Customer Churn Prediction for the Icelandic Mobile Telephony Market

Emilía Huong Xuan Nguyen

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
University of Iceland
2011

Customer Churn Prediction for the Icelandic Mobile Telephony Market

Emilía Huong Xuan Nguyen

60 ECTS thesis submitted in partial fulfillment of a
Magister Scientiarum degree in Mechanical Engineering

Advisor(s)
Prof. Tómas Philip Rúnarsson
Ólafur Magnússon

Faculty Representative
Prof. Birgis Hrafnkelsson

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, September 2011

Customer Churn Prediction for the Icelandic Mobile Telephony Market

60 ECTS thesis submitted in partial fulfillment of a *Magister Scientiarum* degree in Mechanical Engineering

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Hjarðarhagi 2-6
107, Reykjavik
Iceland

Telephone: 525 4000

Bibliographic information:
Emilía Huong Xuan Nguyen, 2011, *Customer Churn Prediction for the Icelandic Mobile Telephony Market*, Master's thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland.

Printing: Háskólaprent, Fálkagata 2, 107 Reykjavík
Reykjavik, Iceland, September 2011

Abstract

In 2010, the penetration of the Icelandic mobile telephony market has reached about 120%. Competition is fierce in such a highly saturated market. Customers become more and more demanding on price and service. New regulations and technologies allow them to switch easily between mobile operators. As the result, customer churn has increased significantly. Facing this challenge, mobile operators shift their attention from customer acquisition to customer retention. The crucial elements of customer retention are accurate churn prediction models and effective churn prevention strategies. The goal of this study is to construct a churn prediction model that can output the probabilities that customers will churn in the near future. Churn prediction is formulated as a classification task of churners and non-churners. Learning algorithms are applied on training data to build classifiers. The data is a set of customers where each one is represented by numerous features and labeled as churner or non-churner. The primarily step involves employing feature selection to search for relevant features, eliminate irrelevant or redundant ones. Afterwards, the reduced data with only relevant features are passed into classifiers trained using machine learning algorithms: (1) C4.5 decision tree, (2) alternating decision tree, (3) Naïve Bayes and (4) logistic regression. The result of this study is twofold. Firstly, churn indicators are identified and insights are provided into churn behavior in postpaid and prepaid sectors for the time period from July 2010 to May 2011. Secondly, churn can be forecasted with a certain accuracy in advance which enables the mobile operator to carry out suitable reactions.

Keywords: mobile telephony market, churns prediction, feature selection, machine learning, data mining.

Útdráttur

Árið 2010, fjöldi farsíma áskrifta á íslenska fjarskiptamarkað hefur náð um 120% af íbúafjölda landsins. Samkeppni er hörð í slíkum mettuðum markaði. Viðskiptavinir verða meira og meira krefjandi varðandi verði og þjónustu. Nýjar reglur og nýr tækni leyfa þeim að skipta auðveldlega milli farsímafyrirtækja. Þess vegna hefur flutningur viðskiptavina í farsímaþjónustu milli samkeppnisaðila aukist verulega. Standa frammi fyrir þeirri áskorun, farsímafyrirtæki hafa fært athygli þeirra frá öflun viðskiptavina til viðskiptavina varðveislu. Mikilvægir þættir í varðveislu viðskiptavina eru nákvæm brottfallsspá og áhrifarík markaðsáætlun. Markmið þessa verkefnis er að byggja brottfallsspárlíkan sem gefur líkurnar á því að viðskiptavinir muni hætta í náinni framtíð. Brottfallsspá er leyst sem flokkunarverkefni. Flokkarar eru þjálfaðar samkvæmt reikniritum með sögulegum gögnum. Gögnin eru í formi safn af viðskiptavinum þar sem hverjum og einum er lýst með fjölmörgum breytum. Fyrsta skref felst í því að framkvæma breytuval til að finna helstu vísbendingar um mögulegt brottfall. Síðan eru marktækar breytur settar í flokkunarvélum þjálfaðar með machine learning aðferðunum: (1) ákvörðunartré, (2) Naïve Bayes og (3) logistic aðhvarfsgreiningu. Niðurstöður þessara rannsókna fela í tveimur þáttum. Í fyrsta lagi, vísbendingar um mögulegt brottfall eru fundnar sem veita innsýn í brotföllin í áskrifta- og frelsi þjónustu á tímabilinu frá Júlí 2010 til Maí 2011. Í öðru lagi geta líkönin spáð fyrir um brottföll fyrirfram með ákveðin nákvæmni sem gerir fyrirtæki kleift að framkvæma viðeigandi viðbrögð.

