Predicting Customer Churn at a
Swedish CRM-system Company

by

David Buö and Magnus Kjellander

LIU-IDA/LITH-EX-A–14/028-SE

2014-06-23
Final thesis

Predicting Customer Churn at a Swedish CRM-system Company

by

David Buö and Magnus Kjellander

LIU-IDA/LITH-EX-A–14/028-SE

2014-06-23

Supervisor: Patrick Lambrix, IDA Linköping University
Martin Modéer, Lundalogik AB

Examiner: Jose M. Peña, IDA Linköping University
Abstract

This master thesis investigates if customer churn can be predicted at the Swedish CRM-system provider Lundalogik. Churn occurs when a customer leaves a company and is a relevant issue since it is cheaper to keep an existing customer than finding a new one. If churn can be predicted, the company can target their resources to those customers and hopefully keep them. Finding the customers likely to churn is done through mining Lundalogik’s customer database to find patterns that result in churn. Customer attributes considered relevant for the analysis are collected and prepared for mining. In addition, new attributes are created from information in the database and added to the analysis. The data mining was performed with Microsoft SQL Server Data Tools in iterations, where the data was prepared differently in each iteration.

The major conclusion from this thesis is that churn can be predicted at Lundalogik. The mining resulted in new insights regarding churn but also confirmed some of Lundalogik’s existing theories regarding churn. There are many factors that need to be taken into consideration when evaluating the results and which preparation gives the best results. To further improve the prediction there are some final recommendations, i.e. including invoice data, to Lundalogik of what can be done.
Acknowledgement

First, we would like to thank Lundalogik AB for giving us the opportunity to do this master thesis, especially our supervisor Martín Modéer for all the help but also all the colleagues at the office. Further we would like to thank our examiner Jose M. Peña and our supervisor Patrick Lambrix from the Department of Computer and Information Science at Linköping University. Lastly, a big thanks to our friends and families!

David Buö and Magnus Kjellander  
Stockholm, June 2014
## Contents

1 Introduction .............................................. 1
   1.1 Background ........................................ 1
   1.2 Purpose ............................................ 2
   1.3 Problem Definition ................................. 2
   1.4 Limitations ....................................... 3
   1.5 Structure of the Report ......................... 3

2 Research Method ...................................... 5
   2.1 Case Study ......................................... 5
      2.1.1 Case Study Design ............................ 6
      2.1.2 Data Collection ............................. 7
      2.1.3 Data Analysis ............................... 7
      2.1.4 Reporting ..................................... 8

3 Theoretical Background ............................... 9
   3.1 Customer Relationship Management ............. 9
   3.2 Customer Churn Management .................... 10
   3.3 Data Mining ..................................... 11
   3.4 CRISP-DM ........................................ 12
      3.4.1 Business Understanding ..................... 13
      3.4.2 Data Understanding ........................ 13
      3.4.3 Data Preparation ............................ 13
      3.4.4 Modeling .................................... 15
      3.4.5 Evaluation ................................. 15
      3.4.6 Deployment ................................ 16
   3.5 Data Quality ...................................... 16
      3.5.1 Accuracy .................................... 16
      3.5.2 Completeness ................................ 17
      3.5.3 Consistency ................................ 18
      3.5.4 Time Dimensions ............................ 18
   3.6 Classification .................................... 19
      3.6.1 Decision Tree Induction .................... 20
      3.6.2 Microsoft Decision Tree Algorithm ....... 21
   3.7 Evaluation Metrics ............................... 22
      3.7.1 Evaluating Classifier Performance ....... 22
Bibliography

Appendix

A Data at Lundalogik

B Attribute Completeness

C Iteration 1

D Iteration 2

E Iteration 3

F Iteration 4

G Iteration 5
Chapter 1

Introduction

1.1 Background

The amount of available data have increased the need to automatically find and uncover valuable information and to transform it into valuable knowledge. The evolution of database and information technology since the 1960s and the collection and storage of large amounts of data in various repositories have led to abundance of data and a need for powerful data analysis tools. This has been described as "a data rich but information poor situation". The data in the various repositories have been used by decision makers to get information driven by their domain knowledge and intuition because of the lack of tools to successfully automate the mining of valuable information. This is referred to as "data-tombs" in the literature. The repositories or data-tombs store huge amounts of data but are rarely visited because of the costly and time consuming operation of manually extracting valuable information. This has led to the birth of data mining which provides a toolset to discover valuable information from these data-tombs. (Han et al., 2012)

Data mining or "knowledge mining from data" have several different definitions in the literature, but we choose to use the definition used by Han et al. (2012): "Data mining is the process of discovering interesting patterns and knowledge from large amounts of data". There are many different techniques to discover interesting patterns e.g. logistic regression, decision trees, neural networks and cluster analysis. To determine which technique that is the most suitable depends solely on the situation and the form of the data to be mined. (Han et al., 2012)

There is an on-going trend to improve the process of data mining as a part of the overall business intelligence and customer relationship management (CRM) strategy across the organizations in various industries (Gunnarsson et al., 2007). By using data mining, organizations can find interesting patterns in their customer’s behaviour that would be almost impossible for a
human to manually detect. Whether it is patterns in customer purchase behaviour or information about which customer that has the highest possibility of leaving the company (churn) depends on the business objectives of the data mining process. Customer churn affects organizations negatively due to a loss of income and because of the cost for the organization to attract new customers (Risselada et al., 2010). In the field of data mining churn prediction has become well researched. Churn prediction is when data mining is implemented to make predictions about what customers that are about to leave. Gunnarsson et al. (2007) have performed data mining and churn prediction on a company in the newspaper industry, Coussement and De Bock (2013) performed churn prediction on data delivered by Bwin Interactive Entertainment which operates in the gambling industry. Similar studies can be found on organizations in various industries but there is lack of research in the field of churn prediction in the software industry.

This master thesis was conducted at Lundalogik AB in Stockholm for the Department of Computer and Information Science at Linköping University. Lundalogik AB is a CRM-system provider with head quarter in Lund, Sweden. Offices are also located in Gothenburg, Stockholm, Oslo, and Helsinki. Lundalogik AB are specialized in developing and selling CRM-system solutions for a wide spectrum of industries and have customers in all the Nordic countries.

1.2 Purpose

Today, Lundalogik know the churn rate among their customers. This rate is a statistical measure of the churning customers and no further analysis of it is done. If Lundalogik could predict which customers that are about to churn they believe that they can prevent customers from churning. The purpose of this thesis is to create prerequisites for Lundalogik to start predicting churn.

1.3 Problem Definition

The backbone of every data mining task is the data. A famous aphorism in the field of data mining is GIGO - garbage in, garbage out, which highlights the importance of appropriate data for the mining objectives (Pyle, 2009; Gunnarsson et al., 2007). For us to be able to give Lundalogik the prerequisites for predictive analytics we need to investigate which data to extract from the database and how to prepare it for the mining algorithm. With the prepared data as input to the predictive model we must be able to determine whether the preparations improve the result of the analysis or not. This emphasizes the main problems of this thesis:

- Which patterns for churning customers at Lundalogik can be identified?
1.4 Limitations

There are several different models for mining data. We will only use one for our analysis and compare its performance with different data preparations. The literature also recommends ensemble models where several models are combined to improve the result, which also is outside our scope. There are several available tools for data mining such as WEKA and Microsoft SQL Server Data Tools (SSDT). We will use Microsoft SSDT and not write our own algorithm. Further, there are more data that can be collected to get more information about Lundalogik’s customers but we have used the data from their own CRM-system as the foundation for this thesis.

1.5 Structure of the Report

The report is structured as follows. Chapter 2 describes the research method used for this thesis. Chapter 3 describes the theoretical background of CRM-systems, data mining, churn prediction, predictive analytics and data quality. Further, chapter 4 describes the case study conducted at Lundalogik and chapter 5 presents the results of the study. Finally, a discussion of the result is held in chapter 6 including recommendations for Lundalogik how to proceed with churn prediction and recommendations for further research in the field of churn prediction in the software industry.
Chapter 2

Research Method

This section describes the research methodology chosen for our empirical study.

2.1 Case Study

A case study is an empirical research method used when the researcher wants to "investigate contemporary phenomena in their context" (Runeson and Höst, 2009). Yin (2003) describes the situation for when a case study has an advantage as when how or why are asked about contemporary events, where the researcher has no or little control of the events. According to the work of Runeson and Höst (2009) case studies are a suitable research methodology for software engineering because it is a multidisciplinary area that includes other areas with the objective to increase knowledge about individuals and groups etc. (Runeson and Höst, 2009). The same areas are described by Yin (2003) as areas were case studies are commonly used.

The case study research process consist of the following five major steps according to Runeson and Höst (2009):

1. Case study design: Define objectives and plan the case study.

2. Preparation for data collection: Procedures and protocols for the data collection are defined.

3. Collecting evidence: The data collection is executed on the case being studied.

4. Analysis: The collected data is analyzed.

5. Reporting: The report communicates the findings from the analysis and also the quality of the study.
2.1. CASE STUDY

The steps might be iterated since the case study is a flexible strategy, but the objectives for the study should be decided in the beginning of the study. If the objectives change during the study this should rather be considered as a new study (Runeson and Höst, 2009). Research methodologies can serve different purposes such as exploratory, descriptive, explanatory and improving. The exploratory approach is when the purpose is to describe what is happening and when the researcher is searching for new insights, ideas and hypotheses for new research. The descriptive approach serves the purpose of portraying a situation or phenomenon. Explanatory research tries to find an explanation of a situation or problem. Lastly, the improving approach strives to improve a aspect of a studied phenomenon (Runeson and Höst, 2009). This master thesis has an exploratory purpose, which also is the most common purpose for case studies.

2.1.1 Case Study Design

The planning of the case study is crucial for its outcome. The planning should include what the objectives are, what case is studied, the theoretical framework used, which the research questions are, how the data is collected, and where the data is found (Runeson and Höst, 2009). Yin (2003) described the case study design as a blueprint for the research dealing with the objectives previously mentioned. The main objective with the design is to avoid evidence that does not address the initial problems or questions being studied (Yin, 2003). For our case study the objectives, the context of the case and the research questions are described previously in this chapter. The theoretical framework is presented in chapter 3 and the description of the data and its source is presented in chapter 4.

Establish Quality

According to Yin (2003) there are four tests that are common in social research to establish quality: construct validity, internal validity, external validity and reliability. Construct validity can be challenging and regards how to define the situation and identify measures of the situation being studied. Tactics for increasing construct validity is to use multiple sources of evidence, establish a chain of evidence and to get the case study report reviewed by key informants (Yin, 2003). Internal validity is mainly a concern for explanatory case studies and will therefor not be described (Yin, 2003). External validity regards the question of how the results can be interpreted and to what domains it can be generalized. Establishing reliability is to make sure that if the same procedure as described in the study is replicated, the investigator should arrive at the same findings and conclusions (Yin, 2003).
2.1.2 Data Collection

According to Runeson and Höst (2009) there are three levels of data collection techniques. The first degree is when the data is collected in real time, direct by the researcher e.g. in interview and focus groups. The second degree regards indirect methods where data is collected by the researcher directly without interaction with the subjects. The third degree is when the data already is available e.g accounting data (Runeson and Höst 2009). We will be focusing on the third one which includes data already collected for other purposes than the specific case study. This technique requires little resources to collect data (Runeson and Höst 2009). The data is not under control for the researcher and its quality may not be suited for this case study. The data might include company template data that is not interesting for research perspectives and has to be removed. Further, the data might not meet the quality requirements regarding validity and completeness. If data is collected by the researcher, the context, validity, and completeness can be controlled during the collection. The archived data then might need to be combined with additional data from other collection techniques as a complement. The researcher can also investigate the original purpose of the data collection to get a better understanding of it. (Runeson and Höst 2009) The benefits of using archival data is according to Yin (2003) that the data can be reviewed repeatedly, that it is not created as a result of the case study, it contains exact information of an event, it covers a long time span, and that it is precise and quantitative. As mentioned by Runeson and Höst (2009) there are also some weaknesses when using archival data. Yin (2003) suggests that the retrievability can be low, the data can be biased, and that there can be accessibility problems.

We will use data from Lundalogik’s customer data base to make our analysis. To get a better understanding of its purpose we will talk with employees to understand the work flow and the intentions of certain aspects. The validity is difficult for us to ensure but the fact that all inputs in the system are the foundation for Lundalogik’s business can be an argument for some degree of validity.

2.1.3 Data Analysis

The most common approaches to analysis of data are quantitative and qualitative. Quantitative data analysis is often based on statistics and statistical representations of the data. Methods that are used to describe and understand the data of the analysis are often mean values, standard deviations and histograms (Runeson and Höst 2009). Qualitative methods are most commonly used since the case study research is a flexible method (Runeson and Höst 2009). The most important objective of a qualitative analysis is to have a clear chain of evidence to the conclusions that are drawn, which means that it must be possible to follow the extraction of results and conclusions from the data (Runeson and Höst 2009). The analysis in our study
is conducted on quantitative data to find patterns in the data set that characterize a churner.

2.1.4 Reporting

The report should present the findings of the study but also make the reader able to judge the quality of the study (Runeson and Höst, 2009). There is according to Yin (2003) six different alternatives to structure the report: linear-analytic, comparative, chronological, theory-building, suspense and unsequenced. The linear-analytic structure is a standard reporting structure with problem, related work, methods, analysis and conclusions. A comparative structure is when the same case has been repeated at least twice to be compared. The chronological structure is suitable when the study has been performed over an extended time. Theory-building structure can be used to clearly show the chain of evidence to build a theory. Suspense structure starts with the conclusions and follows with the evidence that supports the conclusions. Unsequenced reporting structure can be used when reporting a set of cases. For an academic study the most accepted structure is the linear-analytic structure, which also is used for this master thesis.
Chapter 3

Theoretical Background

This chapter introduces the theoretical background for this master thesis. It describes the general concepts of customer relationship management, customer churn, and data mining.

3.1 Customer Relationship Management

Customer relationship management (CRM) provides the customer with personalized and individual attention regardless of who the customer is interacting with or which part of the organization. Galbreath and Rogers (1999) defines CRM as:

Activities a business performs to identify, qualify, acquire, develop and retain increasingly loyal and profitable customers by delivering the right product or service, to the right customer, through the right channel, at the right time and the right cost. CRM integrates sales, marketing, service, enterprise resource planning and supply-chain management functions through business process automation, technology solutions, and information resources to maximize each customer contact. CRM facilitates relationships among enterprises, their customers, business partners, suppliers and employees.

As described in chapter [1], most businesses have a lot of information, CRM focus on turning information into business knowledge for the organization to be able to better manage customer relationships. CRM is described as a way of creating a competitive advantage and it helps the business to be able to understand which customers are the most profitable, which to keep, which have potential and which are worthwhile to acquire. (Galbreath and Rogers, 1999)

The positive economic impact that can be obtained with a CRM is that according to Galbreath and Rogers (1999) a 5 percent reduction in customer
churn can result in a profit increase from 30 to 85 percent. If businesses also can manage to retain 5 percent more customers than today is equivalent to cutting their operating expenses by 10 percent. This concludes from the fact that it costs five to seven times more to acquire new customers than retaining the current ones. One should have in mind that the cost of the CRM-system is not included into these calculations. (Galbreath and Rogers, 1999)

3.2 Customer Churn Management

Customer churn is the term used for customers ending the relationship with a company. It has become a significant problem and has gained more and more attention in most industries (Neslin et al., 2006; Hadden et al., 2007). According to Hadden et al. (2007) retaining customers is the best market strategy to survive in the industry since it is harder and more expensive to find new customers than retaining current customers.

There are several reasons causing churn and Hadden et al. (2007) divides them into two groups: incidental churn and deliberate churn. A incidental churn happens when the circumstances for a customer changes so that it prevents the customer from further using the product or service. An example of incidental churn is changes in economic circumstances, which makes the product too expensive for the customer. Deliberate churn occurs when a customer actively chose to move their custom to another company that provides a similar service and this is the type of churn that most companies tries to prevent. Examples of deliberate churn is technology factors such as that the competitor offers better and more advanced products, economic factors such as better price and poor support (Hadden et al., 2007).

Hadden et al. (2007) does not believe that all customers should be targets for churn prevention for two reasons. First all customers are not worth retaining. Secondly working with customer retention costs money. By using the customer lifetime value (CLV) decision makers can easier identify profitable customer and develop strategies to target customers these customers (Liu and Shih, 2005).

Variables for Churn

Ballings and Van den Poel (2012) states that both customer characteristics and relationship characteristics are used in many analyses of customer churn. Three variables in the relationship characteristics are identified as the best predictors for customer behaviour: recency (R), frequency (F) and monetary value (M). Coussement et al. (2014) states that the RFM variables represent customer’s past behaviour and can also be used for customer segmentation. Recency represents the time that has passed since the customer made its last purchase (Coussement et al., 2014; Ballings and Van den Poel, 2012). The more time that has passed since the last purchase increases the risk.
for churn (Ballings and Van den Poel 2012). The frequency variable is the number of made purchases by a customer for an arbitrary time period where Ballings and Van den Poel (2012) has concluded that heavy and frequent buyers have higher probability to be loyal with a company and continue buying products from them (Coussement et al. 2014). Monetary value represents the total amount of money spent in past purchases and customers who have spent a high amount of money with a company are more likely to continue purchasing (Ballings and Van den Poel 2012; Coussement et al. 2014). Another top predictor is length of relationship (LOR), which has shown that customers with long term relationships are more likely to be loyal (Ballings and Van den Poel 2012). Other good predictors tested by other researchers are RFM-related predictors such as frequency related. That is where the frequency variables are used to construct more variables such as number of newspapers in last subscription, sum of newspapers across all subscriptions etc. that were used when predicting churn at a newspaper company (Ballings and Van den Poel 2012).

