Because the ticdata set suffers from both large correlations between the variables as the occurrence of conflicting rows, both phenomena are looked into.

4.3.1 Correlation between the Features

Merely by looking at the description of the variables, one good easily imagine that large correlations between the variables exist. This section

covers the treatment of these correlations. The goal is to come up with a proper foundation for a first external feature selection in order to lower the complexity of the model. Although the computational power of computer equipment increases day by day, it is a huge advantage to the complexity of the data set as small as possible.
At the 90 percent level, two things draw attention. One the one hand, there are three pairs of features with a strong correlation, that remains more or less the same in the entire value scope. On the other hand, the other pairs concentrate at the first (zero) category. These pairs consist of respectively the contribution and absolute numbers of several other insurance products. Because of the fact that the largest part of the correlation is due to the focus around the first category (just a few individuals bought this insurance), no transformation or elimination can be made. The first three pairs, however, can be transformed into three variables, without losing any information. The correlations are shown in figure 4.1. Every row of the three figures represents a feature pair. The first figure indicates the correlation denoted by the small circles. The other two are histograms, also plotted in the first as a line on each axe. The first pair is feature number one, Customer sub type, and number five, Customer main type. Both features have an unique category listing (see tables A.3 and A.5). As can be seen from figure 4.1, leaving out feature number five will not lead to any information loss. The same holds for the second (feature 30, Rented house, and 31, Home owners) and third pair (feature 35, National Health Service, and 36, Private Health Insurance). The only difference is that there exist overlapping categories due to quantization errors: the categories represent percentage intervals (see table A.6), when the intervals had been chosen more carefully, the correlations had been on a straight line. So, for this particular data set, removing features 5, 30 and 35, is a legible thing to do.

4.3.2 Conflicting rows

Another interesting observation is that the data set inhibits about 70 conflicting rows. As mentioned earlier, we call two rows "conflicting" when all the feature values are exactly the same, but the dependent variable differs. This is inherent to target selection in direct marketing: two individuals can look exactly the same to the outside world, while their interests in certain products do not match. For the algorithms, however, conflicting rows can cause problems. How deals an algorithm with a case in which first a sequence is called 0 and later on the same sequence carries a dependent value 1? The expectation is that the scoring of an individual should in case of two different training examples with the same feature values, result in a likelihood to response of 50 percent. If more training samples of one dependent value occur, the likelihood should alter accordingly. So, in an ideal configuration conflicting rows should not give rise to any problems.

4. Transformation: data reduction and projection.

   Transformations can be divided into three independent procedures: scal-

ing, feature selection and the treatment of different kind of variables (see section 2.2.1). All three are addressed individually.

4.3.3 Scaling

Some algorithms use in their internal feature selection the so-called T-test. For these algorithms a normalization is required. The T-test is used to investigate whether a calculated coefficient significantly differs from zero. If no scaling is conducted, features with a large value scale will tend to be more significant than attributes with a modest scale. Therefore, all the variables are scaled to zero mean and variance one, a commonly applied transformation.
Because the tendency is that the configuration that probably performs best is made up of several algorithms, this scaling is applied because no harm is done (values still keep their underlying relations) and the calculations are easy and carried out in a split-second.

4.3.4 Feature selection

The three features which can be excluded according to their strong correlation with other features (section 4.3.1) can be excluded here as an external selection. Some configurations make use of external feature selection, others use internal feature selection, which actually is nothing more than external selection carried out a number of times in a loop.

4.3.5 Different types of variables

The data set has four types of variables: 62 categorical, 21 continuous, 2 nominal and one binary attribute. The categorical can be treated as continuous and the binary attribute is the dependent variable. The two features to worry about are number 1, customer subtype, and number 5, customer main type. As found in section 4.3.1, these two features have a very large correlation. Since feature number five can be excluded, one nominal predictor is left. There are a few options how to deal with this predictor. The simplest one is to leave it out, but we will see that this is one of the more important predictors. Another option is to transform the nominal values to continuous ones, but this adds new relations between the nominal feature predicates. This does not have to be a enormous problem, but every transformation can be justified. The approach we choose, depends on our faith in the expertise of the people responsible for constructing the category listing. Each value is transformed to its place in the list. So, "stable family" become 10 and "mixed rurals" closes the listing with 41. Whether this transformation can be justified, becomes clear in chapter 5, where the results are presented.
If one has to build its own nominal category list, the aim has to be to establish some kind of ordinal sequence, if possible. With a listing as customer sub type, one could think of a list increasing in social status. Although the subdivision always stays arbitrarily, this approach is better then just

transforming the value according to, for instance, their appearance in the alphabet.

5. Matching the goals of the KDD process to a particular data mining method.

   The goal of this KDD process is to predict which customers are likely to buy a caravan policy.

6. Choosing the data mining algorithm(s).

   The mining algorithms used in this these are linear and logistic regression, Chaid, Neural Networks and a fuzzy modeling technique. Each algorithm is fully described in chapter 3.

7. Data mining: searching for patterns of interest in a particular representational form.

   The mining of the algorithms consist of two different phase in this thesis work. The first phase consist of the five stand-alone algorithms in their most optimized configurations. The second phase is the introduction of the combinations which uses some results from the first phase.

4.3.6 Combining the algorithms

The five mining algorithms described in chapter 3 all have different backgrounds and constrains. This statement is intuitive and the second foundation for the combining is a practical reflection: all algorithms roughly consist of two major components; one which accounts for the feature selection and another that scores the customers according to their likelihood of response. Although some algorithms execute both tasks in the same computational procedure, every algorithm can be added an external feature selection. Magidson [16] reported that selecting features using Chaid and scoring by logit is a powerful combination. This procedure can be applied in general: one configuration selects the important features and a second configuration uses these features for the client scoring. Because there are usually a lot of attributes in a direct marketing database, feature selection is necessary to identify the important features. Every algorithm selecting features, however, suffers from algorithm specific assumptions: normally distributed data, for instance. Because the total number of attributes is high, several feature selecting algorithms can be applied. These rarely come up with exactly the same feature subsets. If we assume that we have sufficient feature selecting algorithms (a domain expert can also be an "algorithm"), we have the availability of several feature subsets which inhibit relevant attributes to the target selection problem.
The next question is how to assign a score to each client, given these feature subsets. One option is to construct one large feature set, with all features present in the feature sets given by the feature selecting algorithms. The disadvantage of this approach is that the unique structures of the former feature sets are lost. Another related problem, is what configuration should be used to score the clients only using the features present

in this set? Other options of combining feature sets before scoring the prospects will have to conquer the same problems. The solution is simple: score all feature sets separately by algorithms which are known or expected to do well. This results in different scores for each client in the database. Assign all scores a weight depending on the expected performance by the specific configuration and multiply the scores with these weights before adding them up resulting in a final client score.

Mathematically expressed, if we have a data set with $\boldsymbol{N}$ customers and $\boldsymbol{K}$ attributes, and $\boldsymbol{R}$ feature selecting algorithms ($\boldsymbol{FSA_r}$) selecting $\boldsymbol{R}$ feature subsets $\boldsymbol{FT_r}$ with $\boldsymbol{k(r) < K}$ features each, we also need $\boldsymbol{R}$ scoring algorithms ($\boldsymbol{SA_r}$) giving each customer $\boldsymbol{n \leq N}$ a score $\boldsymbol{sc_{nr}}$ and the total score for client $\boldsymbol{n}$ is given by:

$$\boldsymbol{SC_n = \sum_r \alpha_r sc_{nr},} \tag{4.2}$$

in which $\boldsymbol{\alpha_r}$ is the weight factor for scoring algorithm $\boldsymbol{r}$. In the ideal case, the sum of these weights equals 1:

$$\boldsymbol{\sum_r \alpha_r = 1.} \tag{4.3}$$

There are three special cases:

(a) The feature selecting algorithm can be the same for all $\boldsymbol{R}$ scoring algorithms, hence the same feature (sub)set is used for different scoring configurations. Example: finding the optimal number of neurons in the hidden layer of a Artificial Neural Network.