Lykilorð: fjarskiptamarkaður, brottfallsspárlíkan, breytuval, vélrænn lærdómur, gagnanám.

Table of Contents

List of Figures

List of Tables

Abbreviations

SVMs	support vectors machine
ANN	artificial neural network
Logistic	logistic regression
ADtree	alternating decision tree
ID3	ID3 decision tree
C4.5	C4.5 decision tree
J48tree	commercial version of C4.5 decision tree in WEKA
CFS	correlation-based feature selection
LVF	filter version of Las Vegas algorithm
ROC	receiver operating characteristics graph
wr-J48tree	J48tree decision tree based wrapper
wr-ADtree	alternating decision tree based wrapper
wr-Bayes	Naïve Bayes based wrapper
wr-logistic	logistic regression based wrapper

Notations

$\mathbf{x}$	an instance
y	the class
y_j	class label j of the class
$\mathbb{C}$	set of all class labels
$\mathbf{x}^{(k)}$	feature vector associated with instance k
$y^{(k)}$	class label associated with instance k
x_i	feature i of an instance
$\{x_{i1}, x_{i2}, \ldots, x_{im}\}$	possible outcomes of feature i

w_i	weight of feature i
$x_i^{(k)}$	value of feature i of instance k
d	number of features
n	number of instances
c	number of classes
D	a data set
S	a feature subset
S_{best}	the best feature subset according to some metrics
F	a feature
v	value of feature F
f	feature selection evaluation function
TP	a true positive
TN	a true negative
FN	a false negative
FP	a false positive
P	the total number of positive instances in the data set
N	the total number of negative instances in the data set
OA	overall accuracy
AUC	area under ROC curve
OCL	overall confidence level
cf	confidence factor for J48tree
min	minimum number of instances per node for C4.5 tree
nb	number of boosting iterations for ADtree

Acknowledgements

Finally the exciting, challenging and instructive graduate years have now come to an end. I'm especially grateful to those who have accompanied with me through this memorable journey.

First of all, I deeply thank Prof. Tómas Philip Rúnarsson for introducing me to the magical world of Machine Learning and for supervising this research. His advice, guidance, suggestion and above all his infectious enthusiasm have encouraged me to complete this research.

I would like to sincerely thank Ólafur Magnússon for his advice, support and patience. He has provided me information and practical assistances on the behalf of Síminn. I would also like to thank all BI colleges: Björgvin, Svana, Bjarni and Gunnar for their help and advices, and for creating a friendly and fun atmosphere at work.

Special thanks to Ýmir Vigfússon and Leon Danon for introducing me to the exciting world of social network analysis and for helping me with the network features extraction.

My deepest gratitude goes to my family. I can't thank them enough for their unconditional love, endless support and constant encouragement.

This research was supported by Síminn, for which I am very grateful.

Reykjavík, 12. July 2011.

Emilía Huong Xuan Nguyen

1 Introduction

Customers are the center of all focuses in almost every industry that offers products and services. Successful business practitioners are those who understand their customers, know what they want and fulfill their needs. Insights into customer minds are gained through marketing research and business experiences. In addition to these traditional sources, valuable knowledge is also hidden in the company's in-house database that contains massive information about customers, their subscriptions, transactions and purchases. This study attempts to mine the database of an Icelandic mobile operator in a search for relevant information that can be utilized for churn prediction. This study is carried out for Síminn. The data is extracted from the data warehouse of Síminn.