### 3.3 Data Mining

Data mining is the process of discovering interesting patterns in large sets of data. Data mining can be used on many sources of data such as databases, data warehouses, transactional data, data streams and the World Wide Web (Han et al. 2012). The main idea of data mining is to find data patterns or trends in the data that would have been really hard to recognize manually. The data mining process roughly contains the following procedures according to Han et al. (2012):

1. **Data cleaning** - the process of removing noise, inconsistent data and missing values.
2. **Data integration** - using data from more than one source.
3. **Data selection** - select the most appropriate data for the analysis.
4. **Data transformation** - the data needs to be transformed by using summaries and aggregations.
5. **Data mining** - extraction of data patterns.
6. **Pattern evaluation** - identify the interesting patterns.
7. **Knowledge presentation** - visualization and presentation of the mined knowledge.

There are two general categories of data mining functionalities: descriptive and predictive. The descriptive category focuses on characterizing the data depending on the properties of the data set while the predictive category uses induction on the data to be able to do predictions. This thesis
will focus on the predictive category and more precisely on classification. Classification is the process of finding a model that describes data classes by their common properties. The model is used to predict a class label for a data tuple without class label by determining which class the tuple is most similar to (Han et al., 2012). Section 3.6 describes classification more in detail.

Even more important than the algorithm used for data mining is the data itself. According to Gunnarsson et al. (2007) appropriate data is needed for a mining project. Otherwise the results will not be satisfying. The quantity of data is also important, even more important than having a great algorithm. According to Domingos (2012) a dumb algorithm with a lot of data is better than a clever algorithm with little data.

### 3.4 CRISP-DM

CRISP-DM was developed by the CRISP-DM consortium in 1996 and is a process model that describes the data mining process. A data mining project includes more than the mining itself as described earlier in this chapter. The CRISP-DM model is iterative and includes six steps, as can be seen in figure 3.1 below. (Chapman et al., 2000)

![Figure 3.1: The CRISP-DM model by Jensen (2012)](image)

Figure 3.1 describes the CRISP-DM model and its six phases. The phases are dependent on each other and what is done in a phase is determined by the outcome of the previous one. Since the method is iterative, going back and forth between phase is often needed. The arrows in Figure 3.1 show the most frequent routes between the different phases. Reaching the deployment phase does not mean that the mining project has ended, as the outer circle indicates. The information and experience gained from the first iteration are used to improve the mining project in the next iteration. (Chapman et al., 2000)
According to Mariscal et al. (2010) CRISP-DM is the most widely used methodology for data mining. The following section describes the phases of CRISP-DM in detail.

3.4.1 Business Understanding

Before the process of actually mining data it is important to define the objectives for the project. This requires a rigid understanding of the business and its objectives to fully understand what the project is set to accomplish and what benefits the business want to achieve. The objectives can be specific such as reducing the customer churn with a certain percentage or find customers for a targeted mailing campaign. Also, the evaluation method that will be used for the evaluation of the results should be determined early in the process since it is important to know that the result can be evaluated. This phase further includes assessment of resources, that is the available experts, tools, data etc need to be listed. Theses factors are important for planning and for the outcome of the project. (Chapman et al., 2000)

3.4.2 Data Understanding

The goal of this phase is to collect the data and get an understanding of it. If the data is not understood one can not know what can be done with it. Understanding includes identifying quality issues and detect interesting insights from which a hypothesis can start to develop. (Chapman et al., 2000)

3.4.3 Data Preparation

The data needs to be prepared before the mining models can operate on it. Raw data is often inconsistent and includes much more information than what is needed for the mining project. Data from different sources does not come in the same format and needs to be merged to a consistent data set. (Chapman et al., 2000) Preparation of data includes cleaning, integration of data from different sources, reduction, and transformation (Han et al., 2012). The steps will be further described in the following sections.

Data Cleaning

Data cleaning is the process of smoothing noise, filling in missing values, and correcting inconsistencies in the data. Data is often noisy and incomplete when extracted for a mining project and does not have a quality good enough to be mined. If the data set is a incorrect representation of the real-world, the result of the prediction will probably also be incorrect. The result of this is that the user will not trust the outcome of the mining project and it can also confuse the mining model which results in unreliable conclusions. (Han et al., 2012)
It is a common problem that attributes are missing in tuples in the data. The reason for this varies but if not handled the mining algorithm will have less information to operate on. A simple but not very effective method according to Han et al. (2012) is to ignore the tuple. The effectiveness increases when the number of missing values in the attributes increases in the tuple. When a tuple is ignored all other non-missing attributes are lost which could have been valuable in the analysis.

Another technique mentioned by Han et al. (2012) is to replace the missing value with a constant. A risk with this strategy is that the mining algorithm might find this as a valuable attribute, even though it has no meaning. Kimball and Caserta (2004) makes a difference between if the value is unknown or does not exist. The null value can then be replaced by either Unknown or Not applicable.

Noise is another problem that can be solved by cleaning. A variable may have a random variance or error which is noise. A technique to smooth noise is binning. Binning means grouping values together in bins. For example, bins for a numeric attribute could be specified saying that all values between 1 and 5 belong to bin one, values between 6 and 10 belong to bin two and so on. Binning can also be done by clustering values together and creating bins. (Han et al., 2012)

Data Integration

Data integration is needed to merge data from several sources, databases or tables. With a careful integration it is possible to reduce the redundancies and inconsistencies in the resulting data set that later on should be mined. Two of the major tasks in the data integration is to match attributes and objects from different sources and to examine if there is any correlation between two given attributes to minimize redundancy. Duplicates of tuples in the data set should also be resolved. The data integration might sound easy to execute but sometimes it can be hard to determine how the two different sources relate to each other. (Han et al. 2012)

Data Reduction

Data reduction is the technique for making the data set smaller but without loosing the integrity of the original data. In a data mining situation the data is likely to be very large and the reduction of the data should make the mining model more efficient without affecting the analytical result. A part of data reduction is selecting which attributes to use for the mining project. When predicting churn not all available attributes in the database are relevant. Attributes like telephone number and name of the company are likely to not have any significant effect on the prediction. Selecting the most relevant attributes can be done by an expert in the domain. Reducing the number of attributes can make the patterns identified by the algorithm easier to understand. (Han et al. 2012)
Data Transformation

In the data transformation step data are transformed into a format appropriate for mining. If this is done right, it will improve the result of the mining. This can be done with several techniques and which ones to use depends on the project. Examples of techniques to be used are smoothing, attribute construction, aggregation, normalization, discretization and concept hierarchy generation. (Han et al., 2012)

At times, the original data does not contain all necessary attributes for the mining process or can be extended with additional attributes to improve the result. These attributes might be collected from other sources or constructed. The attributes can be constructed from an existing set of attributes which gives the constructed attribute a meaning in the context of the project that the set of existing attributes lack themselves. (Han et al., 2012)

A real world database contains thousands of transactions of individual events. A data base might contain for example transactions of all sales for company. These individual sales transactions might not be informative from a mining point of view. But, if they are aggregated to sales per year for a certain area it can provide significant information for the mining. This is called aggregation and is a common technique used for analysis at several abstraction levels. (Han et al., 2012)

Another technique for transforming data mentioned by Han et al. (2012) is discretization. Discretization is used for attributes that are raw numeric values. The numeric attribute is replaced by an interval or conceptual label. For example, an attribute can be replaced by interval labels, 25-35 and 35-45, or conceptual labels, young or adult. These labels can then be organized into concept hierarchies, forming a tree structure. At the end of this phase the data should be ready to be mined. (Chapman et al., 2000)

3.4.4 Modeling

It is in the modeling phase the actual data mining begins. Here, modeling techniques are selected and implemented. Which techniques that are selected depends on the goal of the data mining project since different models solve different problems. Different techniques often have specific requirements of the data and therefore it is common that going back to the preparation phase is needed (Chapman et al., 2000).

3.4.5 Evaluation

Once the models are implemented and are considered to have desirable quality, it is time to evaluate them. Evaluation is important to conclude if the models fulfill the business objectives according to the selected evaluation methods. When this phase is over it should also be determined how the results should be used. (Chapman et al., 2000)
3.4.6 Deployment

The mining yields a lot of new knowledge. For this knowledge to be useful it needs to be presented in an understandable way for the business to be able to use it in the daily business. (Chapman et al., 2000)

3.5 Data Quality

Due to the growth of available information, the demand of data that is correct or of high quality, has also increased. The definition of high quality data is rather subjective and J.M Juran put it elegantly into words when he defined: ”data to be of high quality if they are fit for its intended uses in operations, decision making and planning” (Redman, 2004). According to The Data Warehousing Institute’s report on data quality, organizations in the U.S believe that they have more high quality data than they actually have. This perception costs U.S businesses more than 600 billion dollars a year in data quality problems (Batini and Scannapieco, 2006). Since data quality depends on the situation and the context for the data to be used, this section describes the dimensions of data quality, metrics for how to measure the quality dimension and common methods for increasing the quality of data. The dimensions, metrics and methods are later on used in this report to determine the quality of the data before and after the pre-processing of the database that is subject for our analysis.

There are several dimensions for describing data quality and the dimensions vary in the literature. Batini and Scannapieco (2006) defines the dimensions as accuracy, completeness, consistency, and currency. Han et al. (2012) use accuracy, completeness, consistency, timeliness, believability, and interpretability as their dimensions for data quality. To choose which dimensions to measure is the start of every data quality activity (Batini and Scannapieco, 2006). The following section describes these dimensions more thoroughly.

3.5.1 Accuracy

For data to have high quality it needs to be accurate, it needs to describe the reality correctly. Accuracy is defined by Batini and Scannapieco (2006) as ”the closeness between a value \( v \) and a value \( v' \), considered as the correct representation of the real-life phenomenon that \( v \) aims to represent”.

Accuracy can be described from two dimensions, namely syntactic accuracy and semantic accuracy. Syntactic accuracy is the distance between an element \( v \) and all elements in a domain \( D \). For example if \( v=\text{Jack} \) and \( v'=\text{John} \), \( v \) is syntactically correct since it exists in the domain of names. If instead \( v=\text{Jck} \) it is syntactically incorrect since there is no \( v'=\text{Jck} \) in the domain of names. Syntactic accuracy uses comparison functions to evaluate the distance between \( v \) and the values in \( D \). An example of such a
comparison function is edit distance that calculates the minimal number of operations to transform a string \( s_1 \) to \( s_2 \) (Batini and Scannapieco 2006).

The other type of accuracy, semantic accuracy, is the distance between a value \( v \) and the true value \( v' \). If \( v=\text{Jack} \) and \( v'=\text{John} \) the tuple is a semantic error. Semantic accuracy cannot be measured by functions as syntactic accuracy and needs to be measured by a binary statement such as correct or incorrect. To measure semantic accuracy the true value needs to be known or able to be inferred from additional knowledge. When a semantic error is due to a typo, semantic accuracy measures can be used to correct this by inserting the syntactically closest value under the assumption that that value is true. Another way to check semantic accuracy is comparing the same data in different sources. The problem here is to identify the same real world tuple in the different sources, called the object identification problem. (Batini and Scannapieco 2006)

### 3.5.2 Completeness

Another dimension of data quality is completeness that Batini and Scannapieco (2006) defines as "the extent to which data are of sufficient breadth, depth, and scope for the task at hand". If a data set is incomplete it means that the set is missing attribute value(s) or some attribute(s) of interest and possibly only containing aggregate date (Han et al. 2012). An important aspect of completeness is to understand why the data is complete/incomplete and what a missing value in the data set infers. Even et al. (2010) identified completeness as a key quality dimension when evaluating quality in a CRM-system. The reasons for a missing value can be that there does not exist a value for the attribute, the value exists but is missing in the data set or because it is unknown if the value exist or not. Missing values are often represented in a model as null and in general this means that the value exists in the real world but is not in the data set for some reason (Batini and Scannapieco 2006). The data set can be incomplete due to faulty input by the user, computational errors or faulty data collection instruments. For a model with null values Batini and Scannapieco (2006) defines several metrics to measure the completeness of model elements:

- **Value completeness (VC)** - Measures the completeness for some values in a tuple as seen in equation 3.1
  
  \[
  VC = \frac{\text{NumberOfNonNullValues}}{\text{NumberOfValuesMeasured}}
  \]  
  (3.1)

- **Tuple completeness (TC)** - Measures the completeness of a tuple for all its values as seen in equation 3.2
  
  \[
  TC = \frac{\text{NumberOfNonNullValuesInTuple}}{\text{NumberOfTotalAttributes}}
  \]  
  (3.2)
• Attribute completeness (AC) - Measures the completeness of null values of an attribute as seen in equation \(3.3\)

\[
AC = \frac{Number\ of\ NonNullValuesIn\ Attribute}{NumberOfTotalTuples}
\]  

(3.3)

• Relation completeness (RC) - Measures the completeness in a whole relation by evaluating the information available with respect to maximum possible information as seen in equation \(3.4\)

\[
RC = \frac{TotalNumberOfNonNullValues}{TotalAttributes \times TotalTuples}
\]  

(3.4)

### 3.5.3 Consistency

Consistency describes the violation of semantic rules in the data. The data model should have integrity constraints to ensure the data is consistent. These integrity constraints are the instantiation of the semantic rules. For data to be consistent all instances of the database must fulfill the integrity constraints. The constraints can be divided into two categories, namely intrarelational constraints and interrelation constraints. Intrarelational constraints are constraints that concern single attributes or several attributes of a relation. An example of an intrarelational constraint is that the attribute Age can only be between 0 and 120 years. (Batini and Scannapieco, 2006)

Interrelation constraints concern attributes of several relations. For interrelation constraints to apply, an attribute of one relation must correspond to an attribute in another relation. The first attribute is dependent on the second. It is common that constraints are dependencies. A simple type of dependency is the key dependency, which is commonly used. A key dependency ensures that each individual tuple has a unique identifier. For example could an attribute social security number for an entity person be used as a key. This means that there can be no duplication in the relation. (Batini and Scannapieco, 2006)

Another type of dependency is inclusion dependency. Inclusion dependency means that some columns of a relation are contained in other columns of that same relation or in columns of other instances of relations. An example is a foreign key constraint. (Batini and Scannapieco, 2006) A last type of dependency is functional dependency. Two sets of attributes, X and Y, for a relation r satisfy functional dependency if every pair of tuples, a and b, fulfills \(a.X = b.X\) and \(a.Y = b.Y\). (Batini and Scannapieco, 2006)

### 3.5.4 Time Dimensions

Both Han et al. (2012) and Batini and Scannapieco (2006) have a time dimension for describing data quality. The time dimension describes how
3.6. CLASSIFICATION

CHAPTER 3. THEORETICAL BACKGROUND

the data changes and gets updated with time in perspective. The names of the time dimensions vary in the literature but in general the dimensions describe how current the data is for the upcoming task and the fluctuations of how the data varies over time (volatility). The perspective of how current the data is, is an important perspective since the data could be too old for the upcoming task, e.g. in a marketing campaign where an advertisement will be sent by mail to the recipients, an important factor is how up to date the address of the recipient is. To measure the time dimension of the data quality, currency can be used. Currency means how current the data is and is measured on a scale from 0 to 1 where 0 is low currency and 1 is high currency. To measure the currency on the range from 0 to 1 the time scope, \( T \), of the analysis and the time from the beginning of the scope, \( t \) is used. The formula for attribute currency is shown in equation 3.5.

\[
Currency = \frac{t}{T} \quad (3.5)
\]

As an example, for a given year in the scope of the analysis the currency is measured as the number of years from the time scope until the last update divided by the total number of years in the scope. If the scope \( T \), is 10 years from year 2000, a tuple last updated in 2009 will have the value \( t = 9 \) and the currency 0.9 as seen in equation 3.6.

\[
\frac{t}{T} = \frac{9}{10} = 0.9 \quad (3.6)
\]

3.6 Classification

Data analysis with classification creates a model that tries to describe an important data class. The models are called classifiers and are used to predict categorical class labels. In the churn prediction case we can build a classification model to classify churners and non churners. Classification techniques are used in churn prediction, but also fraud detection, target marketing, performance prediction and medical diagnosis. \[\text{Han et al., 2012}\]

Classification is a process that consists of two steps. The first step is the learning step where the classification models are constructed based on a training set of data. A classification algorithm is used to learn from the training set to create the classifier. Depending on if the data set includes the class label in the training set of data the learning step is either called supervised learning or unsupervised learning. The difference between the two learning forms is that the supervised learning has the class label included in the training data set and unsupervised learning does not. The second step is where the actual classification occurs but first the predictive accuracy of the classifier should be estimated by running the classifier on a test set of data. The test set of data is not the same as the training set and does not include the class label (but they are known for evaluation reasons). The classifier runs through the test set and classifies the tuples. When the
classifier is ready, the accuracy of the classifier is evaluated by the percentage of the correct classified tuples compared to the actual class label of the test data set. (Han et al., 2012)