(b) The scoring algorithm can be the same for all $\boldsymbol{R}$ feature selecting algorithms. Example: finding the best feature subset for logistic regression as scoring algorithm.

(c) The feature selecting algorithm is absent or the same as the scoring algorithm and $\boldsymbol{R}$ equals one. This is target selection in the "old-fashioned" way.

Next to the special cases, by manipulating the weight factors, one can build an algorithm selecting–target selection configuration. The total method is summarized in the flowchart given in AD1. The only problem left is how to determine appropriate values for the $\boldsymbol{\alpha}$'s.

Setting the Weight Factors

The weight factor gives a configuration a degree of importance with respect to the other configurations used in a combination. Different configurations rarely come up with scores on the same value scale. So, the second task of the weight factor is transforming the scores given by the different configurations to the same scale. In order to establish a proper distinction between both tasks the weight factor can be split up into two factors: one representing the relative degree of importance and the other to scale the scores.

$$\boldsymbol{\alpha_r = \beta_r \gamma_r} \tag{4.4}$$

In equation 4.4 the weight factor $\boldsymbol{\alpha}$ is split into $\boldsymbol{\beta}$, which represents the degree of importance and $\boldsymbol{\gamma}$ which is the scaling factor. The latter can easily be computed:

$$\boldsymbol{\gamma_r} = \frac{\boldsymbol{N}}{\sum_{\boldsymbol{n}} \boldsymbol{sc_{nr}}} \tag{4.5}$$

Four different sets of weight factors are used. The first, straightforward, way is to give all $\boldsymbol{\alpha}$'s the same value: $\boldsymbol{1/R}$. The second way is to set all $\boldsymbol{\beta_r}$ to 1 and calculate the $\boldsymbol{\gamma_r}$'s according to equation 4.5. In order to satisfy (4.3), the resulting scores $\boldsymbol{SC_n}$ can be divided by the total sum of the scale factors, such that new scaled $\boldsymbol{SC_n}$ values are obtained:

$$\boldsymbol{SC_{n,new}} = \frac{\boldsymbol{SC_{n,old}}}{\sum_r \gamma_r} \tag{4.6}$$

The third way to determine the weight factors is to apply a procedure which optimizes the $\boldsymbol{\beta}$'s, such as linear regression. The drawback of this approach is that some weight factors can become negative. This can be prevented by using a linear regression with positive coefficients.The final way to set the $\boldsymbol{\beta}$ weights is based on domain expertise. A domain expert can be someone with expert knowledge of or intensive experience with the algorithms, the (kind of) data set or, ideally, both. If there are indications that one algorithm performs better on a data set than others, the weight factor $\boldsymbol{\beta}$ for this algorithm can be increased accordingly.

4.3.7 Test structure

We assume that we have the availability of a train and a score set. The first is used for training and a first validation, the second for a out-of-sample test. In order to raise the generalizing power of the model, the train set is randomly divided into three sets, such that the overall percentage of respondents was kept. Two sets are, in turn, used for training and the third as a first validation. Finally, a fourth model is built by adding all three sets. By doing so, four different models are used, which on turn can be treated as one model by assigning each model weights. In AD2 the procedure is drawn. The models mentioned in this figure represent stand-alone algorithms or the configuration of AD1, the combination of several algorithms.
The performance measures on the first validation sets ("val" 1 to 3) are not numerically given in this thesis. The reason follows in section 5.1. They can, however, be a first indication which model trained on the cross sets performs best. The performances that we will report are labeled "score" 1 to 3, the performance of the three models based on the cross sets, "score" 4, the performance of the model based on the entire train set and "avg score", the three cross models added up in the same way we combine different algorithms.