1.1 Motivation

The telecommunication industry in the last years is characteristic of rapid changes, liberalization of market, technical innovations, saturation and intense competition. Customers have enough of alternatives to select from when it comes to the decision of choosing among mobile operators. The fact that they can switch between operators without any difficulty at any time has encouraged customers to be constantly in search for better services at lower cost. Churn is a term used in the telecommunication and many other industries and refers to customers' decision to move their subscription from one service provider to another (Berson, Smith, & Thearling, 2000). Churn is caused by several common reasons such as dissatisfaction with the services and high bills. In addition, customers often receive attractive offers when signing up with a new mobile operator. It is also a well-known fact that a churn customer influences his acquaintances in the same network to churn as well. The average churn in a mobile operator is about 2% per month (Berson, Smith, & Thearling, 2000). Yearly churn rate in Europe is 25%, in US 37% and Asia 48% (Mattersion, 2001).

Since the GSM system was first launched in Iceland in 1994 (Siminn, 2011), the mobile telephony market in the country has never stopped growing. Today the number of mobile subscriptions is about 120% of the population (Icelandic Post- and Telecom Administration, 2011). The year 1998 marked a historical milestone when the monopoly in the Icelandic telecommunications services was abolished and the market was opened for competition (Gislason, 2005). Legislation and technology innovation embrace the environment for fierce competition in the mobile market. It has never been easier for customers to switch between mobile operators. Number portability was enabled since 2004. In addition, operators don't charge for new subscription and no lock-in period is required (Nordic National Regulatory Authorities, 2005). Mobile operators in the forms of mobile network operators, mobile virtual network operators and service providers, all compete for customers. To date, the primary operators in the market are Síminn, Vodafone, Nova and Tal. Figure 1-1 illustrates their market shares in post- and prepaid separately in the time period from 2004 to 2010. A postpaid subscription includes a contract between a customer and the mobile operator. The customer receives periodically bills which claim him to pay a certain monthly fee plus his excess usage. In contrasts, a prepaid subscription doesn't include any contract. The customer has to fill his calling credit beforehand and he is able to use the service only within the limit of his calling credit.

Being dominant in the mobile market for many years, the two oldest and largest mobile operators in Iceland, Síminn and Vodafone are now facing decline in their market shares. Significant downswing can be observed in both post- and prepaid after the entry of Nova in the market in 2007. Nova gained nearly 7% market shares in the postpaid sector only in one year from 2007 to 2008. Thereafter the postpaid market has been quite stable. At the same time, a more dramatic change has taken place in prepaid sector. In three years, Nova has managed to expand increasingly and become dominant in prepaid market with 36% market shares, following by Síminn with 32% and Vodafone 28%.