For a pattern to be interesting it needs to be understandable for humans, valid when extracted from new data, potentially useful, and novel. In summary, a pattern that a human can understand or know what to interpret can be used for further analysis. (Han et al., 2012)

### 3.6.1 Decision Tree Induction

Decision tree classifiers are popular since they do not require any domain knowledge, which makes them useful for exploratory knowledge discovery. The tree structure is rather intuitive, the training and classification steps are fast and in general they have good accuracy (Han et al., 2012). A decision tree is a tree structure where each one of the internal nodes represents a test of an attribute. The branches represent the different outcomes of the test in the internal node. A leaf node in a decision tree contains a class label, that is the class that will be given to the tuple. (Han et al., 2012)

In the late 1970s J. Ross Quinlan developed the ID3 decision tree algorithm. ID3 was later on used by Quinlan as the foundation of the C4.5 algorithm which became a benchmark for evaluating new supervised learning algorithms. At the same time the book Classification and Regression Trees (CART) was published independently from the work of Quinlan, although they follow a very similar approach for learning decisions trees from training sets of data. The algorithms use a greedy approach and the decision trees are constructed top-down and recursively partitioned into smaller subsets as the tree is created. Briefly described, classification with a decision tree of a given tuple \( t \) that is the subject to be classified travels down the decision tree. The attribute values of \( t \) are tested against the internal nodes of the decision trees and the path down the tree to the leaf node can be converted to classification rules. The leaf node itself holds a class label that \( t \) will be labeled (classified) with. The internal nodes that partition the data set use the splitting criterion to determine which way is the best way to partition the data set and which branches to grow from the internal node. (Han et al., 2012)

**Attribute Selection**

To decide where to create nodes in the tree and reduce the input to manageable size the algorithm uses attribute selection, also called feature selection. Attribute selection methods are heuristic procedures that finds the attribute that best differentiates classes. To select which attribute that are most relevant for the analysis every attribute is given a score based on the information it provides and then the attribute with the highest score is selected. (Han et al., 2012)
3.6. CLASSIFICATION CHAPTER 3. THEORETICAL BACKGROUND

How many and which branches the node is split into depends on what type of attribute it is. If the attribute is discrete, the node is branched into the number of possible values the attribute can be. For example, an attribute customer type where the tuples are divided into five different categories is discrete. When splitting at this attribute it would be branched into five branches, one for each customer type. It can also be split into category A or not category A. If the attribute is continuous the splitting criterion is different. For example, when the continuous attribute turnover is encountered by the algorithm, it finds a split point. This split point is a value, for example 100 000 and the node is split into two branches $A \leq 100000$ and $A > 100000$. \cite{Han2012}

3.6.2 Microsoft Decision Tree Algorithm

For the predictive modelling we will use Microsoft SQL Server 2012 and the analysis services provided in the package. The SQL Server suite and the analysis services provides a couple of implemented algorithms for building predictive models. The Microsoft Decision Tree Algorithm is a classification and regression algorithm that supports classification, association and regression. The algorithm can be used for predictive modeling of discrete and continuous attributes. When the attribute is discrete the algorithm identifies the attributes that have high correlation to the predictive attribute. The prediction is then based on the strongest relationships between the attributes and the predictive attribute. If the attribute is continuous the algorithm uses linear regression to be able to determine where to split the decision tree \cite{Microsoft2013b}.

The algorithm has the following requirements according to \cite{Microsoft2013b}:

- The input data must have a single key column that can uniquely identify a tuple in the data set. The key can be a String or integer.
- The algorithm requires at least one predictable column.
- The input attributes can be discrete or continuous. The number of attributes in the input data will increase the processing time.

To select which attributes that are the most useful the algorithm uses feature selection (attribute selection described above) to prevent that unimportant attributes get included in the predictive model. The implemented feature selection in the SQL Server suite is Interestingness score, Shannon’s entropy, Bayesian with K2 Prior, and Bayesian Dirichlet Equivalent with Uniform Prior. For sorting and ranking all non binary continuous numeric attributes the interestingness score is used. The other three alternatives are used for discrete and discretized attributes. For the Microsoft Decision Tree Algorithm the Bayesian Dirichlet Equivalent with Uniform Prior (DBE) is the default method for feature selection. \cite{Microsoft2013c}
Overfitting and Tree pruning

One of the major reasons for the rigorous preparation process of data before mining it is to reduce overfitting. That is when a decision tree learns and reflects irregular properties as a result of outliers and noisy data. The noise can confuse the algorithm since it tries to classify all tuples in the training data, including the noisy ones. This results in a specific model that performs well on the training data but poorly on new data. An overfitted decision tree also tends to be more complex. (Kerdprasop, 2011)

The problem with overfitting can be tackled with tree pruning methods, which use statistical measures to identify and remove branches in the tree that are the least reliable. Pruned decision trees are often less complex and smaller than unpruned trees, which also makes them easier to understand. (Han et al., 2012)

The two most used approaches according to Han et al. (2012) for tree pruning is pre- and postpruning. Prepruning is when the construction of the decision tree is limited early in the creation process at a given node by deciding not to split or partition at the node any further, making the node a leaf node. As described earlier information gain can be used to determine how good a split is and to assure that the node does not fall below a predefined threshold. The difficult part is to determine the threshold to avoid too simple trees or too little simplification. The postpruning techniques remove subtrees from a completed or fully-grown tree. The tree gets pruned by removing a subtree at a given node and replacing it with a leaf node. The class label at that leaf node is the one that is most frequent in the removed subtree. (Han et al., 2012)

3.7 Evaluation Metrics

This section describes how the results of a mining project can be evaluated.

3.7.1 Evaluating Classifier Performance

A very important step of predicting churners is to be able to trust the result of the prediction and measure how good or accurate a classifier is on predicting class labels. To evaluate the performance of a classifier, the concept of training and test set of data can be used. To evaluate a classifier with the same data set (training set) as it used to build the model will create overoptimistic estimates of the prediction. Instead it is better to use a test set of data that was not used for the training of the model. The class label of the tuples in the test set should be known to be able to determine how well the classifier predicts classes. To further be able to understand the different evaluation metrics the following terms must fully be understood: (Han et al., 2012).
3.7. EVALUATION METRICS  

Chapter 3. Theoretical Background

- **Positive tuples (P)** - A positive tuple is a tuple from the class that we find interesting, in our study this will be the class of churners.

- **Negative tuples (N)** - A negative tuple is a tuple that belongs to the other class than the interesting one, in our case non churners.

- **True positives (TP)** - A true positive tuple is a positive tuple that is correctly classified. E.g a churner classified as a churner.

- **True negatives (TN)** - A true negative tuple is a negative tuple that is correctly classified. E.g a non churner classified as a non churner.

- **False positives (FP)** - A false positive tuple is a negative tuple that is incorrectly classified. E.g a non-churning customer classified as churner.

- **False negatives (FN)** - A false negative tuple is a positive tuple that is incorrectly classified. E.g a churning customer classified as a non-churner.

The foundation of evaluating a classifier is to compare the classifier’s prediction with the actual class labels of the tuples. This can be done by creating a confusion matrix as seen in table 3.1 which tells us how good a classifier is in predicting certain classes (Han et al., 2012).

Table 3.1: Confusion matrix for churn prediction (Han et al., 2012)

<table>
<thead>
<tr>
<th>Predicted classes</th>
<th>Actual classes</th>
<th>churn = yes (1)</th>
<th>churn = no (0)</th>
<th>Total</th>
<th>Recognition (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>churn = yes (1)</td>
<td>TP</td>
<td>FN</td>
<td>P</td>
<td></td>
<td>(\frac{TP}{P})</td>
</tr>
<tr>
<td>churn = no (0)</td>
<td>FP</td>
<td>TN</td>
<td>N</td>
<td></td>
<td>(\frac{TN}{N})</td>
</tr>
<tr>
<td>Total</td>
<td>P’</td>
<td>N’</td>
<td>P + N</td>
<td></td>
<td>(\frac{TP+TN}{P+N})</td>
</tr>
</tbody>
</table>

A classifier’s accuracy is seen in the rightmost bottom corner of the confusion matrix in table 3.1. The accuracy measures the percentage of tuples in the test set that is correctly classified and is given by equation 3.7. The accuracy measure is a good measure when the number of positive and negative tuples are balanced.

\[
\text{accuracy} = \frac{TP + TN}{P + N} \tag{3.7}
\]
The error rate of a classifier measures the percentage of tuples in the test set that is incorrectly classified and can be done in two ways as seen in equation (3.8).

\[
\text{error rate} = \frac{FP + FN}{P + N} = 1 - \text{accuracy}
\] (3.8)

Other measures that are of interest are sensitivity and specificity that measure the true positive recognition rate and the true negative recognition rate, respectively. These measures are interesting because they highlight the class imbalance problem, which is when the class of interest is rare in the data set. In situations when the sensitivity is low and the specificity is high the resulting accuracy will still be high because of the majority of negative tuples. This is misleading because the classifier is bad in predicting the interesting class. Sensitivity and specificity are also seen in the confusion matrix (table 3.1) and are given by equations (3.9) and (3.10) (Han et al., 2012).

\[
\text{sensitivity} = \frac{TP}{P}
\] (3.9)

\[
\text{specificity} = \frac{TN}{N}
\] (3.10)

Precision is another measure that is used in classification and it measures the percentage of positive classified tuples that actually are positive. The precision measure is given by equation (3.11).

\[
\text{precision} = \frac{TP}{TP + FP}
\] (3.11)

**Cross Validation**

A technique used for validating classification is cross validation. When using cross validation, the data is split into \(k\) about equal sized partitions or folds. The data for the folds are randomly selected from the complete data set. One of the folds is used as test set and the others are used for training. The algorithm iterates over all folds so that each fold is used once for testing. Assume the data is split into folds \(D_1, \ldots, D_k\). In the first iteration \(D_1\) is used for testing and \(D_2, \ldots, D_k\) for training. In the next iteration \(D_2\) is used for testing and \(D_1, D_3, \ldots, D_k\) for training. This goes on until all folds has been used for testing. The results from all iterations are then averaged. This way, uneven representations of data in test and training sets are reduced. (Witten et al., 2004)

Often the folds are stratified to make them representative. Stratification means that each class in the complete data set should have an equal representation in each fold. When the random selection of data is done it should then be ensured that there is an equal distribution of the classes in each fold. (Witten et al., 2004)
According to Witten et al. (2004) there is some theoretical evidence showing that 10 is the best number of folds to estimate error. Even though there is no clear evidence that this is the best and it is debated, tenfold cross validation has become more or less standard in practice. Witten et al. (2004) then continues by saying that there is no magic with 10 folds. The results from 5 folds or 20 folds is likely to be similar to 10 folds.

### 3.7.2 Microsoft’s Evaluation Metrics

This section describes the evaluation methods provided by Microsoft.

#### Decision Tree

Each iteration of data preparation and algorithm configuration is evaluated by a number of measures provided by Microsoft. As described previously in this chapter a pattern must be understandable by a human to be useful. After running the algorithm the decision tree for the classification is shown. In the tree, the most deciding attributes are shown to be interpreted by the user.

Figure 3.2 shows an example of a decision tree. The nodes represent the attributes that is strongly correlated with the column that is being predicted. When the algorithm splits depends on the attribute, if it is continuous or discrete.
discrete, and on the selected splitting method. The splitting methods were
briefly described in section 3.6.2. In the tree in figure 3.2 the attribute
Start year seems to be significantly correlated with the predictable column
Churn. Following the tree to the right there are more attributes correlated
with Churn but the longer to the right, the correlation is less significant.

(Microsoft 2013b)

The nodes are coloured on a scale from grey to blue. Nodes coloured grey
indicates few churners. The bluer a node gets the more churners there are in
this class. For example, the path StartYear < 2007andMaintradeCategory =
Missing includes a lot of churners. When hoovering over a node the user
gets information about the tuples in this category. In the figure there are in
total 421 cases belonging to this class. Of these, 297 are churners and 124
are non-churners.

Lift Chart

After running the Decision Tree Algorithm, the model must be evaluated.
One way of evaluating the performance of the predictive model is to use the
lift chart seen in figure 3.3. In the following list the features of the lift chart
(figure 3.3) is explained:

- The y-axis in the lift chart shows the percentage of the target popu-
  lation (churners).

- The x-axis in the lift chart shows the percentage of the overall popu-
  lation (total number of customers).

- The dark line at \( x = 50 \) is a ruler that determines the x value that
  will be compared in the mining legend.

Lift charts are used to compare the model with an ideal model and
a random model. These are two theoretical models to evaluate the ones
created against. The ideal model represents a model that always predicts
the outcome correctly. The random model use random guessing to select
customers and represent the result if churners would be evenly distributed
among the overall population. For example if we have an overall population
of 1000 customers including 100 churners we would find 10 churners by
selecting 100 customers randomly. This is what the blue line in figure 3.3
represents. If the ideal model were used instead we would find 100 churners
when 100 customers were selected.

The lift is defined as the difference between the used model and the
random model. Lift charts are often used when one wants to target resources
to improve the response rate. Randomly selecting customers for identifying
churners would result in finding a few churners. If a predictive model is used
the same percentage of the customers could be targeted but the number
of churners would increase. The lift represents how many more churners
that would be found when the predictive model is used instead of randomly selecting customers, this way resources could be allocated better.

With every lift chart a mining legend is provided that is used to interpret the chart. The mining legend can be seen in figure 3.4.

![Microsoft Decision Tree Lift Chart](image)

**Figure 3.3: Microsoft Decision Tree Lift Chart**

Description of the mining legend (figure 3.4):

- At the top the population percentage tells us at what percentage of the total number of customers ($x$-value) the gray ruler is placed.

- The score is a measure to compare different models. It is calculated on a normalized population where a high score is better than a low.

- Target population tells us the percentage of churners that can be targeted with the different models by using $x\%$ of the total number of customers.

- The predict probability is the accuracy for each prediction and is stored in the model for each tuple.
3.7. EVALUATION METRICS

CHAPTER 3. THEORETICAL BACKGROUND

Figure 3.4: Microsoft Decision Tree Mining Legend

To interpret the lift chart we start by looking at the curve for the ideal model. As described earlier, this is the curve for a made-up perfect predictive model that always make the right predictions. In the mining legend (figure 3.4) we can interpret the result as with 49.53% of the total number of customers (x-value of the dark line in the lift chart) we can target 100% of the churners. The random guess model can target 50% of the churners with 49.53% of the total number of customers. These are the two extremes and the constructed model will perform in between.

For the predictive model that was constructed 91.52% of the churners can be targeted with 49.53% of the total numbers of customers, which gives us a lift of 52%. The values (or companies in our case) on the x-axis are ordered by the predict probability for each of the customer. This means that the first 10% of the population is the customers with the highest predict probability. In a real-life scenario this means that to be able to target 91.52% of the churners with the predictive model we need to target the 49.53% of customers with the highest predict probability. In other words, the customers with a predict probability of at least 23.84%. The predict probability is used to filter the targeted customers with e.g. a query (Microsoft, 2013d).

Classification Matrix

Another measure for evaluation provided by Microsoft is the classification matrix. The classification matrix sorts the tuples into four categories that are false positive, true positive, false negative, and true negative. Descriptions of the categories can be found in section 3.7.1.

Figure 3.5: Microsoft Decision Tree Classification Matrix

An example of a classification matrix can be seen in figure 3.5. The predicted value here is churn where 1 indicates churn and 0 indicates not churn. Of $472 + 41 = 513$ predicted non-churners the actual number of non-churners were 472, which are the true negatives. The other 41 that were predicted as non-churners but actually are churners are the false negatives.
On the second row there are $84 + 360 = 444$ customers predicted churners. Of these 84 are non-churners, and namely false positives. The last 360 customers are churners and predicted churners and are the true positives. The accuracy is calculated with equation 3.7, which gives us $\frac{360 + 472}{513 + 444} = 0.87$ or 87% accuracy.

The matrix makes it possible for the user to easily understand the results. The amount in each cell tells how often the model predicted accurately. From the matrix percentages of each category can be calculated and used for analysis. (Microsoft 2013d)

**Cross Validation**

The cross validation report provided by Microsoft includes several different measures. Results are shown for each fold and for the data set as a whole as a mean of the folds. First the classified tuples are divided into categories, false positive, true positive, false negative, and true negative, as in the classification matrix. Next, there are three other measures, namely Log Score, Lift, and Root Mean Square Error (Microsoft 2013a). Microsoft’s Log Score, Lift and Root Mean Square Error from the cross validation will not be considered during our analysis.
Chapter 4

Case study

This chapter describes the context of the master thesis, the data that will be examined and all the steps of the data mining and predictive analysis performed at Lundalogik’s customer database.