8. Interpretation of patterns found, and possibly return to any of the steps 1-7.

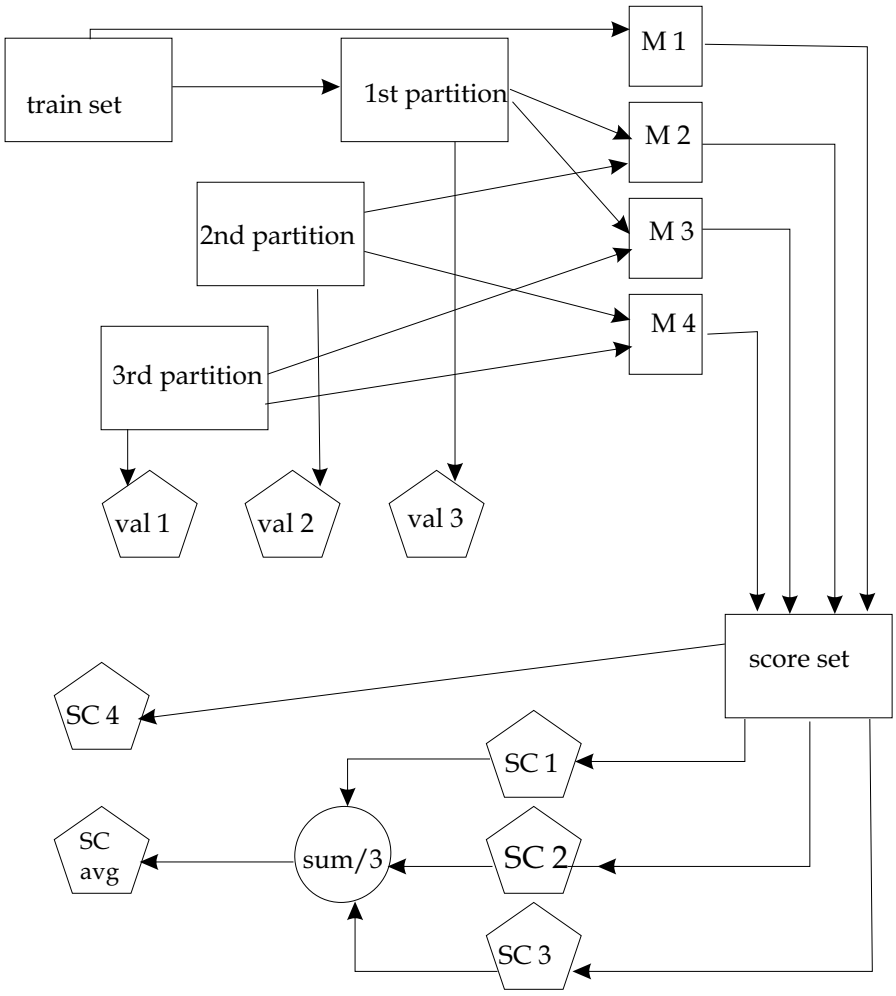

Figure 4.2: Flow chart of data sets and performance measures

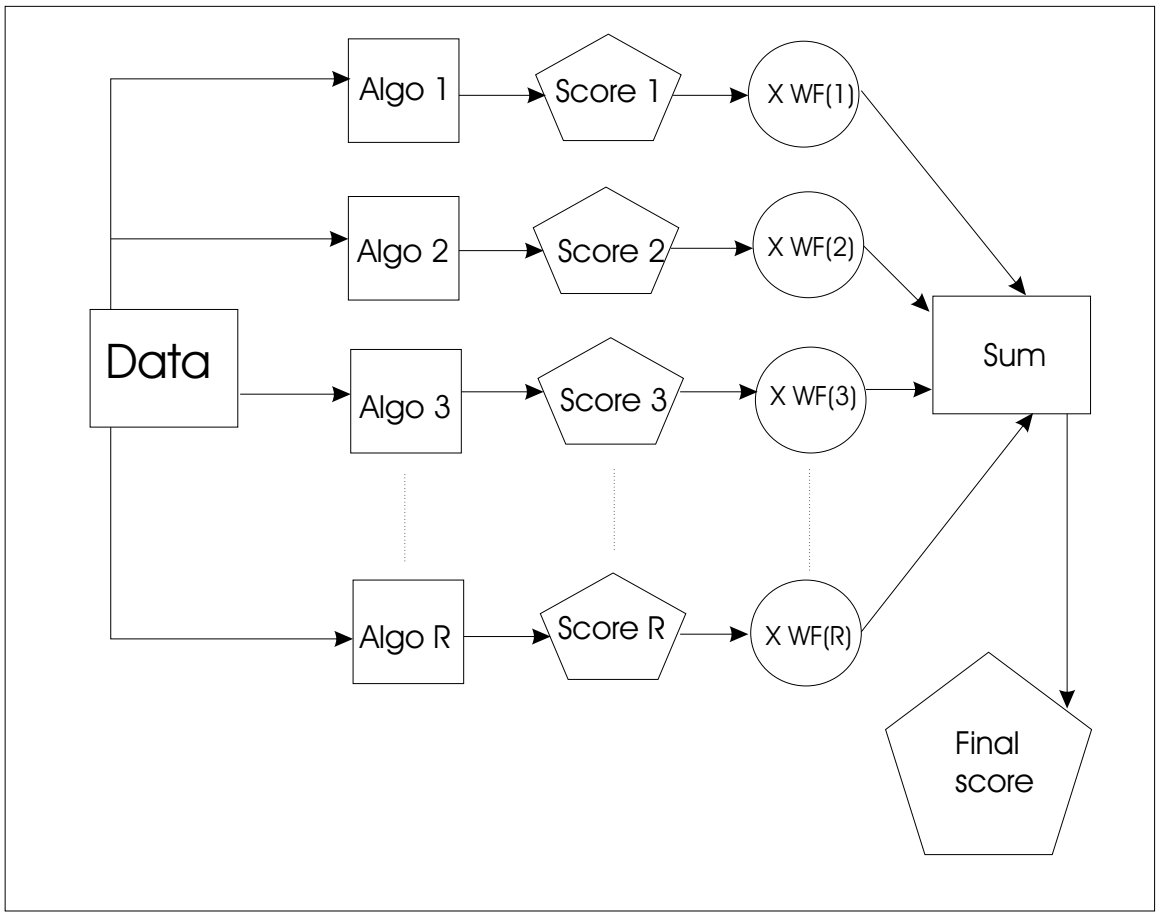

Figure 4.3: Methodology

The interpretation of the eventual feature set on the basis of the original CoIL assignment is part of chapter 5.

9. Consolidating the discovered knowledge: incorporating this knowledge in another system for further action.

   The systems in which the deducted knowledge is incorporated are on the one hand the unseen data of the third subset in the 3 way cross validation. On the other hand all final configurations are used to predict likelihood of response of the individuals in the "score" set of the CoIL challenge. Results on this set are compared to previous obtained results by the contestants of the CoIL challenge and can be found in chapter 5.

Chapter 5

Results

In this chapter the results of the methodology presented in the previous chapter are reported for the data set which was also introduced and investigated in chapter 4. As mentioned earlier, the analysis of direct marketing campaigns entails two stages. First, *feature selection* must determine the variables that are relevant for the specific target selection problem. second, the rules for selecting the customers should be determined, given the relevant features. This stages is referred to as *client scoring*. In the KDD cycle described in chapter 2, external feature selection is part of step 4, and the internal feature selection occurs as a loop over step 4 to 8.
This chapter is divided into three sections: in section 5.1 the results for the optimal configuration for each algorithm used stand-alone are presented and discussed, in section 5.2 the same goes for the combinations and finally, in section 5.3, the results of the previous two sections are compared to the results of the best CoIL submissions.

5.1 Stand-alone algorithms

Linear regression

If we use all features and use linear regression to score all clients, we obtain the gain charts drawn in figure 5.1. The corresponding number of correctly predicted respondents in the first ordered 20 % of all clients can be found in table 5.1.
The second column (labeled "VAL") in this table gives the values of the respon-

set	VAL	SCORE
ran	23	48
1st	61	121
2nd	57	110
3rd	57	115
tot	200	118
avg	-	115

Table 5.1: CPM values for linear regression with all features

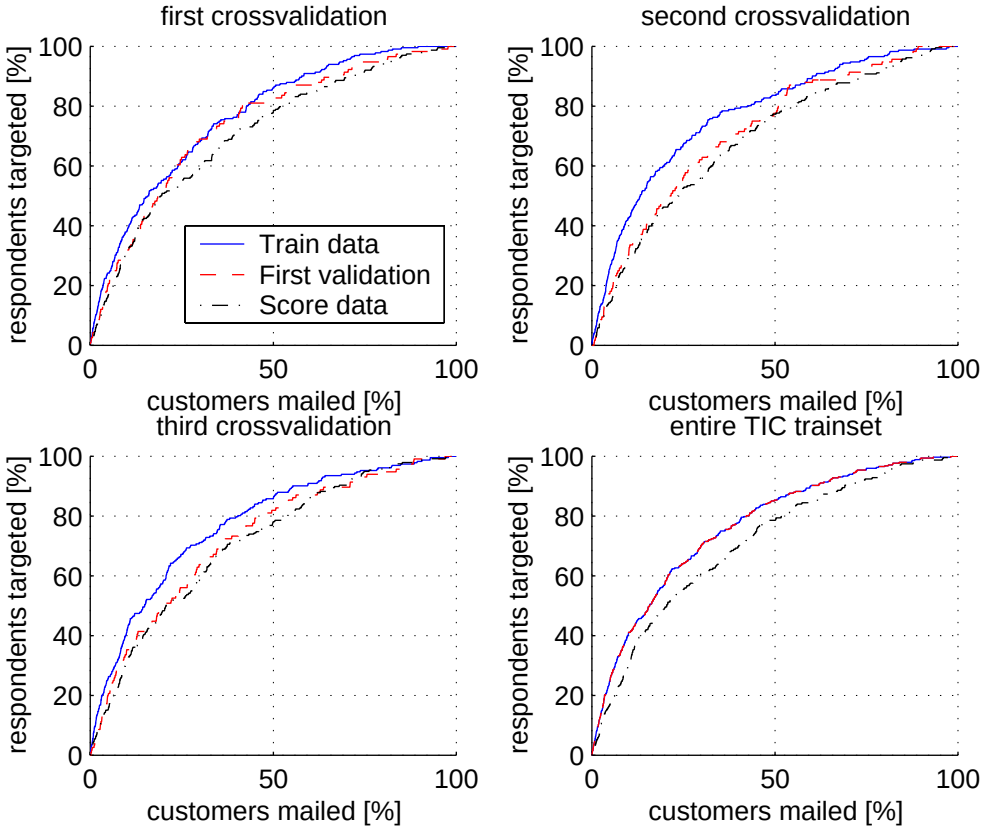

Figure 5.1: Gain chart for linear regression using all features

dents in that part of the train set which has not been used to train the model. The exception is the model which has been trained and validated on the whole train set (tot). The second row gives the number of respondents which would have been correctly identified by random mailing. The other rows represent the models based on the three cross validation sets, the total train set and a model based on all three cross sets (avg). This fourth model is trained and (at first) validated on the entire train set. In order to calculate the gain, the number of the random mailing has to be multiplied by three. The highest score on the score set is 121, based on the model trained on the first cross validation. This is a gain of 252 % compared to random mailing. Despite of the fact that this result is a good one (see section 5.2), how can we predict that the model based on the first cross validation is the best? One way is pick the model that scores best on the "VAL" sets. This approach has a few disadvantages: the first problem is how the fourth model which is based on the entire train set, has to be compared to the other ones. We could divided the "VAL" value by three, but it is obvious that this value is not representative because the validation samples are the same as the train samples. Another option is to discard the fourth model, but as we will see, some models based on the entire train set prove to be the best ones. A third way to overcome this problem is to adopt a new client score based on the client scores given by the models based on the three cross validation sets in the same way as combining the models is suggested in section 4.3.6. The weights are set to 1/3. The resulting number (avg) of respondents for the score set is equal to 115. We believe this is the best way to score clients in real life, although there will be some exceptions. To test this hypothesis, we shall give all results in the same format as table 5.1 without giving the "VAL" values because they do not help in deciding whether to rely on the "tot" or "avg" model.