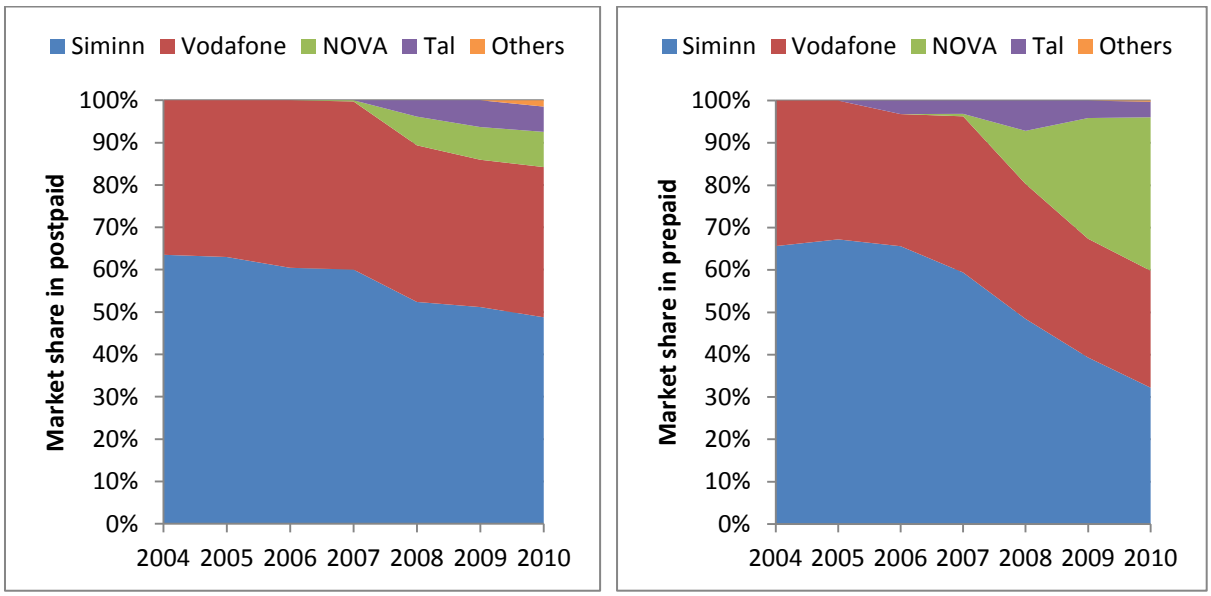

Figure 1-1. Market shares of Icelandic mobile operators in post- and prepaid from 2004 to 2010

Loss of customers equals the loss of future revenue plus loss of initial investment made to acquire those customers. Finding and securing new customers becomes more difficult and costly more than ever due to the intense competition and saturation of the market. The cost of acquiring new customers is considerably higher than keeping current customers (Wei & Chiu, 2002). Facing this challenge, operators need to focus on how to prevent churn. Churn management involves making necessary reactions in order to retain customers who are at the risk of leaving. For example by offering them better services and deals. However, companies don't have enough time and resources to contact to the entire customer base and not everyone needs attention at each time. By forecasting the customer's decision of moving to a competitor, churn prediction provides company information which makes them able to focus on a target group of customers with high risk of churn. Besides that the time aspect is also important. Knowing in advance before the customers really churn will give operator more room to react, and more chances of success in keeping customers. The more accurate the churn prediction is, the more money and resources the company will save.

Churn prediction modeling is an extensive task to deal with. Fortunately, both advanced technical solutions and theoretical expertise make it possible to accomplish the goal. High performance data warehouses and powerful business intelligence solutions enable experts to access, extract and manipulate a huge amount of data needed for churn modeling. Meaningful patterns and decision supportive information can be found in this huge amount of data using data mining methods and techniques. This study can be applied not only to the telecommunication industry but also in other domains with similar characteristic such as the insurance and banking industry.

1.2 Objectives and Contribution

The ultimate goal of this project is to construct a comprehensive solution consisting of a database and model for churn prediction in postpaid and prepaid markets. The desirable characteristics of churn prediction model are accuracy and complexity. Churn prediction is not a one-time only task but a repeated process that takes place in the dynamic business environment. A solution for churn prediction therefore needs to have the following properties:

(1) It can predict churn in advance with high accuracy, meaning that it can capture as high portion of churners as possible.
(2) It can score each customer with churn likelihood which denotes the probability that he will churn in the near future.
(3) It has reasonable implementation time so that model update and prediction can be made on a regular basis. The frequency is decided by the mobile operator which can be for example once in a month, in every two months or in each yearly quarter.
(4) It can be integrated smoothly into the business daily process.
(5) It can be improved and updated with ease in order to reflect potential changes in the market environment.

The objectives of this study is to establish the first foundation of churn prediction solution by answering the following questions:

i. Is it possible to predict churn? Which resources and techniques are needed to complete the task?
ii. Which features are useful for churn prediction?
iii. Which information the churn prediction models provide?

In this study, churn prediction is formulated as a classification task of churners and non-churners. Learning algorithms are applied on training data to build classifiers. The data is a set of customers where each one is represented by numerous features and labeled as churner or non-churner. The classifier is trained so that it can distinguish between churners and non-churners based on features associated with them. In order to achieve the objectives, the work performed can be divided into the following steps.