4.1 Business Understanding

Lundalogik’s core business is to develop and sell two different CRM-systems. We focus in this chapter on the more simple and most sold system LIME Easy. LIME Easy is sold as a package of licenses and a service contract. The price for the system is 4000 SEK for each licence and 800 SEK for the service contract for each license and year. The customer must buy this package and there is no way to exclude the service contract. After the customer has bought this package they have a variety of add-on’s such as integration with the current economic system to purchase after their liking. A customer can also buy help from a consultant that can help with imports from other CRM-systems and setting up the environment etc. The service contract that every customer is forced to buy includes support from Lundalogik’s support center and possibility for additional educational seminars of the system. When the service contract has ended the customer can choose to end its contract but they will loose the possibility for support and upgrades of the system.

4.1.1 Definition of Churn

The definition of churn at Lundalogik is a customer that ends the service contract. When a customer buys a license, the customer owns the software which generates a one time income for Lundalogik. For Lundalogik to keep making money on the customer, the customer has to sign new service contracts. If a customer ends the service contract they can still use the software even though it is seen as a churn from Lundalogik’s perspective.
Eventually, with this definition in mind, what we look for is customers who are about to end their service contract.

4.2 Data Understanding

Lundalogik registers all their customer contacts in a custom built version of their own CRM-system. The information is added by the employee that has new information about a company including sales people, consultants, support team and application developers. The core function of the system is to keep all the information about Lundalogik’s customers in the system so that it is shared and is not lost. This is very valuable in for if an employee retiring. All customers of the retiring employee are recorded in the system and not on a local file, which makes it easier for the company to retain the customers even though the person that initiated the contact and was responsible for the customer has retired.

The main layout of the system is designed as a business card. Every customer has a company card with all the company information including all the people that are known at the company. The customer information is stored in standalone tabs like history, licenses, support errands, invoices etc. where each tab has its own table in the database. An example of a company card can be seen in figure 4.1. When a sales person contacts a customer they update the history in the company card with a note of what has been made and what has been decided. The information can be as simple as [2014-01-02 - Meeting booked: Product demonstration] with additional emails and documents attached if necessary. All information entered in a company card and its tabs are stored in a corresponding table in the database. A description of the relevant tables for our analysis follows below.

**Company Table**

The company table (table A.1 in appendix A) contains all companies that Lundalogik have registered in their CRM-system. The table includes general company information such as name, address, visiting address, contact person etc. It also contains a unique ID for each company, the buying status of the company, what kind of customer the company is and what status they have on their service contract. The table has a lot of attributes that are rarely used and does not provide any important information about the customer company. The table also contains attributes that are extracted from an external service for company information. Examples of attributes that are extracted are: turnover, number of employees, maintrade category and credit rating.
4.2. DATA UNDERSTANDING

CHAPTER 4. CASE STUDY

Figure 4.1: Snapshot of a company card with dummy values

License Table

The license table (table A.2) contains all licenses for products that Lundalogik has sold. The table includes general information about the license such as start date, end date, quantity, type, which company that owns the license, and which product the license is for. There is also a unique ID for all licenses, which also is the primary key. All products a customer has requires a license and is stored in this table. When a customer upgrades the quantity or buys more products a new license is issued. The table contains all old licenses so that all licenses a customer has can be found.

History Table

The history table (table A.3) contains all the notes that has been made in the history. Every note has an unique ID which is primary key. The table also contains the user who made the note, the date and time of the note, the topic, which customer the note concerns, and the text for the note among other things. All interaction with a customer is manually logged by a user in the history.

SOS Table

An SOS occurs when a customer contacts Lundalogik with a problem concerning LIME. Every SOS has a unique ID as primary key. The table also
contains general information about the SOS such as the date, the customer who contacted, how the contact was made etc. When an SOS is taken care of, a done date is inserted. An SOS can be given a priority; important ones can be sorted out. Notes for the SOS are also stored describing the problem.

**Product Table**

The product table (table A.4) contains all the products that Lundalogik offers. All products have a unique ID as primary key which is used for identification in the other tables. The products are divided into product families. There is a value, parent, for each product to find out which family it belongs to. The products also have a level value where the level in the hierarchy the product is at is stored. This makes it possible to select all products within a certain family. Naturally, the name of the product is also in this table.

### 4.3 Data Selection

To be able to prepare the data the most relevant attributes from the previously described tables is extracted to a new single table. Other attributes from the tables are used later on in the study to make aggregations, creation of new attributes and cleaning tasks for the final data warehouse.

#### 4.3.1 Data Quality at Lundalogik

After the extraction we analyzed the data quality before the preparation. To be able to see how the data quality gets affected the quality is analyzed before and after the preparation. For this task we have made assumptions and/or calculations of the accuracy, attribute completeness, consistency and currency with the extracted attributes before the preparation begins.

**Accuracy**

The accuracy is very hard for us to measure. We have no possibility to investigate if the values in the database are accurate. The only indication that is found regarding whether the data is accurate or not in the information that Lundalogik has bought from PAR. PAR is short for ”postens adressregister” which is a register of companies in Sweden. Since about 50% of the companies does not have information from PAR this is not a good enough measure to conclude whether the data is accurate or not. Because of these reasons the accuracy dimension of the data quality is ignored during the quality evaluation of the different iterations.
Attribute Completeness

First the attribute completeness (AC) is analyzed in table 4.1 where the attributes that do not have 100% attribute completeness is shown. As seen in table 4.1 the attributes that contain information about a customer’s main-trade category, number of employees and turnover has a lot of missing values. This information is bought by PAR and because of that there is no possibility to access the information if it is not present in these attributes. The attributes named customer type, postal area and salesperson are attributes were all the information comes from inside Lundalogik and it might be possible to clean these values to increase the attribute completeness.

Table 4.1: Attribute Completeness

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Null Count</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer type</td>
<td>234</td>
<td>96%</td>
</tr>
<tr>
<td>maintrade category</td>
<td>2668</td>
<td>52%</td>
</tr>
<tr>
<td>no of employees</td>
<td>2373</td>
<td>57%</td>
</tr>
<tr>
<td>postal area</td>
<td>142</td>
<td>97%</td>
</tr>
<tr>
<td>salesperson</td>
<td>736</td>
<td>87%</td>
</tr>
<tr>
<td>turnover</td>
<td>1901</td>
<td>66%</td>
</tr>
</tbody>
</table>

Currency

Regarding the time dimension and currency described in chapter 3, the data quality might be dependent on how current the data is. The timestamps in the database are incorrect as a result of a database migration and therefor we will use the date when the customer got acquired as a measure of currency since the company information is rarely updated. Table 4.2 shows the total number of customers that Lundalogik has acquired over the years. As described in chapter 3 the currency is calculated with regards to the scope of the study and/or the data. In table 4.2 we have a scope, $T = 12$ years from 2002 to 2013 for the acquired companies. To be able to determine the currency for the whole data set we will use the average currency for the data set given by equation 4.1.

$$
\frac{\sum_{t=1}^{T} \frac{T}{T} \cdot N_t}{N_{tot}}
$$

(4.1)

By using equation 4.1 the calculated currency for the data set is 0.42,
which tells us that the data is not very current. When investigating the customers that were acquired in 2002 we find out that there must have been some kind of migration of old customers that has happened in 2002 since almost every customer of the 1837 customers has been created the exact same date and time. Another interesting observation is that another common property of these attributes is that the information related to these companies is very sparse and incomplete.

Table 4.2: Acquired customers

<table>
<thead>
<tr>
<th>$t$</th>
<th>Year</th>
<th>Customers ($N_t$)</th>
<th>Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2002</td>
<td>1837</td>
<td>0.08</td>
</tr>
<tr>
<td>2</td>
<td>2003</td>
<td>234</td>
<td>0.17</td>
</tr>
<tr>
<td>3</td>
<td>2004</td>
<td>314</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>2005</td>
<td>336</td>
<td>0.33</td>
</tr>
<tr>
<td>5</td>
<td>2006</td>
<td>342</td>
<td>0.42</td>
</tr>
<tr>
<td>6</td>
<td>2007</td>
<td>410</td>
<td>0.50</td>
</tr>
<tr>
<td>7</td>
<td>2008</td>
<td>398</td>
<td>0.58</td>
</tr>
<tr>
<td>8</td>
<td>2009</td>
<td>450</td>
<td>0.67</td>
</tr>
<tr>
<td>9</td>
<td>2010</td>
<td>419</td>
<td>0.75</td>
</tr>
<tr>
<td>10</td>
<td>2011</td>
<td>390</td>
<td>0.83</td>
</tr>
<tr>
<td>11</td>
<td>2012</td>
<td>345</td>
<td>0.92</td>
</tr>
<tr>
<td>12</td>
<td>2013</td>
<td>108</td>
<td>1.00</td>
</tr>
</tbody>
</table>

$N_{tot} = 5583$

4.4 Data Preparation

With the data quality analysis in mind the data should now be prepared for modelling. First this section describes the initial data preparation made to the data set and highlights quality issues in the different aspects of the data. Secondly the attribute constructions and aggregation are described. Lastly, since CRISP-DM is a iterative framework, the pre-processing and analysis are performed in iterations and therefore we strive to improve or vary the data quality between the iterations to see how it affects the result of the churn prediction.
4.4.1 Data Cleaning

After the initial extraction of the interesting attributes from the tables described previously in this chapter, cleaning of the data is necessary. The goal of the cleaning is to decrease the number of inconsistent values, remove noise and incomplete tuples and/or attributes (Han et al., 2012).

**Noise**

The database stores potential customers (prospects), customers that have a trial license, customers that are outside of Lundalogik’s segment and customers that have chosen a competitor. All these customers are removed from the data warehouse since they are outside of the scope for this analysis.

**Consistency**

As described earlier, Lundalogik’s customers have a service contract that comes with the first purchase of the system and has an expiration date of one year. The customer can then choose to renew the contract for another year or terminate the contract. Some of the customers have been distributors of the system and received a service contract that never ends. These customers are an exception from their regular business model and do not represent a normal customer. Noise can be handled by removing the tuples that create this noise and since the number of tuples with a never ending service contract is relatively low (< 20) the tuples will simply be removed from the data warehouse (Han et al., 2012).

In the attributes **buying status** and **service status** Lundalogik records information about the relation they have with the customer. In the **buying status** they record if the customer is a potential customer, a current customer or a churn. The **service status** attribute records information about their relation from the service contract perspective. They record information if the customer has got a new invoice for renewal of the service contract, if the customer has returned the service contract or if the customer has chosen to end the service contract. These attributes are not dependent on each other but they give us information about the same thing, namely if the customer has churned or not. Since the attributes do not have any constraints or dependencies a customer can have a churn status in the **service status** attribute but still be classified as a customer in the **buying status**. To decide which one of the attributes that best represents the customer situation the license table has been used to determine whether a customer is a churner or a current customer. In the license table the **end date** attribute tells us if the customer has ended its license or not. If the **end date** is NULL the license is still active but since customers can have upgraded their licenses there can be many different licenses to consider whether the customer has churned or not. The main problem here is that the internal processes for registering a churn have changed. To make a somewhat credible classification
of churn an SQL-script with an if-else-structure was constructed to process each case step by step. The goal of the script is to classify whether the customer has at least one active license or not. The attribute `license_active` was constructed and it could be 0 for an inactive license and 1 for an active license. To determine whether the customer has churned or not we used the sum of the `license_active` attribute for a given customer. If the sum is greater than 0 the customer is still a customer, else a churn.

Lundalogik has an internal classification of customer depending on how many licenses the customers has bought. This classification is stored in the company table as the attribute `customertype`. They have five different class labels for their customers depending on how many licenses the customer has but sometimes the rules for the customer classification is violated as a result of a subjective classification by the salesperson. Although the classification is violated the data already includes a `quantity` attribute and we will not neglect the domain knowledge of experienced salespersons at Lundalogik. By correcting these violations the customer type classification would result in a discretized attribute of the `quantity` attribute, which we can achieve without wasting domain knowledge from within the company.

**Missing Values**

Each customer in the database should have a responsible salesperson from Lundalogik but 703 of the tuples are missing this value. Many of these customers were added to the customer database in 2002 and there is no or very little interaction with these customers. There might be different reasons for why the responsible salesperson is missing such as changed internal procedures. The database for the CRM-system has been subject for several migrations and updates, which can be seen in the tuples without salesperson since the information in these customers is very sparse. These customers do not very well represent Lundalogik’s current business model or how they work today and we will investigate if these customers affect the result of the data analysis.

Some of the customers have not been classified in the `customer type` attribute and to be able to fill in these missing values we have used Lundalogik’s class label rules that are dependent on the quantity of licenses to determine which class label the company should have.

**4.4.2 Data Reduction**

Data reduction is used to create a smaller set of the data that retains the integrity and properties of the original data set (Han et al., 2012). All attributes in the customer database have not been used as described earlier in this chapter and the most relevant attributes were extracted to the foundation for our data warehouse. Due to our small data set, no further data reduction is performed to further decrease the size of the data.
4.4.3  Data Transformation

In this section all the aggregated attributes are presented and explained.

Aggregations

We have created a number of aggregations in our data warehouse for analysis. The aggregated attributes are described in table 4.3. This section now describes and explains the different aggregations, what the information tells us and why it is important for the analysis.

Changes in Number of Licenses

According to Ballings and Van den Poel (2012) frequency related variables usually are good predictors. Because of this fact the net change in licenses for a customer from the beginning of the relations is aggregated into the attribute license_changed_from_start. This information is aggregated from the license table where every license a customer has had should be recorded, the old licenses with the old quantity should have been inactivated and the latest license with the current quantity should be marked as active. To determine whether licenses are active or not, the license_active is used, which is more accurate than the active attribute in the original license table.

Number of Changes in Licenses

Another aggregated frequency related attribute is no_of_changes_in_licenses that is describing how many license changes a customer has made.

Number of Additional Products

The attribute no_of_products is aggregated from the information of how many additional products a customer has. The information is stored in the license table as product licenses for each additional product the customer has bought. The attribute is a count of how many different products except for the initial CRM-system the customer has bought.

Number of Licenses for Additional Products

This is an aggregated frequency related attribute to the additional products a customer might have bought. The information also comes from the license table described in the previous section but instead of counting the number of products the total amount of additional licenses is summarized.

Cost of Consultation

Variables that describe monetary value are good predictors (Ballings and Van den Poel 2012; Coussement et al. 2014). The cost_of_consultation attribute is the sum of all the consultations a customer has bought. Consultation can be bought to get started with the system, migrate information and to get educated in the system. The consultation cost is registered in a
Table 4.3: Aggregations

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>license_changed_from_start</td>
<td>Describes the net changes of licenses.</td>
</tr>
<tr>
<td>no_of_changes_in_licenses</td>
<td>Describes the number of changes in licenses a customer made.</td>
</tr>
<tr>
<td>no_of_products</td>
<td>Lundalogik is offering a number of additions to their products. This attribute represents the number of other products a customer has bought.</td>
</tr>
<tr>
<td>no_of_licenses</td>
<td>Describes the quantity of licenses sold for the additions.</td>
</tr>
<tr>
<td>cost_of_consultation</td>
<td>All the cost of consultation logged on a customer summed in SEK.</td>
</tr>
<tr>
<td>relation_length</td>
<td>The length of the relation measured in months.</td>
</tr>
</tbody>
</table>

Relationship Length
The length of relation is a top predictor except for the RFM-variables [Ballings and Van den Poel 2012]. For this reason the length of every customer’s relation with Lundalogik is aggregated into the attribute relation_length. The attribute is aggregated from the license table where the first license determines the starting year and the last licence or 2013-12-31 determines the end date.

Attribute Constructions
We also have created some attributes for the data warehouse. The constructions are combinations of conclusions from other attributes in the data base and are described in table 4.4.

When a customer has bought a CRM-system the interaction with Lundalogik decreases. The customer should now be able to handle the system after the initial education and will probably only contact Lundalogik if they experience problems with the system or need guiding in how to work with the system. To increase the interaction with the customers the marketing
department at Lundalogik offers the customers a broad range of events such as seminars, educations and information.

**Average Number of Invitations Received**

Customers get invited by e-mail or phone and all the invitations that are sent out are recorded by the marketing staff in the CRM-system. The marketing staff also records which persons and from which company that have accepted the invitation and which persons that actually participated. To be able to compare the number of invitations between customers an average depending on the relationship length in years is constructed. The constructed attribute is called `mrkt_invi_avg`. To summarize, the attribute `mrkt_invi_avg` stores the average amount of invitations a customer has received per year.

**Average Number of Participated Events**

With the information about how many events a customer has been invited to, we also want to know how many events they have participated in. To be able to aggregate information about how many events a customer has participated in we must understand how Lundalogik records the customer participation at an event. Lundalogik has six different statuses to record who will and who will not attend an event. Two of these are representing participation, which means that these two statuses are of interest when aggregating the participation for marketing events. The constructed attribute is an average of the number of events a customer has participated in per year and is called `mrkt_part_avg`.

**Average Number of Support Errands**

Lundalogik has a support department that takes care of most of the support errands when a customer is having problems or needs help in some way with their CRM-system. The support errands are recorded in the support table and the support is only available for customers that have a valid service contract. To be able to know whether the number of support errands influence the churn or not an average of the number of support errands per year is constructed. The attribute is called `sos_avg`. This attribute holds the average amount of support errands per year a customer has had.