External feature selection based on correlation is mentioned in section 3.1. If we adopt the rule (equation 3.8) that only includes those features with a absolute correlation twice or large then the mean absolute correlation to the dependent attribute of all independent features, the feature set obtained is given in table 5.2 for the tic data set. We shall refer to this feature set as correlation fea-

Table 5.2: Features with large correlation to Caravan policies owners

No	Name	No	Name
16	High education level	59	Contribution fire policies
18	Lower level education	61	Contribution boat policies
42	Average income	65	Number of pr. 3rd party insurance
43	Purchasing power class	68	Number of car policies
44	Contribution pr. 3rd party	82	Number of bicycle policies
47	Contribution car policies		

tures A. If we adopt a second rule which excludes features that have a large correlation (¿ 90 %) to each other we obtain the same feature set as listed in table 5.2 without features 61, 65 and 68. This set is named correlation features B. Linear regression on the correlation feature subset A results in CPM's (see section 4.2) which are given in table 5.3. In this table the different values are given for all five different models. These values will act as benchmark for the other configurations.

1st cross validation	111
2nd cross validation	112
3rd cross validation	112
entire tic data set	110
avg cross validation	110

Table 5.3: CPM values for linear regression with correlation features

Logit

No	name	98 %	90 %	85 %	80 %	75 %
2	Number of houses					x
4	Average age		x	x	x	x
10	Married				x	x
22	Middle management				x	x
28	Social class C			x	x	x
32	1 car				x	x
41	Income ¿ 123K			x	x	x
47	Contribution car	x	x	x	x	x
55	Contribution life				x	x
59	Contribution fire	x	x	x	x	x
76	Number of life					x
80	Number of fire	x	x	x	x	x
82	Number of boat			x	x	x

Table 5.4: Features selected by logit as significant

The features listed in table 5.4 can be seen as the most important features selected by logit. For instance, 85 of the 100 selections all selected the 7 features (tabled in the 85 % column): 4, 28, 41, 47, 59, 80, 82. Of the 100 selections, 2 just gave the intercept as single important predictor, these selections resulted in an especially bad (but not bad enough to activate the escape flag [1]) conditioned data set, which caused this difference. This is the reason that 98 % features are listed instead of the 100 % features, which is empty. One way or the other, the three most important features selected by logit are: 47, 59 and 80.

If only the seven features significant at the 85 % level (table 5.4) are used for the input (scaled), this results in the gain charts shown in figure 5.2. The choice for the best 7 features was made after inspecting table 5.4 with the combination of a high percentage and just a few predictors in mind. The resulting CPM's are listed in table 5.5. The CPM's of logit on the correlation feature subset of table 5.2 are also listed. These values are slightly better, although the improvement hold for all train sets.

[1] Some selections are extremely bad conditioned, in these cases the "escape flag" is activated; the current run is terminated and a new initialization, e.g. a new selection is used

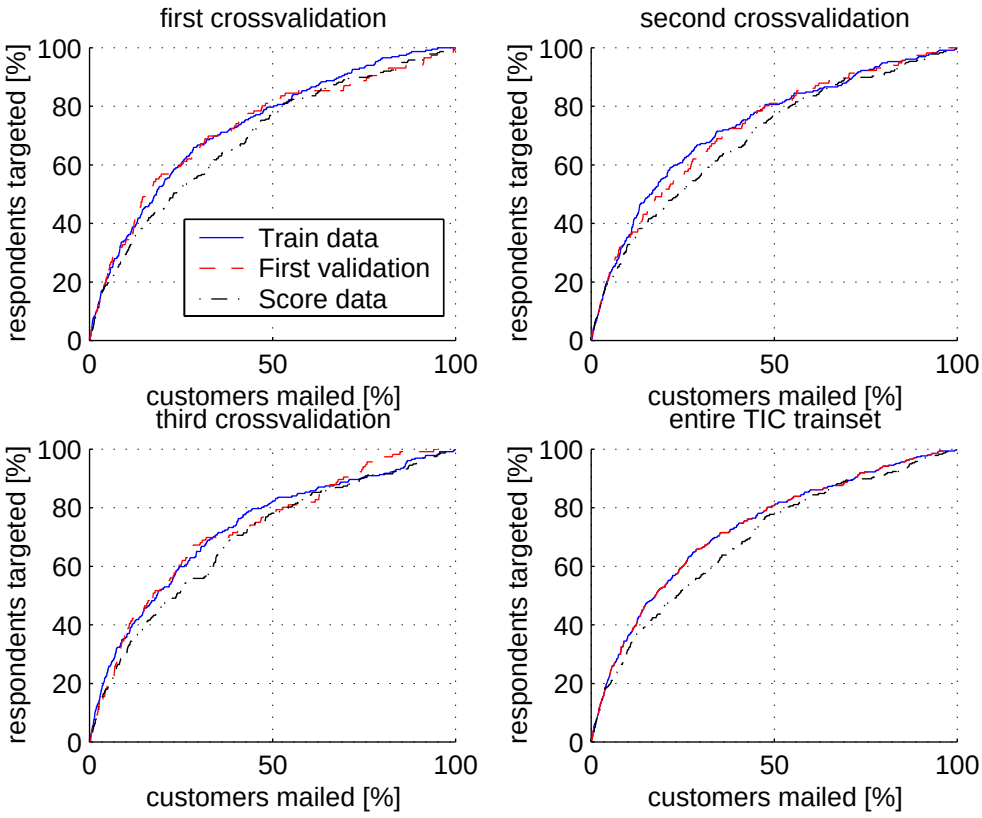

Figure 5.2: Gain chart for logit with 7 best logit features

CPM	log feat	corr feat A	corr feat B
1st cross validation	110	107	107
2nd cross validation	108	111	111
3rd cross validation	104	112	112
entire tic data set	103	109	111
avg cross validation	103	109	111

Table 5.5: CPM values for logit

Chaid

On the data set used in this thesis, Chaid did not performed well. To support this statement the resulting gain chart is drawn in figure 5.3. This gain chart

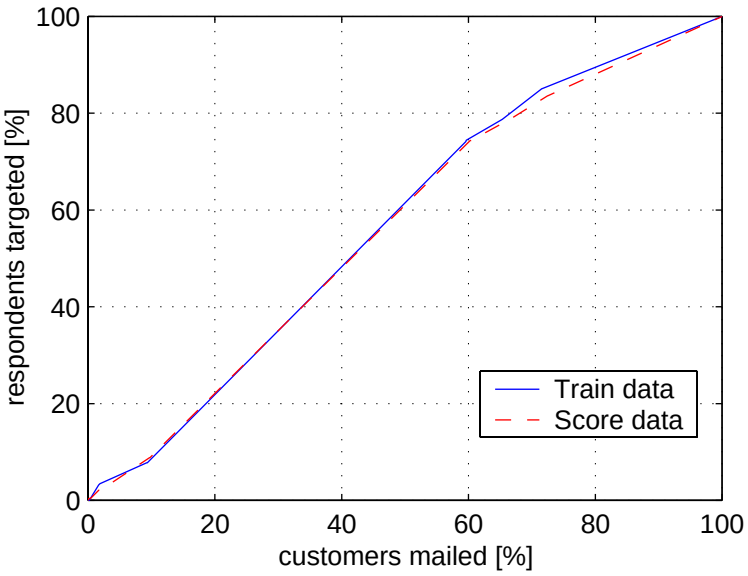

Figure 5.3: Gain chart for Chaid

differs from the other ones. The model is build on the entire train set. The only reason why this gain chart is presented is to show that Chaid does not do well on this data set. It can, however, be used to construct initial feature subsets.

Neural networks

Two neural networks are build; the first has six neurons in the hidden layer and is build on the correlation features set A. The second NN has two neurons and the input is the correlation subset B. The number of neurons in the hidden layer is determined using a trail and error approach. The CPM values are listed in table 5.6.