1. A framework for data acquisition is set up. The data used for churn modeling and prediction is generated from a special database. This database is defined by a relation model that connects different sources in the operator's data warehouse into one center table which contains customers and their associated features. The features belong to six categories: demographics, billing data, refill history, calling pattern, CDR billed and calling network features. The data is extracted from this database according to the timeline described in chapter 4.2.
2. The data is prepared for modeling and testing. The preprocessing step involves transferring the raw data from the data warehouse to the analysis platform. After the work of data transforming and sampling has been done, different feature selection methods are carried out to search for the most critical indicators that influence churn. The original data is dimensionally reduced by choosing only relevant features, eliminating those that are irrelevant or redundant. The results of feature selection reveal the relevancy of each feature category and each individual feature to the target concept which is churn.
3. Churn prediction models in the form of classifiers are built by employing four machine learning algorithms: C.45 decision trees, alternating decision tree, Naïve

Bayes and logistic regression. C4.5 is a decision tree builder using entropy for splitting nodes and adopting pruning to increase generality power. Alternating decision tree enhances the performance and interpretability of decision tree by adopting boosting techniques. Naïve Bayes is a probability approach based on Bayes's theorem. Logistic regression is a linear parametric model which outputs continuous probability value. The following primary churn indicators are underlined repeatedly by all classifiers: customer age, his rate plan and marital status, in which land area he lives, amount of calls and text messengers he receives, his out-net calls amount and expense. In addition, churners appear to be those who are well-connected to the social network by either having many neighbors, having well-connected neighbors or are well-connected themselves. The connectivity is measured by metrics come from social network analysis.

4. The best classifiers are chosen for post- and prepaid separately by making comparisons between different combinations of feature selection and learning algorithms. The classifiers' performances are evaluated according to three metrics: the overall accuracy, the area under ROC curve and prediction confidence level. All postpaid classifiers achieve above 60% overall accuracy on a testing set of 30 thousand instances. All prepaid classifiers achieve above 70% overall accuracy on a testing set of 40 thousand instances. These performances are comparable to the results achieved by prior researches on churn prediction.

1.3 Outline

The remaining of this thesis is structured as follows. Chapter 2 starts with a literature review of churn prediction in the telecommunications industry following by brief introduction of the data mining process. Chapter 3 provides theoretical background of feature selection methods, the learning algorithms and the evaluation metrics used in this study. Chapter 4 presents the work done in steps 1 and 2 as mentioned in chapter 1.2. It describes the data preparation process, the data itself and the features. The work done in steps 3 and 4 as given in chapter 1.2 is presented in chapter 5. It gives and discusses detailed results of model building along with model comparison. Chapter 6 sums up the research and suggests future work.

2 Application of Data Mining to Churn Prediction

Data mining (knowledge discovery) is an interdisciplinary field that involves the extraction of hidden predictive information from large databases (Thearling, 1999). For business purposes, data mining provides tools and techniques to search for meaningful pattern and decision support knowledge within the huge amount of raw data. It helps business practitioner either to confirm hypotheses or find new things in the data that have not been known, provides them valuable insight and competitive advantages (Witten & Frank, 2005). The process of data mining requires the cooperation of fields such as database system, data warehousing, machine learning, statistics. The first two fields take care of the data storage, data integration and access while the next two fields offer analytical tools to mine the data. This study focuses on the analytical part of data mining. How machine learning is applied to churn prediction in past and current researches is reviewed in chapter 2.1. Recently, new research direction employing social network analysis for churn prediction has risen along with the traditional machine learning approach. The later part of chapter 2.1 introduces several studies in this promising area and explains how social network analysis is utilized in this study to extract features. Chapter 2.2 goes through the process of a data mining project defined by the so-called CRISP-DM process model and relate it to the context of this study.