**Average Number of History Records**

Whenever Lundalogik has any interaction or meeting with a customer they record short descriptive records in the history table. This is the core of their CRM-system, all information and interactions with a customers is stored as history events. This information is valuable for Lundalogik since regardless of who that is in contact with a customer they all have the same information (given that they use the history correctly). To reflect the attention a
customer has been given the attribute $\text{history}_{\text{avg}}$ is created. The attribute is the average number of history entries for a given customer per year.

**Churn**

The predictive attribute is the variable that is subject for the prediction. The $\text{churn}$ attribute is the predictive attribute for our analysis and is created based on the date when the $\text{service agreement}$ ends. If the date is before 2013-12-31 the customer has churned, otherwise they are still considered a customer.

**Start Year**

As described earlier in this chapter the currency of the data was not great. To be able to see if the currency or the year a customer got acquired by Lundalogik the attribute $\text{start}_{\text{year}}$ is constructed, which simply holds the year when the customer got acquired.

**Discretizations**

**Postal Area**

The attribute postal code has been discretizised into nine different regions in Sweden. There are many postal codes with very few customers in each one. By grouping them into larger areas (regions) will make the postal codes easier to interpret.

### 4.4.4 Structure of Pre-processing

As described earlier in this section the data preparation differs among the iterations. Data mining is a process and it is important to remember that some of the decisions made in the beginning regarding the data might change later in the process (Gunnarsson et al., 2007). The changes are made based on observations in the data, its properties and to be able to analyze if or how the data quality affects the churn prediction. For all iterations the attribute $\text{Sales person}$ has been randomized for ethical reasons.

**Iteration 1**

In the first iteration all data is selected for the mining and prepared as described previously in this section without any further discrimination. The number of rows in the data warehouse can be seen in table 4.5.

**Iteration 2**

A trend in the data is that the earlier start date a customer has, the more missing values it has. The further back in time, the less representative for
### Table 4.4: Attribute constructions

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrkt_invi_avg</td>
<td>Describes how many marketing events arranged by Lundalogik a customer has been invited to per year during the relation. Only events that require an effort from the customers are included.</td>
</tr>
<tr>
<td>mrkt_part_avg</td>
<td>Describes how many of the marketing events arranged by Lundalogik a customer has participated in per year during the relation. Only events that a customer physically can participate in are included.</td>
</tr>
<tr>
<td>sos_avg</td>
<td>The average number of sos a customer has sent per year during the relation.</td>
</tr>
<tr>
<td>history_avg</td>
<td>The average number of history records for a customer per year during the relation.</td>
</tr>
<tr>
<td>churn</td>
<td>This is the predictable column for the analysis. This is a discrete attribute where 1 indicates churn and 0 not churn. All customers where the end date of their service agreement before 2013-12-31 are classified as churners.</td>
</tr>
<tr>
<td>start_year</td>
<td>The year that a customer got acquired by Lundalogik.</td>
</tr>
</tbody>
</table>

### Table 4.5: Number of rows for each iteration

<table>
<thead>
<tr>
<th>Iteration</th>
<th>No. of rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4597</td>
</tr>
<tr>
<td>2</td>
<td>3267</td>
</tr>
<tr>
<td>3</td>
<td>3267</td>
</tr>
<tr>
<td>4</td>
<td>3267</td>
</tr>
<tr>
<td>5</td>
<td>2930</td>
</tr>
</tbody>
</table>
customers of today. Therefore, customers with start date before 2004 are ignored for iteration 2. As seen in table 4.2, a lot of the data comes from year 2002 and we will investigate how the churn prediction gets affected with and without this data.

**Iteration 3**

To smooth noise in the data some of the continuous columns where noise is present have been discretized. The discretization method used is binning and the data is binned through clusters. Also, as in iteration 2 customers with start date earlier than 2004 is ignored.

**Iteration 4**

Some attributes have a high ratio of missing values, or low attribute completeness. For the columns maintrade category and number of employees company it is about 50 percent after the initial pre-processing. For this iteration the columns are excluded from the data. Also, all customers with start date earlier than 2004 are ignored.

**Iteration 5**

In the last iteration customers where maintrade category and number of employees company are missing are ignored to increase the completeness of the analyzed data. Also, customers with start date earlier than 2004 is ignored.

### 4.5 Modeling

For the modeling we have used Microsofts Decision Tree Algorithm described in chapter 3. To improve the results, we have tried a few different configurations of the algorithms. This is done through the parameters also described in chapter 3. We also have tried different preparations of our data warehouse to see how that affects the results. According to CRISP-DM we have moved forth and back between the preparation and modeling step after observing the results.

#### 4.5.1 Algorithm Configuration

Before running the algorithm the data is split into a training set and a test set. 70 percent of the data is used for training and 30 percent for testing. The *MINIMUM_SUPPORT* is used to determine the minimum number of leaf cases needed to split the node. This parameter is set to 2% for all three configurations.

The attribute selection method is not invoked automatically. The *MAXIMUM_INPUT_ATTRIBUTES* parameter must be set for the algorithm to
invoke attribute selection and is for the three different configurations set to 0, 5, and 8 respectively. For a model that contains more input attributes than specified in the parameter the uninteresting attributes will be ignored. 0 in the parameter means that attribute selection remains uninvoked and is used to compare how feature selection affects the result of the model.

If a model contains more columns than the numbers described in chapter 3 there are different methods for attribute selection. We have used the default method for the Microsoft Decision Tree Algorithm, which is Bayesian Dirichlet Equivalent with Uniform Prior (BDEU). Although, since our model contains continuous attributes, the default method will be overridden and the interestingness score will be used to ensure consistency since BDEU can't be used for continuous values.
Chapter 5

Result & Analysis

This chapter presents and analyzes the results of the data mining.

5.1 Data Quality Results

The mining was performed in iterations with different preparations of data for each iteration. For each preparation the algorithm was configured in three different ways. The configurations were the same for all iterations, the difference between the iterations is the way the data is prepared. Below follows a short description of how the data quality has changed as a result of the data preparation for each iteration. Completeness (attribute completeness) and currency is used as the key dimensions for evaluating how the quality changes. The consistency dimension will be constant throughout the iterations after the initial data preparation.

5.1.1 Iteration 1

The attribute completeness after the initial data preparation and pre-processing is found in Appendix B. For all the attributes an increase in attribute completeness has occurred. The sales person attribute had the highest increase with 11% and postal area the lowest with 1%. The result is expected, during the cleaning of missing values sales person was one of the attributes with a lot of missing values and sparse information. We used an approach where we removed the tuples with missing values and sparse information, which results in a higher attribute completeness.

The currency of the data before the initial preparation was 0.42 and has now increased to 0.46 which gives us slightly more current data for the analysis.
5.1.2 Iteration 2

A trend in the data is that the earlier start year a customer has, the more missing values it has. The further back in time, the less representative for customers of today. Therefore, customers with start date before 2004 were ignored for iteration 2. The effect on the attribute completeness can be seen in appendix B table B.2. The most significant changes were in the attributes maintrade category and no of emp comp with an increase of 11% from iteration 1. The currency has increased to 0.61 from 0.46. Since the distribution of the starting year is skewed we can also look at how the median currency have changed. The median for the currency has increased from 0.5 to 0.58.

5.1.3 Iteration 3

For iteration 3 all the continuous attributes were discretized by binning by clusters. The data quality with regards to attribute completeness and currency has not been affected but noise in the continuous attributes is smoothed out as a result of the discretization (Han et al., 2012).

5.1.4 Iteration 4

For iteration 4 the attribute completeness was increased even more than for previous iterations. Two columns were ignored, Maintrade Category and No. of Emp. Comp., which gave us a higher relation completeness for the data.

5.1.5 Iteration 5

In iteration 5 we also removed the missing values in the attributes Maintrade Category and No. of Emp. Comp. but in a different manner. Instead of ignoring the attributes we removed the tuples that have missing values in both the attributes. The increased attribute completeness is seen in table B.3 in appendix B.

5.2 Mining Results

In this section the results of the different mining iterations is presented and later on analyzed. The algorithm configurations and preparations have been described in chapter 4. For each iteration the result will be presented in the form of decision trees, lift charts and performance.

5.2.1 Iteration 1

As described in chapter 4 the first iteration has only been subject to the initial preparation. All figures for iteration 1 are presented in appendix C.
Configuration A

For configuration A where we have used the default settings for the algorithm, the decision tree has four levels, and the most likely churners are found under the internal node named Mrkt Part Avg < 0.2. This means that companies with a participation average per year less than 0.2 are likely to churn. If their average is < 0.02 they are very likely to churn. Another interesting observation is the node named Mrkt Part Avg >= 2. If companies have 2 or more in average participation in marketing events they are not very likely to churn.

Configuration B

The decision tree for configuration B has five levels and is more complex than for configuration A. The most likely churners are found in the path Start Year < 2004 and Salesperson = 232. This means that companies that were acquired before 2004 and the responsible salesperson is 232 are very likely to churn. We can also see that customers in the node named Start Year >= 2012 are not very likely to churn. Another observation is that four of the nodes are representing attributes with and without missing values. This means that the missing values in the attribute Maintrade Category affects the result from the mining model.

Configuration C

In the decision tree for configuration C there are five levels and about the same complexity as for configuration B. The most likely churners are found in the path Start Year < 2004 and No Of Products < 1. Old customers with no additional products for their CRM-system have a high probability to churn regardless of whether the Turnover attribute is missing or not. In level 4, the splitting criteria for most of the nodes depends on whether the attribute Turnover is missing or not. This does not seem to have an significant effect on churn since the amount of churners does not vary much between the branches.

Lift Chart

The lift chart for iteration 1 can also be seen in appendix C. The lift for configuration A is higher than for configuration B and C. The lift curve for configuration A follows the ideal model up to 40% of the overall population. Configuration B and C have almost the same lift for the whole population. With 50 % of the overall population configuration A can predict about 90% of the churners and configuration B and C about 70%. That gives us a lift of 40% for configuration A and about 20% for configuration B and C. In the mining legend in C we can also see these figures. The score for configuration A is the highest (0.96) and lowest for configuration B (0.84).
Performance

In table 5.1 the performance of the classifier for iteration 1 is shown. Regarding the accuracy, configuration A is the best one with 89% accuracy. Configuration B and C have similar accuracy around 70%. The error rate is calculated by $1 - \text{accuracy}$ and therefore configuration A also has the lowest error rate. Configuration B and C has about 30% error rate, which means that 30% of the tuples are incorrectly classified. The sensitivity measures the true positive recognition rate. That is how many of the churners were classified as churner compared to the total amount of churners in the data (Han et al., 2012). The sensitivity is 84% for configuration A, 70% for configuration B and 66% for configuration C. The specificity measures how well the classifier classifies non churners (Han et al., 2012). Configuration A has the best specificity with 94%, significantly higher than configuration B with 72% and configuration C with 78%. Regarding the precision which measures the percentage of positive classified tuples that actually are positive, configuration A is outperforming the other configurations with 93%.

<table>
<thead>
<tr>
<th>Config.</th>
<th>Accuracy</th>
<th>Error rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.89</td>
<td>0.11</td>
<td>0.84</td>
<td>0.94</td>
<td>0.93</td>
</tr>
<tr>
<td>B</td>
<td>0.71</td>
<td>0.29</td>
<td>0.70</td>
<td>0.72</td>
<td>0.72</td>
</tr>
<tr>
<td>C</td>
<td>0.72</td>
<td>0.28</td>
<td>0.66</td>
<td>0.78</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Cross Validation

In table C.2 a summation of the cross validation for iteration 1 is shown. The partition size for this iteration is 300 tuples. For configuration A the average number of true positives are 42%. This is significantly higher than for configuration B and C which both have an average of 35%. The same trend is also seen for the true negative tuples where A classifies 48%, B 36% and C 37% correctly. For the false positives and false negatives configuration A misclassified 3% and 8 % respectively, B 14% and 15% and C 13% and 15%. As we can see, configuration B and C have almost identical numbers for the cross validation. The performance for the cross validation is presented in table 5.2 where we can see that all the performance metrics for the cross validation is similar to the performance when using a test set for evaluation.
Table 5.2: Performance with cross validation for Iteration 1

<table>
<thead>
<tr>
<th>Config</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.89</td>
<td>0.83</td>
<td>0.95</td>
<td>0.94</td>
</tr>
<tr>
<td>B</td>
<td>0.71</td>
<td>0.70</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>C</td>
<td>0.72</td>
<td>0.71</td>
<td>0.74</td>
<td>0.73</td>
</tr>
</tbody>
</table>

5.2.2 Iteration 2

For the second iteration all the tuples with $\text{Start Year} < 2004$ were ignored. All the graphics for iteration 2 can be seen in appendix D.

Configuration A

The decision tree for configuration A tells us almost the same thing as the one for configuration A in iteration 1. The tree has 5 levels and the most likely churners are found where $\text{Mrkt Part Avg} < 0.2$. The sub-tree under the node also has high probability for churn, regardless if it is a very old customer or not. We can also see as in iteration 1 that customers with a higher participation average are not very likely to churn regardless of how many invitations to events they get.

Configuration B

For configuration B we can clearly see that the tree is much less complex than for iteration 1 with the same configuration. The tree does now only have 3 levels and the most likely churners are found in the path $\text{Start Year} < 2007$ and $\text{Maintrade Category} = \text{Missing}$ and $\text{Start Year} \geq 2007$ and $\text{Start Year} < 2010$ and $\text{No Of Emp Comp} = '0 anställda' \text{ (eng: employees)}$. Like in iteration 1 we have more likely churners in very old customers where the $\text{Maintrade category}$ is missing and in customers of customer type E with 0 employees. Customer type E means that the customer has bought a small amount of licenses.

Configuration C

The decision tree for configuration C is also less complex than its corresponding tree for iteration 1. The most likely churners are found in the pattern $\text{Start Year} < 2007$ and $\text{Cust Type} = 'E'$ and $\text{No Of Emp Comp} = \text{Missing}$. Another pattern where customers are likely to churn, is the one that we previously identified for configuration B where $\text{Start Year} \geq 2007$ and $\text{Start Year} < 2010$ and $\text{No. Of. Emp. Comp.} = '0 employees'$. In the sub-tree of $\text{Cust. Type} = E$ we can see that the model is dependent on
missing values in the attribute No. Of Emp. Comp.. Overall the decision trees for iteration 2 is less complex than the trees in iteration 1.

**Lift Chart**

The lift chart for iteration 2 looks almost like the chart from iteration 1. With help from the mining legend we can see that it scores a bit lower and reaches a target population of a couple percentages less than the models in iteration 1.

<table>
<thead>
<tr>
<th>Config.</th>
<th>Accuracy</th>
<th>Error rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.86</td>
<td>0.14</td>
<td>0.7</td>
<td>0.98</td>
<td>0.95</td>
</tr>
<tr>
<td>B</td>
<td>0.69</td>
<td>0.31</td>
<td>0.48</td>
<td>0.84</td>
<td>0.68</td>
</tr>
<tr>
<td>C</td>
<td>0.7</td>
<td>0.3</td>
<td>0.68</td>
<td>0.72</td>
<td>0.64</td>
</tr>
</tbody>
</table>

**Performance**

The performance for iteration 2 can be seen in table 5.3. The accuracy for iteration 2 is slightly lower than for iteration 1 and configuration A is performing best with 86% accuracy. Configuration B and C have 69% and 70% respectively in accuracy. The sensitivity is 70% for configuration A, 48% for B and 68% for configuration C. Regarding the specificity, it has increased for iteration 2 except for configuration C where it has decreased by 6 percent. This means that the classification with configuration A and B has become better in classifying non-churners. The precision has increased to 95% for configuration A. For configuration B and C it has decreased to 68% and 64% respectively compared to iteration 1.

**Cross Validation**

Table D.2 shows the results from the cross validation of iteration 2. The size of the partitions in this iteration has decreased to 228 as a result of the removal of tuples with Start year < 2004. Configuration A managed to identify on average 31% as true positives. The corresponding number for configuration B and C is 20% and 27%. Configuration A has a low number of false positives, only 2% on average. Configuration B and C have a few more with 9% and 10% respectively. For the number of true negatives configuration A also has the highest number with 56% on average. The performance for the cross validation is seen in table 5.4 overall the
accuracy for the cross validation is slightly higher than the accuracy with a test and training set. For configuration C we can see that the accuracy, specificity and precision has increased with 5, 11 and 9 percent respectively. The sensitivity has decreased with 5 percent.

Table 5.4: Performance with cross validation for Iteration 2

<table>
<thead>
<tr>
<th>Config</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.87</td>
<td>0.74</td>
<td>0.96</td>
<td>0.93</td>
</tr>
<tr>
<td>B</td>
<td>0.69</td>
<td>0.47</td>
<td>0.85</td>
<td>0.69</td>
</tr>
<tr>
<td>C</td>
<td>0.75</td>
<td>0.63</td>
<td>0.83</td>
<td>0.73</td>
</tr>
</tbody>
</table>

5.2.3 Iteration 3

In iteration 3 the data preparation is the same as in iteration 2 but in addition all continuous attributes are discretized by clusters. The graphics are found in appendix E.