CPM	NN I	NN II
1st cross validation	103	110
2nd cross validation	98	109
3rd cross validation	95	111
entire tic data set	121	114
avg cross validation	108	114

Table 5.6: CPM values for the NNs

Fuzzy modeling

The fuzzy modeling technique is initialized with 5 initial cluster and set at depth 11.

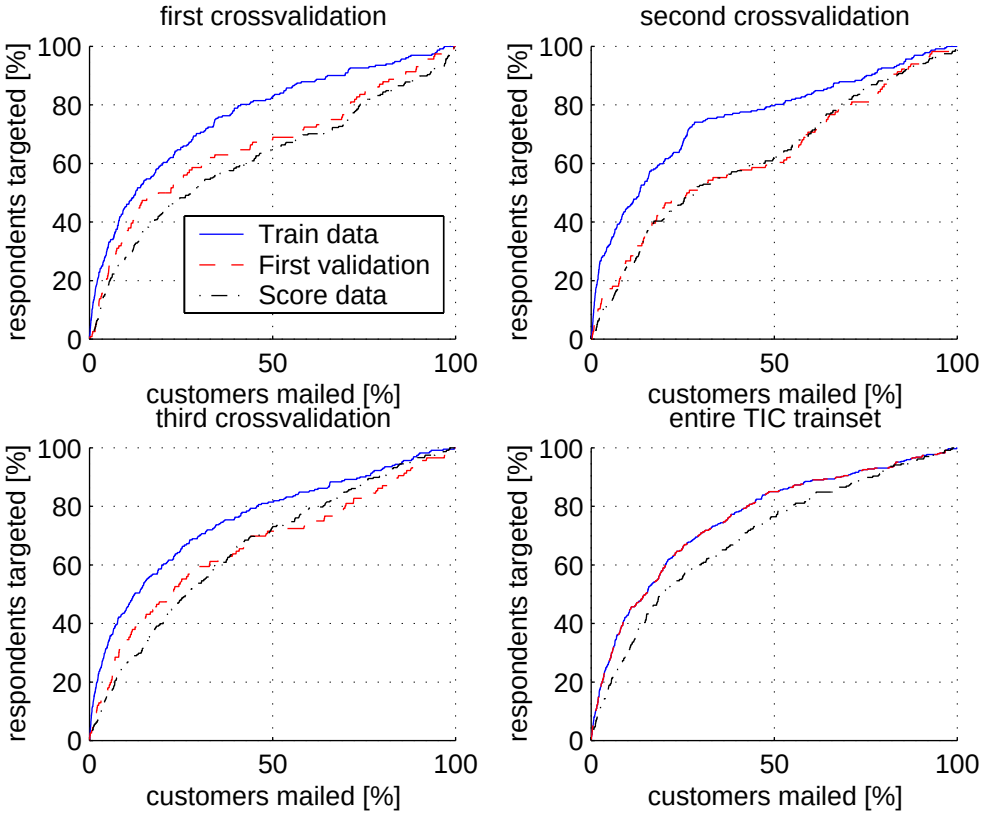

Figure 5.4: Gain chart for neural network I

1st cross validation	112
2nd cross validation	99
3rd cross validation	110
entire tic data set	98
avg cross validation	117

Table 5.7: CPM values for the fuzzy modeling technique

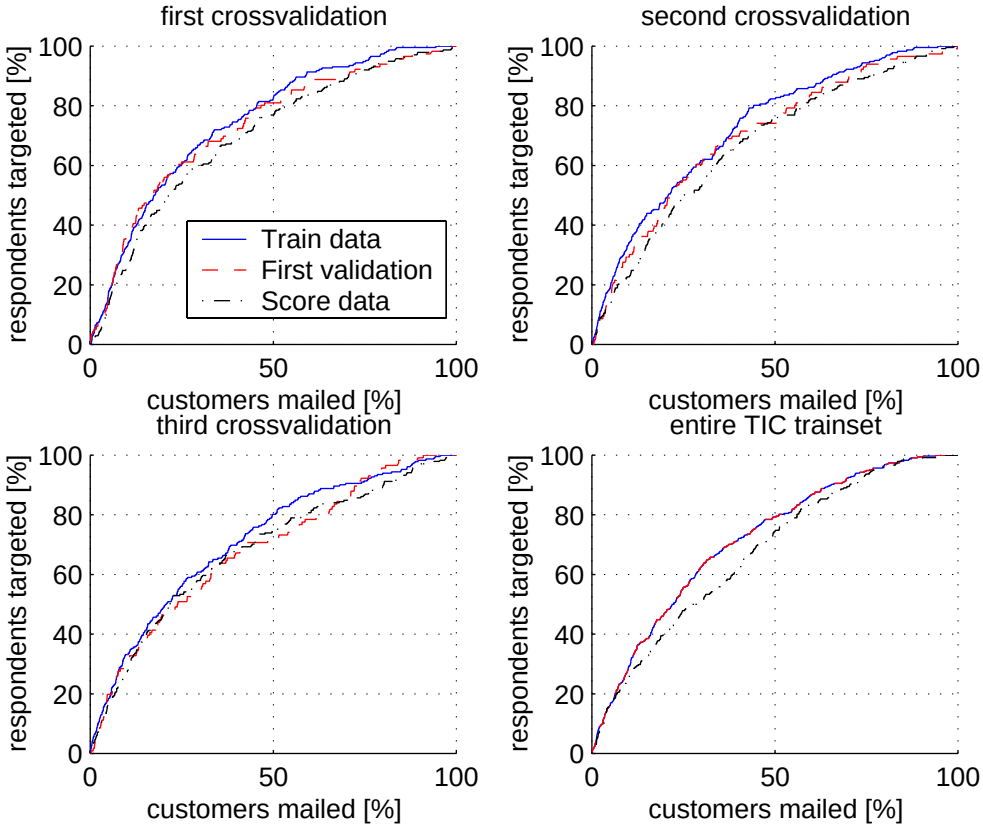

Figure 5.5: Gain chart for fuzzy model

5.2 Combing the algorithms

Two examples are given to explain the combinations as mentioned in section 4.3.6.

Example 1

In this combination four algorithms are used. The first algorithm is linear regression applied on all 85 variables (no feature selection). The second one is logistic regression, on a feature subset ($\boldsymbol{k(2) = 8}$) also build by logit using a t-test to test whether a coefficient significantly differed from zero. The third one is a feed-forward back-propagation neural network with 2 neurons in the hidden layer deployed on a feature subset based on the degree of absolute correlation to the dependent variable where interaction effects between variables are accounted for ($\boldsymbol{k(3) = 8}$), trained in 500 epochs. The fourth and final one is given by a fuzzy modeling technique [7] with 5 initial clusters for each variable ($\boldsymbol{k(4) = 8}$). The different values for the weight factors are given in table 5.8. In the first

	equal	scaled	optimized	dom exp		all
β_1	$0.25/\gamma_1$	1	2.1061	1.5	γ_1	0.2131
β_2	$0.25/\gamma_2$	1	0	0.5	γ_2	0.2038
β_3	$0.25/\gamma_3$	1	2.3412	1	γ_3	0.2066
β_4	$0.25/\gamma_4$	1	0.1793	1	γ_4	0.3765

Table 5.8: Weight factors for example 1

column the four different sets of weight factors are listed, the γ's are the same for all listed β values. Note that the resulting α values are scaled except the "dom exp" -β values. The resulting α's can be scaled by applying equation 4.5, where the γ_r is substituted by α_r. In this way, the weights set by the domain expert (labeled "dom exp") are easier to explain. They are chosen given the following observations. First of all, the logit model seems to be the weakest link in this combination. The procedure to optimize the weight factor even excludes logit. This seems to be a bit exaggerated, but a relative low value is desirable. Furthermore, the weight factor for the linear regression is set a bit higher because it is built on all features, and subsequently is expected to represent more information. The resulting gain charts for the configuration with the weight factors determined by a domain expert are drawn in figure 5.6.
The numerical performance measure used in the CoIL Challenge was the number of respondents found in the first 20 % of the customers sorted by the $\boldsymbol{SC_n}$ scores. The total number of respondents in the score set equals 238. A random mailing (no model used) results in a performance measure of circa 48 (47.6). So, every value greater than this one is a gain to random mailing and the maximum number is 238. The values for the four algorithms used stand-alone and the combinations with the four sets of weight factors can be found in table 5.9.

Example 2

The second combination exists of two algorithms: the same logit configuration as in example 1 and a neural network with 6 neurons in the hidden layer on a feature subset based on correlation to the dependent variable. This second feature

	lin	log	nn	fuz	equal	scaled	optim	dom exp
score 1	121	110	110	113	118	118	119	123
score 2	110	108	109	90	114	115	113	114
score 3	115	104	111	111	115	113	117	116
score 4	118	103	114	108	114	116	117	120
avg score	-	-	-	-	116	115	119	119

Table 5.9: Numerical performance measures example 1

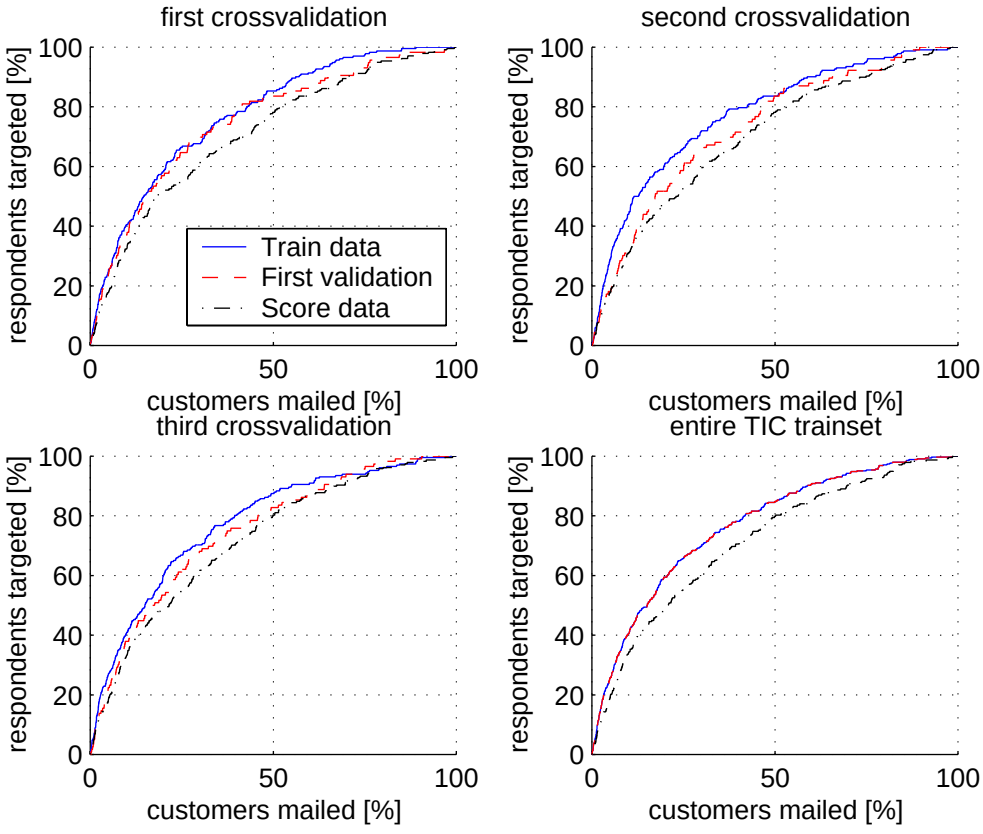

Figure 5.6: Gain charts example 1

subset contains 11 features. The different values for the weight factors are listed in table 5.10. The dissentient choice of the domain expertise weights can

	equal	scaled	optimized	dom exp		all
β_1	$0.5/\gamma_1$	1	1.3897	0.5	γ_1	0.6456
β_2	$0.5/\gamma_2$	1	0.2929	1.5	γ_2	0.3544

Table 5.10: Weight factors for example 2

be explained by the fact that this particular neural network performs really well on the data and logit does less, as can be seen in example 1. The combination works very well as can be observed by looking at the numerical performances which are listed in table 5.11 and the gain charts for the domain expertise configuration which are drawn in figure 5.7.

	log	nn	equal	scaled	optim	dom exp
score 1	110	103	105	107	115	104
score 2	108	98	106	106	110	104
score 3	114	95	106	108	114	106
score 4	103	121	122	115	123	125
avg score	-	-	108	108	115	108

Table 5.11: Numerical performance measures example 2

5.3 Comparison to previous results and discussion

In this section, the results of the previous section are discussed and compared to the results of the contestants of the CoIL challenge 2000. From table 5.9, we can see that the model based on the entire train set correctly scores 120 in the first 20 % ranked prospects in the out-of-sample test. The model on the first cross validation does even better: 123 respondents. In figure 5.8 the results of the 43 participants are visualized. The best score is 121. Even the other models do very well (in the best 5% of the CoIL participants).
In example 2 even a higher value is reached: 125 for the combination on the entire train set. Although the configuration with the neural network scored 121 stand-alone, the combination with logit adds another 4 successfully predicted respondents.
The models build on the three cross validation sets ("avg score") do not have a better result than the combinations on the entire train set.
Although some configurations used in the two examples do very well applied stand-alone, there is a significant gain in using approach of combining target selection algorithms.
Every configuration, hence algorithm, introduces its own systematic errors because the assumptions underlying the algorithm do not hold to their full extend in the real world. Optimizing the weight factors using linear regression treats each systematic error as a random error in the addition sum (equation 4.2). Following this observation, optimizing the weights is more justified when a larger