Configuration A

The result for iteration 3 with configuration A is similar to the results of iteration 1 and 2. The tree has 4 levels and the most likely churners are those companies with low average of marketing participation.

Configuration B

The result from configuration B is identical to the one from iteration 2.

Configuration C

For configuration C there are differences between iteration 2 and 3. We can see that the binning of the attribute Turnover affects the tree. The second level of the tree is the same as for iteration 2 but when looking at the third level in the decision tree we can see that customers in the bin called \( \geq 248004 \ldots \) are likely to churn. We can also see that old customers with lower turnover than previously described and that are of customer type E are likely to churn.

Lift Chart

The lift chart for iteration 3 is found in appendix E. The chart is very similar to iteration 2 but with slightly higher lift for configuration A and
B and slightly lower for configuration C as seen in the mining legend. The overall score for the models is higher for each of the configurations compared to iteration 2 but lower for all configuration compared to iteration 1.

Performance

The performance for iteration 3 is shown in table 5.5. Configuration A has an accuracy of 86% while configurations B and C only reach about 70%. Regarding the sensitivity, configuration A also scores the best with 81% here as well. 81% is significantly higher than the sensitivity for configuration B which reaches 49%. Configuration C performs between A and B with 64%. The specificity for configuration A is 90%, which again is better than configuration B and C. For the last measure, the precision, configuration A reaches 86%, configuration B 72%, and configuration C 67%. The precision is about 20% better for configuration A than configuration C.

Overall, configuration A scores the best on all measures. Configuration C is performing better than configuration B when classifying churners but configuration B is better in classifying non-churners.

Table 5.5: Performance Iteration 3

<table>
<thead>
<tr>
<th>Config.</th>
<th>Accuracy</th>
<th>Error rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.86</td>
<td>0.14</td>
<td>0.81</td>
<td>0.9</td>
<td>0.86</td>
</tr>
<tr>
<td>B</td>
<td>0.7</td>
<td>0.3</td>
<td>0.49</td>
<td>0.86</td>
<td>0.72</td>
</tr>
<tr>
<td>C</td>
<td>0.71</td>
<td>0.29</td>
<td>0.64</td>
<td>0.76</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Cross Validation

In table E.2 the cross validation for iteration 3 is shown. The partition size is 229.3 on average which is about the same as for iteration 2 since no further tuples have been discriminated. Configuration A identified 33% true positive tuples while configuration B only identified 19% and configuration C 26%. For the true negatives configuration A identified 54%, B 49% and C 46%. Configuration A performed best for the false positives and false negatives with 4% and 9% respectively. Configuration B misclassified 9% and 22%, and configuration C misclassified 12% and 16%. Regarding the performance of the cross validation in table 5.6 we can see that the accuracy is about the same for the cross validation and the evaluation with the test set. The sensitivity is a bit lower and the specificity higher for the cross validation, except for the specificity for configuration B where it is 2% lower.
### 5.2. MINING RESULTS

#### CHAPTER 5. RESULT & ANALYSIS

#### 5.2. MINING RESULTS

Table 5.6: Performance with cross validation for Iteration 3

<table>
<thead>
<tr>
<th>Config</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.87</td>
<td>0.79</td>
<td>0.93</td>
<td>0.88</td>
</tr>
<tr>
<td>B</td>
<td>0.69</td>
<td>0.47</td>
<td>0.84</td>
<td>0.68</td>
</tr>
<tr>
<td>C</td>
<td>0.72</td>
<td>0.61</td>
<td>0.79</td>
<td>0.67</td>
</tr>
</tbody>
</table>

#### 5.2.4 Iteration 4

For iteration 4 the attributes *Maintrade Category* and *No Of Emp Comp* were ignored and customers with *Start Year* < 2004 excluded from the mining. The graphics are displayed in appendix F.

**Configuration A**

As in previous iteration there has been no major difference in the decision tree for configuration A. Customers with low marketing participation are more likely to churn than customers with high marketing participation. One difference is that the patterns with the attribute *Mrkt Invi Avg* that describes the average number of marketing invitations is no longer a predictor for churn.

**Configuration B**

For configuration B that had patterns with missing values in its decision tree for the previous iterations, we can now see that the most likely churners are located on the path *Start Year* < 2007 and *Cust Type = E*. This means that relatively old customers that have a low number of licenses are likely to churn. If they are not customer type E, they are likely customers of type D.

**Configuration C**

The result in configuration C have changed a lot compared to the previous iteration. It now shows the pattern that customers with low marketing participation are more likely to churn regardless of the starting year as seen in the level 5 leaf node. This decision tree is actually identical to the decision tree for iteration 2 with configuration A.

**Lift Chart**

In the lift chart in Appendix F we can see that configuration C has increased its lift for all populations. The lift during the previous iteration was about
20% but is now at 35% and more similar to the lift curve of configuration A than B. The lift curve for configuration A and B are pretty much the same with a similar score and target population as in previous iterations.

Performance

Table 5.7 shows the performance for iteration 4. For all measures configuration A and C score exactly the same. The accuracy for A and C is 86% while it is 70% for configuration B. The sensitivity for A and C is 82% and the specificity 90%. For configuration B the sensitivity is 69% and the specificity 70%. Configuration A and C also outperforms B when it comes to precision. B has a precision of 62% while A and C have 85%. Overall configuration A and C performs better than configuration B for iteration 4.

<table>
<thead>
<tr>
<th>Config.</th>
<th>Accuracy</th>
<th>Error rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.86</td>
<td>0.14</td>
<td>0.82</td>
<td>0.9</td>
<td>0.85</td>
</tr>
<tr>
<td>B</td>
<td>0.7</td>
<td>0.3</td>
<td>0.69</td>
<td>0.7</td>
<td>0.62</td>
</tr>
<tr>
<td>C</td>
<td>0.86</td>
<td>0.14</td>
<td>0.82</td>
<td>0.9</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Cross Validation

The cross validation for iteration 4 is presented in table F.2. The average partition size is 226.9 where configuration A managed to identify 35% true positives. Configuration B managed to identify 24% and configuration C 35% on average. Configuration A and C identified 52% true negatives and configuration B 43%. The level of misclassified tuples is low in configuration A and C with 6% false positives for both. Configuration A misclassified 8% as false negatives and configuration C 7%. The performance for the cross validation is shown in table 5.8 where we can see that the performance is about the same for configuration A and C with the cross validation. The cross validation has a slightly higher precision for configuration A and C compared to when using a test set for the evaluation. Regarding configuration B we can see that all the performance metrics except for the specificity are lower for the cross validation.

5.2.5 Iteration 5

For the last iteration the attributes Maintrade Category and No Of Emp Comp are no longer ignored but the tuples that have missing values in both
Table 5.8: Performance with cross validation for Iteration 4

<table>
<thead>
<tr>
<th>Config</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.87</td>
<td>0.82</td>
<td>0.90</td>
<td>0.86</td>
</tr>
<tr>
<td>B</td>
<td>0.66</td>
<td>0.56</td>
<td>0.73</td>
<td>0.60</td>
</tr>
<tr>
<td>C</td>
<td>0.87</td>
<td>0.82</td>
<td>0.90</td>
<td>0.86</td>
</tr>
</tbody>
</table>

these attributes are excluded. As in previous iteration the customers with Start Year < 2004 are excluded. The graphics are displayed in appendix G.

Configuration A

For configuration A the decision tree is less complex than for earlier iterations and the patterns are now only depending on the attribute Mrkt Invi Avg. Customers with a low level of invitations from the marketing departments are more likely to churn according to the tree.

Configuration B

In the decision tree for configuration B we can see that companies with 0 employees that were acquired before 2010 are more likely to churn. Companies that has more than 0 employees and were acquired after 2010 are not very likely to churn.

Configuration C

In the final execution with configuration C we can see the pattern that has been occurring through the iterations that customers that have Mrkt Part Avg < 0.2 are more likely to churn than customers with higher participation average. The tree is not very complex with only three levels and six nodes.

Lift Chart

Looking at the lift chart for the last iteration we can see the same trend as for iteration 4. Configuration A and C follow each other close to the ideal model with about 35% in lift for 50% of the population and they both have high overall score. Configuration B is still performing with a lift around 18% for 50% of the overall population and has the lowest overall score.

Performance

Table 5.9 shows the performance for iteration 5. Configuration A has the best accuracy, almost 90%, which is significantly better than for configura-
tion B with 67%. Configuration C performs slightly below configuration A with 86%. Configuration A and C have a very high specificity with 98% and 97%. The sensitivity has decreased for all configurations compared to iteration 4 and is now 75% for configuration A, 23% for configuration B and 68% for configuration C. In comparison with iteration 4, the specificity and precision has increased for all configurations.

Table 5.9: Performance Iteration 5

<table>
<thead>
<tr>
<th>Config.</th>
<th>Accuracy</th>
<th>Error rate</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.89</td>
<td>0.11</td>
<td>0.75</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td>B</td>
<td>0.67</td>
<td>0.33</td>
<td>0.23</td>
<td>0.95</td>
<td>0.72</td>
</tr>
<tr>
<td>C</td>
<td>0.86</td>
<td>0.14</td>
<td>0.68</td>
<td>0.97</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Cross Validation

The cross validation is shown in table G.2. For iteration 5 the partition size has shrunk to 168.7 due to the removal of tuples including missing values in the Maintrade category and No of emp comp attribute. We can see in the table that the number of churners has shrunk as a result of the removed tuple and the number of identified true positive tuples is 26% for configuration A, 27% for configuration C and only 10% for configuration B. The number of classified true negatives are 63% for configuration A, 61% for B and 63% for configuration C. Configuration A has 1% false positives and 10% false negatives. B has higher percentage of false positives and false negatives with 3 and 26 percent respectively. Configuration C falls between A and B with 2% false positives and 11% false negatives. Comparing the performance of the validation with a test set and with cross validation (seen in table 5.10) we can see that all configurations have a very high specificity. This means that they are very accurate when predicting non-churners. For configuration B the accuracy, sensitivity and precision is slightly higher when using cross validation.

5.3 Analysis

This section describes, compares and analyzes the results from the iterations.
Table 5.10: Performance with cross validation for Iteration 5

<table>
<thead>
<tr>
<th>Config</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.89</td>
<td>0.73</td>
<td>0.98</td>
<td>0.95</td>
</tr>
<tr>
<td>B</td>
<td>0.71</td>
<td>0.27</td>
<td>0.95</td>
<td>0.76</td>
</tr>
<tr>
<td>C</td>
<td>0.88</td>
<td>0.70</td>
<td>0.97</td>
<td>0.94</td>
</tr>
</tbody>
</table>

5.3.1 Iteration 1, 2, and 3

Looking at iteration 1, 2 and 3 (appendix C, D, and E) the decision trees for configuration B and C, some nodes are split on whether an attribute value is missing or not. For example, in figure C.3 the leaves in level five are *Maintrade category not = Missing* and *Maintrade category = Missing*. A missing value in this attribute tells us that a company has no retrieved information from PAR, which does not provide so much interesting information for Lundalogik regarding churn.

With this in mind we now look at the decision trees for the different configurations for iteration 1, 2 and 3. The configuration without splits depending on if a value is missing or not is configuration A. The decision tree for configuration A tells us that the most likely churners have *Mrkt Part Avg < 0.2*. This means that companies that do not get involved with the marketing events arranged by Lundalogik are more likely to churn. Although this might be an interesting result, Lundalogik has not always been recording their marketing events in their CRM-system. This means that the low participation average might be due to old customers not having any marketing participation registered.

But if we change the point of view to "Which customers are not very likely to churn?" we can see that customers that have a marketing participation average $\geq 2$ are not very likely to churn. This result might be even more important since it tells Lundalogik that if the customers get involved and want to improve their ways of working with the CRM-system the customer are more likely to stay.

Performance

For all three iterations configuration A have the highest accuracy, about 15% higher than B and C. Iteration 1 has a slightly higher accuracy than iteration 2 and 3 for all configurations which means that using all tuples makes the classification more accurate. Regarding the sensitivity for configuration A, iteration 2 have a significantly lower value than for iteration 1 and 3. For configuration B iteration 1 performs 20% higher than iteration 2 and 3 while configuration C performs consistent over all three iterations. In
contradiction to the sensitivity, the specificity for configuration B is significantly lower in iteration 1 than in 2 and 3. This means that configuration B in iteration 1 is better at recognizing churners but worse at recognizing non-churners than iteration 2 and 3. The precision for configuration A is lower in iteration 3 than the other two iterations. In iteration 1 and 2 the precision for configuration A is respectively 93% and 95% showing that in these two iterations almost all tuples classified as churners are churners. For configuration B and C the precision is highest in iteration 1, lower in iteration 2, and then it is higher in iteration 3 again even though not as high as for iteration 1. In iteration 1 there are more tuples for the algorithm to learn from, then it is reduced in iteration 2. The binning in iteration 3 seems to have a some positive effect on the precision for configuration B and C since it increases again from iteration 2.

**Number of Employees Company**

The attribute *No. of Emp. Comp.* is often decisive in the decision trees for iteration 1, 2 and 3. When drilling down into the mining structure to examine the tuples that support the rule we find that when *No. of Emp. Comp = '0 employees'* often indicates that the company has declared bankruptcy, been part of a corporate fusion or disused in some way. This is actually an interesting result for Lundalogik, since it tells us that many companies churn because of previously mentioned reasons, which is natural and out of Lundalogik’s possibility to influence.

### 5.3.2 Iteration 4 and 5

In iteration 4 and 5 the dependence on whether an attribute is missing or not is reduced. In iteration 4, the attributes *Maintrade Category* and *No. of Emp. Comp.* are ignored and in iteration 5 tuples where these two attributes are missing are ignored. Ignoring the attributes in iteration 4 seems to have little effect on the decision tree for configuration A. It is, as for the previous iterations, dependent on the number of marketing events a customer has taken part in. A difference from previous iterations is that configuration C also is mainly dependent on *Mrkt Part Avg*. Looking at the lift chart (appendix F) we can see that the curves for configuration A and C are very similar. In table 5.7 we also notice that A and C have the exact same values for all measures. This is due to the reduction in input attributes which leaves the feature selection in configuration C uninvoked. Configuration B on the other hand results in a completely different decision tree (appendix F). Here, the *Start Year* and *Cust Type* are decisive for the classification. *Cust Type* E and D are small customers with few licences. The ones most likely to churn are the ones with *StartYear < 2007* and *CustType = E*, which means that they are small customers that have been customers for a long period of time. This is also the tree with the highest
5.3. ANALYSIS

CHAPTER 5. RESULT & ANALYSIS

predict probability. It is significantly higher with 40% than for configuration A and C with 19% and 17% respectively.

Removing tuples in iteration 5 results in fewer tuples for the algorithm to learn and therefore the decision trees in that iteration are smaller. As in iteration 4, configuration A and C depends only on marketing participated (appendix G). The curves in the lift chart for these two configurations are very similar, as well as the values in the mining legend. The decision tree for configuration B is different. Here, No. of Emp. Comp. and Start Year are the decisive attributes. Looking at the mining legend, configuration B has the highest predict probability. Compared to iteration 4 all values are lower which might be due to the fact that the number of rows in iteration 5 is fewer.

Performance

As can be seen in table 5.7 and table 5.9 in the previous section, configuration B has lower values for all measures in both iteration 4 and 5 compared to the other configurations. The accuracy for configuration B is about the same for iteration 4 and 5 with a slight advantage for iteration 4. Although the accuracy is similar between the iterations the sensitivity for iteration 5 is very low, only 23%. This means that configuration B in iteration 5 is bad in predicting churners and it highlights the importance of using sensitivity and specificity to be able to determine whether the classifier is good in predicting the positive or negative tuples. As we can see when comparing the specificity for iteration 4 and 5 the latter is much better in predicting non-churners. With the business objectives for the analysis in mind, the performance for predicting churn is better when attributes with a lot of missing values are ignored (iteration 4) compared to when removing the tuples that has a lot of missing values in iteration 5.

5.3.3 Comparing Evaluation Methods

Looking at the cross validation compared to the training and test set there are a few observations to be mentioned. Overall, the accuracy for the performance using training and test set and cross validation are very similar. It differs only a few percent between the iterations for each configuration. As well as when training and test set are used, configuration A has the highest accuracy, configuration C the next highest, and configuration B the lowest.

Regarding the sensitivity, the performance between the two different evaluation methods is about the same except for configuration B in iteration 4. The sensitivity when using a test set for validation is 13 percent higher than for the cross validation.

For configuration C in iteration 2 the accuracy, specificity and precision is lower and the sensitivity is higher when using a test set compared to when using the cross validation.
These deviations might be a result of the selection of data when using a training and test set. The cross validation performance is calculated on the average of 10 folds and should be less affected by the data selection.

Except for these previous mentioned deviations the results from both evaluation methods are very similar and can be used to interpret the performance of the classifiers. Lundalogik should consider whether they can afford to use 30% of the data as a test set compared to the cross validation where they can use all the data for training and testing. They should also consider the performance issues that can occur in the case when using cross validation for large data sets.
Chapter 6

Discussion

This chapter will discuss the results from the case study. It also includes comments about the validity of the results, method critic, recommendations for future work, and a final conclusion.