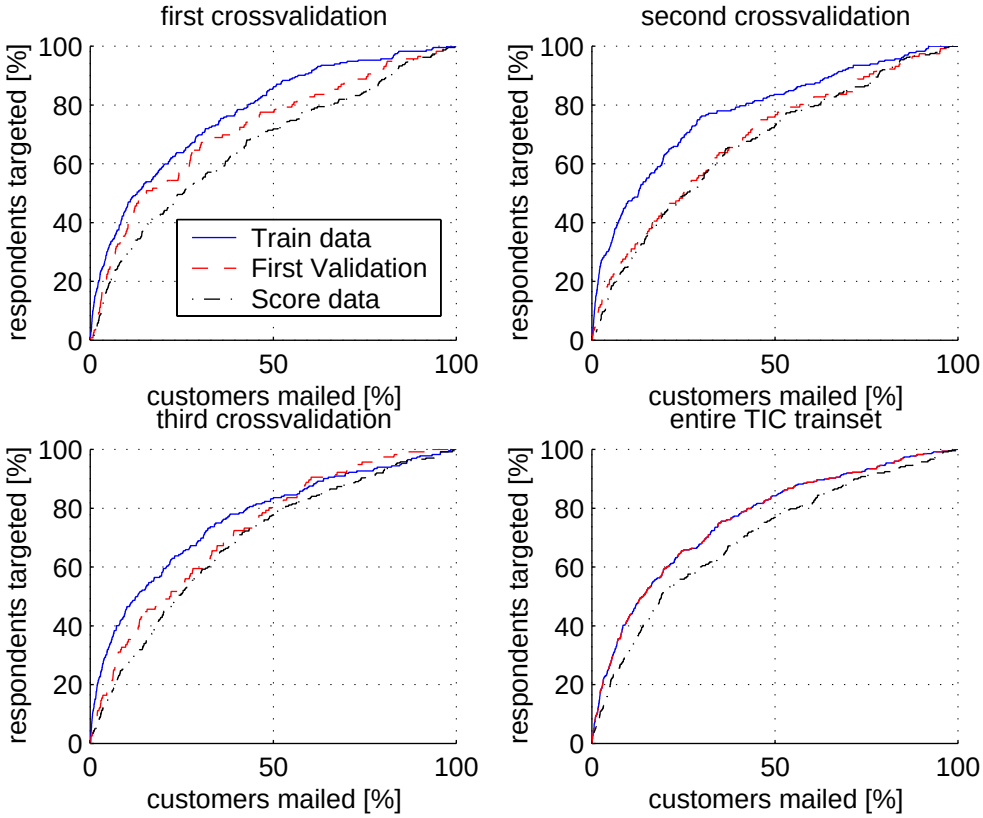

Figure 5.7: Gain charts example 2

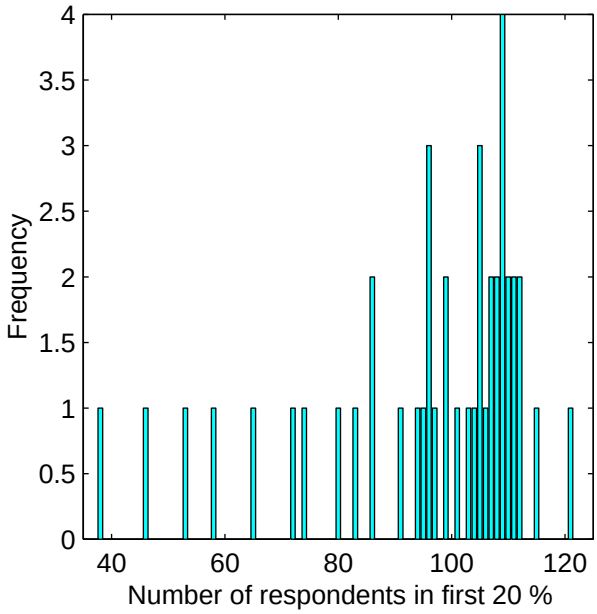

Figure 5.8: Results participants of CoIL 2000

number of algorithms are combined. Systematic errors imply that all values in a set are shifted in the same direction in the same amount - in unison. This is in contrast to random errors where each value fluctuates independently of the others. So, when it comes to optimizing the weight factors, a combination of more than a few algorithms with *different* assumptions is the best option.
The results subscribe this statement: the combinations of the models build on the three cross sets ("avg score") do not outperform the combinations on the entire train set, because the same assumptions hold. On the other hand , the β weights of the optimized set in example 1 are closer to the weights set by the domain expert than in example 2. The domain expert obtained the best results, so in example 1 the weights are nearer to his weights, because more algorithms, hence different assumptions are used.

Chapter 6

Conclusions and Recommendations

The conclusions of this thesis project are given in this section, followed by some recommendations for future research.

Conclusions

Despite the fact that many algorithms are developed for the purpose of target selection, no universal algorithm exist. The strength of the approach presented in this thesis is that the structure and specific characteristics of each feature subset are maintained and scored individually. If proper action is taken to prevent over-training in building the algorithms, combining them will not introduce any kind of over-training. The quality of the train samples are equally important in preventing over-training. Spurious relations can be found, especially in case of a large number of algorithms in a combination.
By adequately combining the scores, results are achieved out-performing all participants in the CoIL challenge 2000. The best results are obtained by using the weights chosen by a domain expert. If such a person is not available, the best way seems to be optimizing the weights using a linear regression method with positive coefficients. A fairly large number of algorithms with different assumptions is needed for adequately optimized weights.

Different kinds of variables do not need special treatments. Except for the nominal variables which have to be transformed adding new non-existing relation between the variables values.
NNs need scaling to operate and feature selection by logit requires scaling. Linear regression, Chaid and the fuzzy modeling technique do not need scaling of the inputs.

Experiments on the data set pointed out that the NN performs best, followed closely by linear regression. The fuzzy modeling technique and logit did a bit less. Chaid is found to fail on the data set investigated.
Several combinations of the algorithms are proposed, and the combination of logit and NNs does outperform every contestant of the CoIL challenge. Although the extra gain is marginal in the high cost – high impact environment

every gain increases profit.

Recommendations for further research

Further research has to be done to find out where the limits of combining algorithms are at. To put in other words, how sensitive this approach is to adding a large number of different combinations. Another question is how cross validation could better be integrated in this technique.

Some general recommendations are:

1. The format in which the data set is recorded must be improved. The people responsible for the data collection need to contact the ones responsible for target selection. This could, for instance, prevent quantization errors.

2. Instead of optimizing the algorithms itself, they could be optimized on the gain chart. For instance, the NN is not updated on the known output, but by optimizing the gains on the gain chart.

3. Interaction effects can be added as extra features. In this way, more features are present, but perhaps better modelling capabilities introduced.

Appendix A

Tables of the features and datasets

In this appendix the feature labels with a short describtion are listed in tables A.1 and A.2.
Next to the feature labels, the different categories, as named in the 'domain'–column (see tables A.1 and A.2, are given. The first category is L0, the customer sub type categories, see table A.3. This category consist of nominal values, 41 in total. The only other nominal category is L2, as listed in table A.5, and has eleven different values. As stated in section 4.3.1, the features who use these two categories have a strong correlation, where L0 is merely more detailed. Category type L1 (table A.4) contains categorical age values. Category type L3, percentage and L4 the annual contribution to a certain insurance in Dutch guilders (see tables A.6 and A.7).

Table A.1: Listing of the attributes 1 to 43

Nr	Name	Description	Domain
1	MOSTYPE	Customer subtype	see L0
2	MAANTHUI	Number of houses	1 to 10
3	MGEMOMV	Avg size household	1 to 6
4	MGEMLEEF	Avg age	see L1
5	MOSHOOFD	Customer main type	see L2
6	MGODRK	Roman catholic	see L3
7	MGODPR	Protestant	
8	MGODOV	Other religion	
9	MGODGE	No religion	
10	MRELGE	Married	
11	MRELSA	Living together	
12	MRELOV	Other relation	
13	MFALLEEN	Singles	
14	MFGEKIND	Household without children	
15	MFWEKIND	Household with children	
16	MOPLHOOG	High level education	
17	MOPLMIDD	Medium level education	
18	MOPLLAAG	Lower level education	
19	MBERHOOG	High status	
20	MBERZELF	Entrepreneur	
21	MBERBOER	Farmer	
22	MBERMIDD	Middle management	
23	MBERARBG	Skilled labourers	
24	MBERARBO	Unskilled labourers	
25	MSKA	Social class A	
26	MSKB1	Social class B1	
27	MSKB2	Social class B2	
28	MSKC	Social class C	
29	MSKD	Social class D	
30	MHHUUR	Rented house	
31	MHKOOP	Home owners	
32	MAUT1	1 car	
33	MAUT2	2 cars	
34	MAUT0	No car	
35	MZFONDS	National Health Service	
36	MZPART	Private health insurance	
37	MINKM30	Income ¡ 30.000	
38	MINK3045	Income 30 to 45.000	
39	MINK4575	Income 45 to 75.000	
40	MINK7512	Income 75 to 122.000	
41	MINK123M	Income ¿123.000	
42	MINKGEM	Average income	
43	MKOOPKLA	Purchasing power class	

Table A.2: Listing of the attributes 44 to 86

Nr	Name	Description	Domain
44	PWAPART	Contribution private third party insurance	see L4
45	PWABEDR	Contribution third party insurance (firms)	
46	PWALAND	Contribution third party insurane (agriculture)	
47	PPERSAUT	Contribution car policies	
48	PBESAUT	Contribution delivery van policies	
49	PMOTSCO	Contribution motorcycle/scooter policies	
50	PVRAAUT	Contribution lorry policies	
51	PAANHANG	Contribution trailer policies	
52	PTRACTOR	Contribution tractor policies	
53	PWERKT	Contribution agricultural machines policies	
54	PBROM	Contribution moped policies	
55	PLEVEN	Contribution life insurances	
56	PPERSONG	Contribution private accident insurance policies	
57	PGEZONG	Contribution family accidents insurance policies	
58	PWAOREG	Contribution disability insurance policies	
59	PBRAND	Contribution fire policies	
60	PZEILPL	Contribution surfboard policies	
61	PPLEZIER	Contribution boat policies	
62	PFIETS	Contribution bicycle policies	
63	PINBOED	Contribution property insurance policies	
64	PBYSTAND	Contribution social security insurance policies	
65	AWAPART	Number of private third party insurance	1 to 12
66	AWABEDR	Number of third party insurance (firms)	
67	AWALAND	Number of third party insurane (agriculture)	
68	APERSAUT	Number of car policies	
69	ABESAUT	Number of delivery van policies	
70	AMOTSCO	Number of motorcycle/scooter policies	
71	AVRAAUT	Number of lorry policies	
72	AAANHANG	Number of trailer policies	
73	ATRACTOR	Number of tractor policies	
74	AWERKT	Number of agricultural machines policies	
75	ABROM	Number of moped policies	
76	ALEVEN	Number of life insurances	
77	APERSONG	Number of private accident insurance policies	
78	AGEZONG	Number of family accidents insurance policies	
79	AWAOREG	Number of disability insurance policies	
80	ABRAND	Number of fire policies	
81	AZEILPL	Number of surfboard policies	
82	APLEZIER	Number of boat policies	
83	AFIETS	Number of bicycle policies	
84	AINBOED	Number of property insurance policies	
85	ABYSTAND	Number of social security insurance policies	
86	CARAVAN	Number of mobile home policies	0 or 1

Table A.3: Category L0

Value	Label
1	High Income, expensive child
2	Very Important Provincials
3	High status seniors
4	Affluent senior apartments
5	Mixed seniors
6	Career and childcare
7	Dinki's (double income no kids)
8	Middle class families
9	Modern, complete families
10	Stable family
11	Family starters
12	Affluent young families
13	Young all american family
14	Junior cosmopolitan
15	Senior cosmopolitans
16	Students in apartments
17	Fresh masters in the city
18	Single youth
19	Suburban youth
20	Etnically diverse
21	Young urban have-nots
22	Mixed apartment dwellers
23	Young and rising
24	Young, low educated
25	Young seniors in the city
26	Own home elderly
27	Seniors in apartments
28	Residential elderly
29	Porchless seniors: no front yard
30	Religious elderly singles
31	Low income catholics
32	Mixed seniors
33	Lower class large families
34	Large family, employed child
35	Village families
36	Couples with teens 'Married with children'
37	Mixed small town dwellers
38	Traditional families
39	Large religous families
40	Large family farms
41	Mixed rurals

Table A.4: Category L1

Value	Age
1	20-30 years
2	30-40 years
3	40-50 years
4	50-60 years
5	60-70 years
6	70-80 years

Table A.5: Category L2

Value	Label
1	Successful hedonists
2	Driven Growers
3	Average Family
4	Career Loners
5	Living well
6	Cruising Seniors
7	Retired and Religeous
8	Family with grown ups
9	Conservative families
10	Farmers

Table A.6: Category L3

Value	Percentage
0	0
1	1 to 10
2	11 to 23
3	24 to 36
4	37 to 49
5	50 to 62
6	63 to 75
7	76 to 88
8	89 to 99
9	100

Table A.7: Category L4

Value	Contribution
0	f 0
1	f 1 to 49
2	f 50 to 99
3	f 100 to 199
4	f 200 to 499
5	f 500 to 999
6	f 1000 to 4999
7	f 5000 to 9999
8	f 10.000 to 19.999
9	f 20.000 and beyond

Bibliography

[1] K.A. Forcht and K. Cochran, "Using Data mining and Dataware-housing techniques", *Industrial Management and Data Systems*, 1999

[2] J.R. Bult, *Target selection for direct marketing*,PhD thesis, Rijks Universiteit Groningen, The Netherlands, 1993

[3] J. Donovan, *D.I.Y. direct marketing*, Kogan Page Ltd. London, 2000

[4] V.G. Morwitz and D.C. Schmittlein, "Testing new direct marketing offerings: The interplay of management judgment and statistical models", *Management Science* vol. 44, No. 5, pp. 610-628, May 1998

[5] R. Webber, "Intelligent systems for market segmentation and local market planning", *Intelligent systems for finance and business*, John Wiley and Sons Ltd England, 1995.

[6] W.H. Greene, *Econometric Analysis*, Macmillan Publishing Company New York, 1993

[7] M. Setnes and U. Kaymak, "Fuzzy modeling of client preference from large data sets: an application to target selection in direct marketing", *IEEE Transactions on Fuzzy Systems*, vol. 9, No. 1, 2001

[8] G.V. Kass, "An exploratoty technique for investigating large quantaties of categorical data *Applied Statistics* vol. 29, No. 2, pp. 119-127, 1980

[9] M. Ben-Akiva and S.R. Lerman, *Discrete choice analysis: Theory and Application to Travel Demand*, MIT Press England, 1985

[10] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "Knowledge discovery and Data mining: towards an unifying framework", *Proceedings of KDD-96: International Conference on Knowledge Discovery and Data mining*, pp. 82-88, 1996

[11] J. Kleinberg, C. Papadimitriou and P. Raghavan, "A microeconomic view of data mining", *Data mining and knowledge discovery 2*, pp 311-324, 1998

[12] R. Babuska, *Introduction to Neural Networks*, Delft University of Technology, 2000

[13] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Wadsworth: Belmont USA, 1984

[14] D.M. Hawkins and G.V. Kass, "Automatic Interaction Detection", *Topics in Applied Multivariate Analysis*, pp. 267-302, Cambridge Univ Press: Cambridge, 1982

[15] E.B. Moser, "Chaid-like Classification Trees", *Multivariate Statistics, course material*, www.stat.lsu.edu.faculty/moser/exst7037/exst7037.html, 2000

[16] J. Magidson, "Improved statistical techniques for response modeling-Progression beyond regression", *Journal of Direct Marketing* vol. 2 No. 4 pp 6-18, 1988

[17] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal", *Journal of the American Statistical Association*, vol. 58 pp. 415-434, 1963

[18] J. A. Sonquist, E. L. Baker and J. N. Morgan, *Searching for Structure*, Institute for Social Research, University of Michigan, Ann Arbor, 1973

[19] D. Shepard, *The New Direct Marketing*, $\mathbf{2^{nd}}$ edition, Irwin Professional Marketing, 1995

[20] S. J. Long, *Regression Models for Categorical and Limited Dependent Variables*, Sage Publications, 1997.

[21] J. Zahavi and N. Levin, "Applying neural computing to target marketing", *Journal of Direct Marketing*, vol. 11 No. 4 pp. 77-93, 1997

[22] D. J. Finney, *Probit Analysis*, Cambridge University Press, UK, 1964

[23] N. Draper and H. Smith, *Applied regression analysis*, New York: John Wiley & Sons, 1966 (Revised ed., 1981)

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Ca, USA: Morgan Kaufmann Publishing, 1993

[25] D. G. Morrison, "On the Interpretation of Discriminant Analysis", *Journal of Marketing Research*, pp. 156-163, 1969

[26] N. Levin and J. Zahavi, *Predictive modeling using segmentation*, http://www.urbanscience.com/ Predictive_Modeling_Using_Segmentation.pdf, 1999

[27] R. Lowry,*Concepts and Applications of Inferential Statistics*, interactive statistics textbook, http://faculty.vassar.edu/ lowry/ch8pt3.html, 1999

[28] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm", *Proceedings International Conference on Machine Learning*, pp 148-156, Morgan Kaufmann, San Francisco, 1996

[29] Garson, *Logistic regression*, lecture notes, http://www2.chass.ncsu.edu/garson/pa765/logistic.htm, 2000

[30] S. Quigley, "Regression analysis", *SST Help guide* http://emlab.berkely.edu/sst/regression.html

[31] R. Potharst, U. Kaymak and W. Pijls, "Neural networks for target selection in direct marketing", *Neural networks in Business: techniques and application*, pp 89-110, Idea Group Publishing, London, 2002

[32] SPSS Inc., *SPSS Chaid for Windows 6.0*, Prentice -Hall, 1993