6.1 Significance of the Result

This section discusses the significance of the results from our analysis.

6.1.1 Data Preparations

Our data preparation operations have tried to achieve the highest possible attribute completeness and currency by cleaning incorrect data, removing noise, excluding old customers, ignoring attributes etc. This has been done with decreasing the number of tuples (companies) in the data warehouse as the major set-back. Our initial data warehouse consisted of 5223 tuples and after the preparations in iteration 5 the data warehouse consisted of only 2929 tuples, which is only 56% of the original data set.

The major reason for noise, inconsistency and missing values in the data at Lundalogik is that the purpose of the data collection is not data mining. The process of updating or inserting new entries varies among employees which results in differently updated data. [Gunnarsson et al.] (2007) mentions this as one major learning from their work in the newspaper industry. Standardizing the data collection for the purpose of data mining by controlling the input and updates in the system makes it easier to sort and analyze the data.

6.1.2 Data Mining

The decision trees show Lundalogik what attributes and patterns that cause churn. The trees needs to be evaluated and interpreted to extract that infor-
mation. If the tree cannot be interpreted in a way that provides information it is not of much use even if it performs well when evaluated with evaluation metrics. Therefore, when using the results for finding potential churners both the tree itself and its performance need to be taken into consideration. From our results we can also conclude that the configuration of the algorithm has noticeable effects on both the decision trees and performance. Finding the optimal configuration of the algorithm is outside our scope but from the three configurations we have used it is evident that it has a significant effect on the results.

There is no data preparation that is overall considerably better than the others. In the first iteration no restrictions are included and the number of rows is the highest, i.e the algorithm has more tuples to learn from. When the data is reduced in iteration 2, 3, and 4 by removing all customers with \textit{Startyear} < 2004 the algorithm has 29\% fewer tuples to learn from. Further, in iteration 5 where the data is reduced once again the algorithm has 36\% less tuples to learn from than in iteration 1. According to Domingos [2012] a poor algorithm with plenty of data is better than a good algorithm with little data. This suggests that the performance would be better the more tuples we have for the algorithm to learn from.

When selecting which measurements to consider for evaluation the business objectives are of importance. If we want to find churners the sensitivity, the true positive recognition rate, might be more interesting than the specificity, the true negative recognition rate. For configuration B, the sensitivity is highest in iteration 1 and 4. In iteration 2 and 3 the sensitivity is 48\% and 49\% respectively. In iteration 4 the sensitivity increases to 60\%. The precision is the percentage of positively classified tuples that actually are positive, which also is interesting if we want to find churners. The precision for iteration 4 is 62\%, a little lower than 68\% for iteration 2 and 72\% for iteration 3. Even though the precision is a little lower for iteration 4, this indicates that it is better to ignore columns with a lot of missing values when it comes to finding churners when the number of tuples to learn from is equal for the iterations.

Configuration A, where no feature selection is used, has the highest performance for all iterations. Looking at the lift charts, we can see that the curves are very close to the ideal model. When feature selection is used in configuration B and C the curves in the lift chart are closer to the random model and the performance is significantly lower. This is interesting since feature selection is used to find the most interesting attributes. It indicates that when interestingness in the attributes is calculated the predictions are not as good as when it is not. Another indication might be that even though the performance of the predictive model is not as good as with feature selection, the patterns are more interesting from an information gain perspective. This might be a symptom of overfitting in configuration A. The model seems to be very specified and its performance is extremely good. Since the training and test set are randomly selected from the data we can assume that
they are homogeneous and therefore the algorithm performs well on the test set. When feature selection is invoked in configuration B and C to prevent overfitting, the performance seems more realistic except for configuration C in iteration 4 and 5. In iteration 4, two attributes were ignored, which resulted in that configuration A’s and C’s performance were identical. As it seems, the feature selection for configuration C is not invoked when we removed the attributes. Feature selection is used to counteract overfitting and when it is not invoked for configuration C in iteration 4 the model seems to be overfitted.

With previous discussion in mind, there are a few things to consider when selecting which preparation to use. First, there is a fine balance between the number of tuples and the data quality. The more tuples that are included the ratio of old-to-new customers are higher. Old customers have more missing values and are more obsolete. The data saved for these customers are not the same as the data saved today. On the other hand, the algorithm has fewer tuples to learn from. Ballings and Van den Poel (2012) found that they could reduce their data set with 69% with only a small decrease in predictive performance. The biggest difference between their work and ours is that they had an initial data set of almost 130 000 customers. Our results for iteration 4 shows that when reducing the data set with 29% there is only a small loss of predictive performance for configuration A (4.25% in average) and B (3.5% in average). For configuration C all performance metrics increased compared to iteration 1 (13% in average).

Next, which attributes to include affects the results. Iteration 4 shows that ignoring attributes with a lot of missing values affects the decision trees and results. Including attributes with missing values results in a few splitting criteria that splits depending on if the attribute is missing or not. Drawing conclusions from such a pattern might not be accurate.

Lastly, the algorithm configuration affects the results significantly. In our study, using feature selection results in lower performance even though the interestingness in the attributes in the decision trees is higher.

6.1.3 Lundalogik

Traits and behaviors of customers that affect churn are of great importance for Lundalogik in the process of preventing it. Therefore, the decision trees are highly relevant, since they are the ones that show these attributes. They can reveal new interesting traits or confirm the theories for churn Lundalogik have today.

Overall, the decision trees confirmed some of the theories Lundalogik had before this master thesis. They suspected that the length of the relation affects churn, which is supported by configuration B. As mentioned by Ballings and Van den Poel (2012), the time that has passed since the initial purchase affects churn which it seems to do here as well. An interesting observation is that, in contradiction to Ballings and Van den Poel
who found that customers with a long relationship to the company are more likely to be loyal, our results suggests the opposite. Further, our results reveal that the number of licenses, which often is correlated with the size of the customer, affects churn. Companies with few employees or few licences are more likely to churn. This can be related to the monetary variable mentioned by Coussement et al. (2014) since the money invested in only a few licenses is not a large sum. None of these findings were suprising to Lundalogik.

The patterns for configuration A revealed that customers with low averages of participation in marketing events are more likely to churn. It also revealed that customers with a higher average of participation are not very likely to churn. This was new information for Lundalogik, they did not suspect that this was a predictor for churn. This shows that the commitment and effort to improve the usage of LIME Easy decrease the probability of churn.

What also needs to be taken into consideration by Lundalogik when evaluating the decision trees except for the attributes is the splitting criteria. For example, in tree 1C (appendix C figure C.4) eight internal nodes, or splitting criteria, depend on whether an attribute has a value or not. That is not a trait or behavior of a customer but more a question regarding the completeness of the database. Drawing conclusions from that information is surely misleading.

6.2 Method Critique

In this section method critique for our master thesis is presented.

6.2.1 Construct Validity

The people who criticise case studies says that case study practitioners often fail to establish operational measures and that subjective judgement is used to collect the data (Yin, 2003). For this thesis we have tried to increase the construct validity by clearly defining what our view of data quality is and how we measure it. The measures are extracted from literature about data quality and other studies. For the evaluation of classifiers the metrics have been extracted from well known literature about data mining such as the book by Han et al. (2012). Although the literature about data mining has the same metrics, the interpretation of the metrics can be rather specific for each context. High precision does not always guarantee that the result is good. Because of this reason we have tried to motivate all the conclusions we have made from the performance metrics to give the reader an opportunity to follow our interpretation of the result.
6.2.2 External Validity

For a case study to have external validity the study’s finding should be generalizable to other studies as well and critics state that single cases often are a poor basis for generalization (Yin, 2003). Our study has not been tested on other databases or companies and it is therefore not known if it is generalizable or not. By striving to have a high reliability we believe that the study and its methods can be used on other similar databases but whether the results will be the same or not remains unknown.

6.2.3 Reliability

The purpose of reliability is to make sure that if another investigator would like to follow the same procedure that they would arrive at the same findings and conclusion as we did (Yin, 2003). Our main working methodology has been CRISP-DM that is a highly recognized methodology for data mining (Mariscal et al., 2010). All operations performed during our preparation of the data have been documented in this report as well as how it has modified the properties of the data.

6.3 Future Work

We think this is a very interesting area with great potential for Lundalogik to continue working with. There is much more to be done in the area than what has been accomplished by this master thesis. First of all invoice data can be incorporated in the analysis. With invoice data you can see exactly how much money each customer has spent will be evident to give more attention to the monetary variable mentioned by Coussement et al. (2014).

Next, Lundalogik should make sure to have complete data on all customers. For example, up until just recently, the field service enddate for customers leaving the company were not saved. This means that there is no easy way to find out when a customer actually churned. For some attributes that we have used there are a lot of missing values. A few splitting criteria depend on whether a value is available or not which does not tell Lundalogik that much about the customer. By updating the data completeness the analysis should generate more informative patterns.

For our thesis a lot of data has been excluded since it was badly updated and an incorrect representation of the real-world. This problem stems from that the data has not been collected for the purpose of predictive analytics (or other analytics for that matter), which is a common problem highlighted by Gunnarsson et al. (2007). Lundalogik should investigate what data that would further improve the churn prediction and make sure that that data is collected. As mentioned by Gunnarsson et al. (2007) it is important to have appropriate data to be able to execute a credible data mining project. If Lundalogik know what data that is needed for churn prediction the sys-
tem and processes can be constructed to make sure that the right data is collected with high quality when entered in the system. As it is today, every sales person has their own way of updating LIME which makes the data inconsistent. If the updates were more consistent it would make the data more consistent. We believe that introducing the term data quality when collecting the data rather than during the analysis is more relevant, but this area needs to be further researched.

Configuring the algorithm to improve the results has been outside of our scope for this thesis. That is an area though, were Lundalogik can continue working. The algorithm is very powerful and can be configured in several different ways. As we can see from our thesis, by invoking feature selection the decision trees contain more information.

6.4 Conclusion

Churn prediction is an interesting area with great potential for Lundalogik. The results from this study confirms some of the theories about churn that Lundalogik had without support from the data:

- Old customers with low number of licenses are more likely to churn.
- Customers tend to churn when they are part of a corporate fusion.

We also found patterns that were not known by Lundalogik:

- Customers with low average in participation during marketing events are more likely to churn.
- Customers that are involved in marketing events and educational seminars are not very likely to churn.

Regarding the different preparation methods used we cannot conclude that one is better than another for the result of the analysis. Although we can conclude that it is better to ignore attributes rather than removing the tuples with missing values when the data set is small with both the decision trees and performance in consideration. Our result seems to be more related to the configurations rather than the preparations.

With the three different configurations we have used different levels of feature selection. Configuration A used no feature selection, configuration B and C used different feature selection parameters where B was most strict. Our conclusion is that feature selection is useful since it separates the interesting attributes from the uninteresting. The algorithm then use the most interesting one for the analysis depending on how strict the feature selection was. Also, feature selection is used to prevent overfitting, which results in more realistic analyses.
The decision trees reveal some interesting attributes related with churn, even though the performance needs to be considered when drawing conclusions from them.

During the iterations with the different preparation methods we can conclude that our definition of data quality has been highly correlated with the preparation methods. Although this is very dependent on how the data quality is measured. If the quality had been evaluated by some other metrics the conclusion might not be the same.

The main conclusion for this thesis is regarding data. Our analysis is performed on data that is not collected with the purpose of predictive analytics. The data is collected to support the employees at Lundalogik in their work and their contact with the customers. For Lundalogik to be successful with their use of predictive analytics the data collection should both support the business processes and the intended analytics.
Bibliography


K. Jensen. Crisp-dm process diagram. No changes made, 2012. URL http://creativecommons.org/licenses/by-sa/3.0/legalcode


### Data at Lundalogik

Table A.1: Company table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idcompany</td>
<td>int</td>
<td>Nominal</td>
<td>Every company has a unique id in the table.</td>
</tr>
<tr>
<td>createduser</td>
<td>int</td>
<td>Nominal</td>
<td>Id of the user who created the company.</td>
</tr>
<tr>
<td>createdtime</td>
<td>datetime</td>
<td>Date</td>
<td>The time the company was created in the table.</td>
</tr>
<tr>
<td>updateduser</td>
<td>int</td>
<td>Nominal</td>
<td>Id of the user who did the last update.</td>
</tr>
<tr>
<td>timestamp</td>
<td>datetime</td>
<td>Date</td>
<td>Date for the last update.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar</td>
<td>Nominal</td>
<td>Contains the name of the company.</td>
</tr>
<tr>
<td>buyingstatus</td>
<td>int</td>
<td>Nominal</td>
<td>A company can be kund, koncernkund, prospekt, and slutat anvanda</td>
</tr>
<tr>
<td>customertype</td>
<td>int</td>
<td>Nominal</td>
<td>Categorization of the customer A,B,C,D,E. Contains the string id and the actual string needs to be selected from the string table.</td>
</tr>
<tr>
<td>servicestatus</td>
<td>int</td>
<td>Nominal</td>
<td>Describes the status of the service contract.</td>
</tr>
<tr>
<td>serviceenddate</td>
<td>datetime</td>
<td>Date</td>
<td>The end date for the service contract.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>idlicense</td>
<td>int</td>
<td>Nominal</td>
<td>Unique identifier for a license.</td>
</tr>
<tr>
<td>createduser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who created the license.</td>
</tr>
<tr>
<td>createdtime</td>
<td>datetime</td>
<td>Date</td>
<td>Date when tuple was created.</td>
</tr>
<tr>
<td>updateuser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who last updated the tuple.</td>
</tr>
<tr>
<td>timestamp</td>
<td>datetime</td>
<td>Date</td>
<td>Time of the last update.</td>
</tr>
<tr>
<td>productlevel1</td>
<td>int</td>
<td>Nominal</td>
<td>Describes the product package the license has.</td>
</tr>
<tr>
<td>startdate</td>
<td>datetime</td>
<td>Date</td>
<td>Start date of the license. When the license became active.</td>
</tr>
<tr>
<td>enddate</td>
<td>datetime</td>
<td>Date</td>
<td>End date of the license. When the license will become inactive. This will be NULL if the customer does not upgrade or end its contract.</td>
</tr>
<tr>
<td>quantity</td>
<td>int</td>
<td>Numeric</td>
<td>The number of licenses the customer has bought.</td>
</tr>
<tr>
<td>type</td>
<td>int</td>
<td>Nominal</td>
<td>Type of license.</td>
</tr>
<tr>
<td>company</td>
<td>int</td>
<td>Nominal</td>
<td>The ID of the company that holds the license.</td>
</tr>
<tr>
<td>productlevel2</td>
<td>int</td>
<td>Nominal</td>
<td>Describes the product type in the productlevel1 package.</td>
</tr>
<tr>
<td>active</td>
<td>int</td>
<td>Binary</td>
<td>Describes if the license is active or not.</td>
</tr>
<tr>
<td>serviceno</td>
<td>nvarchar</td>
<td>Nominal</td>
<td>The ID of the service contract.</td>
</tr>
</tbody>
</table>
Table A.3: History table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idhistory</td>
<td>int</td>
<td>Nominal</td>
<td>Unique identifier for the history tuple.</td>
</tr>
<tr>
<td>createduser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who created the history tuple.</td>
</tr>
<tr>
<td>createdtime</td>
<td>datetime</td>
<td>Date</td>
<td>Date when history entry was created.</td>
</tr>
<tr>
<td>updateuser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who last updated the tuple.</td>
</tr>
<tr>
<td>timestamp</td>
<td>datetime</td>
<td>Date</td>
<td>Time of the last update.</td>
</tr>
<tr>
<td>type</td>
<td>int</td>
<td>Nominal</td>
<td>The type of the history event.</td>
</tr>
<tr>
<td>date</td>
<td>datetime</td>
<td>Date</td>
<td>The date the history event occurred.</td>
</tr>
<tr>
<td>company</td>
<td>int</td>
<td>Nominal</td>
<td>The company the history event corresponds to.</td>
</tr>
<tr>
<td>person</td>
<td>int</td>
<td>Nominal</td>
<td>The person that was subject to the history event.</td>
</tr>
<tr>
<td>sos</td>
<td>int</td>
<td>Nominal</td>
<td>If the history event is a customer problem or support errand this is the ID of the sos errand.</td>
</tr>
</tbody>
</table>
Table A.4: Product table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idproduct</td>
<td>int</td>
<td>Nominal</td>
<td>Unique identifier for the product.</td>
</tr>
<tr>
<td>createduser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who created the product.</td>
</tr>
<tr>
<td>createdtime</td>
<td>datetime</td>
<td>Date</td>
<td>Date the product was created in the database.</td>
</tr>
<tr>
<td>updateuser</td>
<td>int</td>
<td>Nominal</td>
<td>ID of the user who made the last update.</td>
</tr>
<tr>
<td>timestamp</td>
<td>datetime</td>
<td>Date</td>
<td>Date of the last update.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar</td>
<td>Nominal</td>
<td>Contains the name of the product.</td>
</tr>
<tr>
<td>parent</td>
<td>int</td>
<td>Nominal</td>
<td>The parent of the product.</td>
</tr>
<tr>
<td>path</td>
<td>nvarchar</td>
<td>Nominal</td>
<td>Contains the path to the product, i.e. the parent and the product.</td>
</tr>
<tr>
<td>level</td>
<td>int</td>
<td>Ordinal</td>
<td>Products are divided into two levels which is described in this field.</td>
</tr>
<tr>
<td>salestatus</td>
<td>int</td>
<td>Nominal</td>
<td>Describes the sale status of the product.</td>
</tr>
</tbody>
</table>
Appendix B

Attribute Completeness

Table B.1: Attribute Completeness for Iteration 1

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Null Count</th>
<th>AC</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer type</td>
<td>2</td>
<td>100%</td>
<td>4%</td>
</tr>
<tr>
<td>maintrade category</td>
<td>1921</td>
<td>58%</td>
<td>6%</td>
</tr>
<tr>
<td>no of employees</td>
<td>1682</td>
<td>63%</td>
<td>6%</td>
</tr>
<tr>
<td>postal area</td>
<td>93</td>
<td>98%</td>
<td>1%</td>
</tr>
<tr>
<td>salesperson</td>
<td>77</td>
<td>98%</td>
<td>11%</td>
</tr>
<tr>
<td>turnover</td>
<td>1343</td>
<td>71%</td>
<td>5%</td>
</tr>
</tbody>
</table>
### Table B.2: Attribute Completeness for Iteration 2

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Null Count</th>
<th>AC</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer type</td>
<td>1</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>maintrade category</td>
<td>1000</td>
<td>69%</td>
<td>11%</td>
</tr>
<tr>
<td>no of employees</td>
<td>855</td>
<td>74%</td>
<td>11%</td>
</tr>
<tr>
<td>postal area</td>
<td>58</td>
<td>98%</td>
<td>0%</td>
</tr>
<tr>
<td>salesperson</td>
<td>77</td>
<td>98%</td>
<td>0%</td>
</tr>
<tr>
<td>turnover</td>
<td>847</td>
<td>74%</td>
<td>3%</td>
</tr>
</tbody>
</table>

### Table B.3: Attribute Completeness for Iteration 5

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Null Count</th>
<th>AC</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer type</td>
<td>1</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>maintrade category</td>
<td>254</td>
<td>91%</td>
<td>22%</td>
</tr>
<tr>
<td>no of employees</td>
<td>15</td>
<td>100%</td>
<td>26%</td>
</tr>
<tr>
<td>postal area</td>
<td>8</td>
<td>100%</td>
<td>2%</td>
</tr>
<tr>
<td>salesperson</td>
<td>32</td>
<td>99%</td>
<td>1%</td>
</tr>
<tr>
<td>turnover</td>
<td>445</td>
<td>85%</td>
<td>11%</td>
</tr>
</tbody>
</table>
## Appendix C

### Iteration 1

Table C.1: Mining Legend Iteration 1

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Score</th>
<th>Target Population</th>
<th>Predict Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.96</td>
<td>87.37</td>
<td>31.74</td>
</tr>
<tr>
<td>B</td>
<td>0.84</td>
<td>70.73</td>
<td>38.46</td>
</tr>
<tr>
<td>C</td>
<td>0.86</td>
<td>70.44</td>
<td>47.78</td>
</tr>
<tr>
<td>Random Guess Model</td>
<td></td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Ideal Model</td>
<td></td>
<td>98.2</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C. ITERATION 1

Figure C.1: Lift Chart Iteration 1

Figure C.2: Decision Tree Iteration 1 A
APPENDIX C. ITERATION 1

Figure C.3: Decision Tree Iteration 1 B

Figure C.4: Decision Tree Iteration 1 C
Table C.2: Cross validation for Iteration 1

<table>
<thead>
<tr>
<th>Config</th>
<th>Partition Size</th>
<th>Measure</th>
<th>Average</th>
<th>Percentage</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>300</td>
<td>True Positive</td>
<td>124.61</td>
<td>42</td>
<td>5.08</td>
</tr>
<tr>
<td>A</td>
<td>300</td>
<td>False Positive</td>
<td>7.50</td>
<td>3</td>
<td>2.62</td>
</tr>
<tr>
<td>A</td>
<td>300</td>
<td>True Negative</td>
<td>143.20</td>
<td>48</td>
<td>2.60</td>
</tr>
<tr>
<td>A</td>
<td>300</td>
<td>False Negative</td>
<td>24.69</td>
<td>8</td>
<td>4.84</td>
</tr>
<tr>
<td>B</td>
<td>300</td>
<td>True Positive</td>
<td>104.10</td>
<td>35</td>
<td>5.20</td>
</tr>
<tr>
<td>B</td>
<td>300</td>
<td>False Positive</td>
<td>43.10</td>
<td>14</td>
<td>4.25</td>
</tr>
<tr>
<td>B</td>
<td>300</td>
<td>True Negative</td>
<td>107.60</td>
<td>36</td>
<td>4.08</td>
</tr>
<tr>
<td>B</td>
<td>300</td>
<td>False Negative</td>
<td>45.20</td>
<td>15</td>
<td>5.17</td>
</tr>
<tr>
<td>C</td>
<td>300</td>
<td>True Positive</td>
<td>105.58</td>
<td>35</td>
<td>9.42</td>
</tr>
<tr>
<td>C</td>
<td>300</td>
<td>False Positive</td>
<td>39.50</td>
<td>13</td>
<td>14.52</td>
</tr>
<tr>
<td>C</td>
<td>300</td>
<td>True Negative</td>
<td>111.20</td>
<td>37</td>
<td>14.52</td>
</tr>
<tr>
<td>C</td>
<td>300</td>
<td>False Negative</td>
<td>43.72</td>
<td>15</td>
<td>9.62</td>
</tr>
</tbody>
</table>
Appendix D

Iteration 2

Table D.1: Mining Legend Iteration 2

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Score</th>
<th>Target Population</th>
<th>Predict Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0,93</td>
<td>84,02</td>
<td>17,38</td>
</tr>
<tr>
<td>B</td>
<td>0,79</td>
<td>67,31</td>
<td>38,54</td>
</tr>
<tr>
<td>C</td>
<td>0,8</td>
<td>72,4</td>
<td>34,52</td>
</tr>
<tr>
<td>Random Guess Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ideal Model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure D.1: Lift Chart Iteration 2

Figure D.2: Decision Tree Iteration 2 A
APPENDIX D. ITERATION 2

Figure D.3: Decision Tree Iteration 2 B

Figure D.4: Decision Tree Iteration 2 C
Table D.2: Cross validation for Iteration 2

<table>
<thead>
<tr>
<th>Config</th>
<th>Partition Size</th>
<th>Measure</th>
<th>Average</th>
<th>Percentage</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>228.2</td>
<td>True Positive</td>
<td>71.10</td>
<td>31</td>
<td>3.33</td>
</tr>
<tr>
<td>A</td>
<td>228.2</td>
<td>False Positive</td>
<td>5.40</td>
<td>2</td>
<td>3.55</td>
</tr>
<tr>
<td>A</td>
<td>228.2</td>
<td>True Negative</td>
<td>127.20</td>
<td>56</td>
<td>3.76</td>
</tr>
<tr>
<td>A</td>
<td>228.2</td>
<td>False Negative</td>
<td>24.50</td>
<td>11</td>
<td>3.23</td>
</tr>
<tr>
<td>B</td>
<td>228.2</td>
<td>True Positive</td>
<td>44.60</td>
<td>20</td>
<td>3.82</td>
</tr>
<tr>
<td>B</td>
<td>228.2</td>
<td>False Positive</td>
<td>20.49</td>
<td>9</td>
<td>5.05</td>
</tr>
<tr>
<td>B</td>
<td>228.2</td>
<td>True Negative</td>
<td>112.11</td>
<td>49</td>
<td>5.23</td>
</tr>
<tr>
<td>B</td>
<td>228.2</td>
<td>False Negative</td>
<td>51.00</td>
<td>22</td>
<td>3.82</td>
</tr>
<tr>
<td>C</td>
<td>228.2</td>
<td>True Positive</td>
<td>60.60</td>
<td>27</td>
<td>8.82</td>
</tr>
<tr>
<td>C</td>
<td>228.2</td>
<td>False Positive</td>
<td>22.77</td>
<td>10</td>
<td>15.06</td>
</tr>
<tr>
<td>C</td>
<td>228.2</td>
<td>True Negative</td>
<td>109.84</td>
<td>48</td>
<td>15.38</td>
</tr>
<tr>
<td>C</td>
<td>228.2</td>
<td>False Negative</td>
<td>35.00</td>
<td>15</td>
<td>8.77</td>
</tr>
</tbody>
</table>
Table E.1: Mining Legend Iteration 3

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Score</th>
<th>Target Population</th>
<th>Predict Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.94</td>
<td>85.61</td>
<td>17.89</td>
</tr>
<tr>
<td>B</td>
<td>0.81</td>
<td>68.82</td>
<td>38.95</td>
</tr>
<tr>
<td>C</td>
<td>0.82</td>
<td>71.22</td>
<td>35.71</td>
</tr>
<tr>
<td>Random Guess Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ideal Model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure E.1: Lift Chart Iteration 3

Figure E.2: Decision Tree Iteration 3 A
Figure E.3: Decision Tree Iteration 3 B

Figure E.4: Decision Tree Iteration 3 C
### Table E.2: Cross validation for Iteration 3

<table>
<thead>
<tr>
<th>Config</th>
<th>Partition Size</th>
<th>Measure</th>
<th>Average</th>
<th>Percentage</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>229.3</td>
<td>True Positive</td>
<td>75.40</td>
<td>33</td>
<td>6.44</td>
</tr>
<tr>
<td>A</td>
<td>229.3</td>
<td>False Positive</td>
<td>9.91</td>
<td>4</td>
<td>6.12</td>
</tr>
<tr>
<td>A</td>
<td>229.3</td>
<td>True Negative</td>
<td>124.19</td>
<td>54</td>
<td>6.01</td>
</tr>
<tr>
<td>A</td>
<td>229.3</td>
<td>False Negative</td>
<td>19.80</td>
<td>9</td>
<td>6.30</td>
</tr>
<tr>
<td>B</td>
<td>229.3</td>
<td>True Positive</td>
<td>44.50</td>
<td>19</td>
<td>5.92</td>
</tr>
<tr>
<td>B</td>
<td>229.3</td>
<td>False Positive</td>
<td>21.00</td>
<td>9</td>
<td>5.64</td>
</tr>
<tr>
<td>B</td>
<td>229.3</td>
<td>True Negative</td>
<td>113.10</td>
<td>49</td>
<td>5.54</td>
</tr>
<tr>
<td>B</td>
<td>229.3</td>
<td>False Negative</td>
<td>50.70</td>
<td>22</td>
<td>5.98</td>
</tr>
<tr>
<td>C</td>
<td>229.3</td>
<td>True Positive</td>
<td>58.51</td>
<td>26</td>
<td>8.40</td>
</tr>
<tr>
<td>C</td>
<td>229.3</td>
<td>False Positive</td>
<td>28.51</td>
<td>12</td>
<td>9.01</td>
</tr>
<tr>
<td>C</td>
<td>229.3</td>
<td>True Negative</td>
<td>105.59</td>
<td>46</td>
<td>8.95</td>
</tr>
<tr>
<td>C</td>
<td>229.3</td>
<td>False Negative</td>
<td>36.69</td>
<td>16</td>
<td>8.30</td>
</tr>
</tbody>
</table>
Appendix F

Iteration 4

Table F.1: Mining Legend Iteration 4

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Score</th>
<th>Target Population</th>
<th>Predict Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.93</td>
<td>85.3</td>
<td>19</td>
</tr>
<tr>
<td>B</td>
<td>0.8</td>
<td>71.57</td>
<td>40.08</td>
</tr>
<tr>
<td>C</td>
<td>0.94</td>
<td>85.54</td>
<td>17.42</td>
</tr>
<tr>
<td>Random Guess Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ideal Model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure F.1: Lift Chart Iteration 4

Figure F.2: Decision Tree Iteration 4 A
Figure F.3: Decision Tree Iteration 4 B

Figure F.4: Decision Tree Iteration 4 C
Table F.2: Cross validation for Iteration 4

<table>
<thead>
<tr>
<th>Config</th>
<th>Partition Size</th>
<th>Measure</th>
<th>Average</th>
<th>Percentage</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>226.9</td>
<td>True Positive</td>
<td>78.29</td>
<td>35</td>
<td>4.41</td>
</tr>
<tr>
<td>A</td>
<td>226.9</td>
<td>False Positive</td>
<td>12.81</td>
<td>6</td>
<td>3.43</td>
</tr>
<tr>
<td>A</td>
<td>226.9</td>
<td>True Negative</td>
<td>118.69</td>
<td>52</td>
<td>3.13</td>
</tr>
<tr>
<td>A</td>
<td>226.9</td>
<td>False Negative</td>
<td>17.11</td>
<td>8</td>
<td>4.66</td>
</tr>
<tr>
<td>B</td>
<td>226.9</td>
<td>True Positive</td>
<td>53.51</td>
<td>24</td>
<td>11.44</td>
</tr>
<tr>
<td>B</td>
<td>226.9</td>
<td>False Positive</td>
<td>35.02</td>
<td>15</td>
<td>10.69</td>
</tr>
<tr>
<td>B</td>
<td>226.9</td>
<td>True Negative</td>
<td>96.48</td>
<td>43</td>
<td>10.47</td>
</tr>
<tr>
<td>B</td>
<td>226.9</td>
<td>False Negative</td>
<td>41.89</td>
<td>18</td>
<td>11.37</td>
</tr>
<tr>
<td>C</td>
<td>226.9</td>
<td>True Positive</td>
<td>78.39</td>
<td>35</td>
<td>4.48</td>
</tr>
<tr>
<td>C</td>
<td>226.9</td>
<td>False Positive</td>
<td>12.91</td>
<td>6</td>
<td>3.45</td>
</tr>
<tr>
<td>C</td>
<td>226.9</td>
<td>True Negative</td>
<td>118.60</td>
<td>52</td>
<td>3.17</td>
</tr>
<tr>
<td>C</td>
<td>226.9</td>
<td>False Negative</td>
<td>17.01</td>
<td>7</td>
<td>4.74</td>
</tr>
</tbody>
</table>
## Appendix G

### Iteration 5

Table G.1: Mining Legend Iteration 5

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Score</th>
<th>Target Population</th>
<th>Predict Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.94</td>
<td>88.53</td>
<td>10.2</td>
</tr>
<tr>
<td>B</td>
<td>0.76</td>
<td>68.1</td>
<td>36.44</td>
</tr>
<tr>
<td>C</td>
<td>0.92</td>
<td>86.38</td>
<td>11</td>
</tr>
<tr>
<td>Random Guess Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ideal Model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX G. ITERATION 5

Figure G.1: Lift Chart Iteration 5

Figure G.2: Decision Tree Iteration 5 A

Figure G.3: Decision Tree Iteration 5 B

98
APPENDIX G. ITERATION 5

Figure G.4: Decision Tree Iteration 5 C

Table G.2: Cross validation for Iteration 5

<table>
<thead>
<tr>
<th>Config</th>
<th>Partition Size</th>
<th>Measure</th>
<th>Average</th>
<th>Percentage</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>168.7</td>
<td>True Positive</td>
<td>43.50</td>
<td>26</td>
<td>3.27</td>
</tr>
<tr>
<td>A</td>
<td>168.7</td>
<td>False Positive</td>
<td>2.20</td>
<td>1</td>
<td>1.08</td>
</tr>
<tr>
<td>A</td>
<td>168.7</td>
<td>True Negative</td>
<td>106.60</td>
<td>63</td>
<td>0.92</td>
</tr>
<tr>
<td>A</td>
<td>168.7</td>
<td>False Negative</td>
<td>16.40</td>
<td>10</td>
<td>3.32</td>
</tr>
<tr>
<td>B</td>
<td>168.7</td>
<td>True Positive</td>
<td>16.30</td>
<td>10</td>
<td>2.79</td>
</tr>
<tr>
<td>B</td>
<td>168.7</td>
<td>False Positive</td>
<td>5.20</td>
<td>3</td>
<td>1.78</td>
</tr>
<tr>
<td>B</td>
<td>168.7</td>
<td>True Negative</td>
<td>103.60</td>
<td>61</td>
<td>1.85</td>
</tr>
<tr>
<td>B</td>
<td>168.7</td>
<td>False Negative</td>
<td>43.60</td>
<td>26</td>
<td>2.84</td>
</tr>
<tr>
<td>C</td>
<td>168.7</td>
<td>True Positive</td>
<td>41.70</td>
<td>25</td>
<td>3.23</td>
</tr>
<tr>
<td>C</td>
<td>168.7</td>
<td>False Positive</td>
<td>2.80</td>
<td>2</td>
<td>1.16</td>
</tr>
<tr>
<td>C</td>
<td>168.7</td>
<td>True Negative</td>
<td>106.00</td>
<td>63</td>
<td>1.18</td>
</tr>
<tr>
<td>C</td>
<td>168.7</td>
<td>False Negative</td>
<td>18.20</td>
<td>11</td>
<td>3.16</td>
</tr>
</tbody>
</table>
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsmann i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är krävande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet – or its possible replacement – for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

©David Buö and Magnus Kjellander