2.1 Churn Prediction in the Telecom Industry

Customer relationship management (CRM) is defined as "a comprehensive process of acquiring and retaining customers, with the help of business intelligence, to maximize the customer value to the organization" (Ngai, Xiu, & Chau, 2009). CRM framework can be divided distinctly into operational and analytical parts (He, Xu, Huang, & Deng, 2004). Operational CRM focuses on activities and processes concerning direct contact with customers such as marketing, sales and customer service. Analytical CRM focuses on data analysis to detect customer characteristics in order to support operational CRM. Churn management is a field under operational CRM which refers to the process of keeping the most profitable customers in subscription (Kentrias, 2001) and assessing the most effective way that an operator can react against churn (Hung, Yen, & Wang, 2006). On the other hand, churn prediction belongs to analytical CRM and consists of two goals: (1) explain why customers unsubscribe their subscription and move to a competitor, (2) predict which customers are most likely to churn in the near future. The final output of churn prediction is each customer's likelihood of churning, often called churn score. The likelihood can be used to rank customers descending from the one who is most likely to churn to those who are least likely. Mobile operators then decide to contact to the top x percent with the highest churn score and invite them suitable offers to keep them from churning. In addition, the mobile operators can integrate the churn scores together with customer value ranking. This collaboration provides the list of the most profitable customers who are at risk of churning and need to be contacted (SAP, 2011).

Over recent years, churn prediction has increasingly received attention of researchers. Studies focus on the search for methods and features that are the most effective in predicting churn.

The most popular methods that have been used for churn prediction are such as: decision tree, regression, Naïve Bayes and neural network. Assuming that changes in call patterns may include churn warning signals, (Wei & Chiu, 2002) use the call details to extract features that describe the changes in customers' calling patterns during a specific period. These features are passed into decision tree to build classifier. It is unavoidable that random selected data from mobile operator's database used for churn prediction has highly skewed class distribution. Churn is the class of interest but churn cases are many times fewer than non-churn cases. It follows that the classifier is biased towards the non-churn class and may predict all instances as non-churn to gain maximum accuracy. In order to solve this problem, (Wei & Chiu, 2002) adopt the multi-classifier class-combiner approach. The training set is divided into equal subsets. Each subset is used to train one base classifier. A meta-classifier combines outcomes of all base classifiers using weighted voting-based strategy gives the final prediction of a new instance. According to the results, the meta-classifier outperforms the single-classifier approach.

(Hung, Yen, & Wang, 2006) employ a decision tree for churn prediction in the Taiwan postpaid mobile market using several groups of features: customer demographics, billing information, contract/service status, call detail records, and service change log. Features are tested for significance using z-test. According to the analysis, the features that are significant to differentiate between churners and non-churners are: age, tenure, gender, billing amount, number of overdue payment, in-net call duration, number of changes in account information. Customers are then segmented with respect to the significant features. K-means clustering is used to segment the customers into five clusters according to their bill amount (indicates customer value), tenure (indicates customer loyalty) and usage (indicates customer activity). One decision tree is created for each cluster. A base decision tree is also created for all customers without any segmentation. The result shows no significant difference between the performances of decision trees with and without customer segmentation. Comparison of decision tree and artificial neural networks (ANN) verifies that ANN performs better on this particular data set. To make sure that the models work well, the authors track the models in one year from July 2001 to July 2002. The performances of all models remain first steady but then drop dramatically after six months from 80-90% to 0-10% of overall accuracy.

(Hadden, Tiwari, Roy, & Ruta, 2006) explore yet another source of features. They investigate the suitability of data containing customer complaints and repairs interactions with the operator for churn prediction. The most significant features are identified and the performances of ANN, classification tree and regression are compared. The findings reveal the pros and cons of each method. Classification tree achieves the highest overall accuracy while ANN has the lowest. However, ANN has the most churn cases correctly classified while regression has the most non-churn cases correctly classified.

Another black box alternative, besides ANN used in churn prediction is support vectors machine (SVMs). (Archaux, Laanaya, Martin, & Khenchaf, 2004) compare the application of SVMs and ANN to churn detection in a prepaid mobile operator. The database used in this study is composed of different types of data in 6 months: invoicing data, usage data, contractual data, data relating to subscriptions and cancellation of services, demographics and customer value. The training set contains 6000 cases. The results show that classifiers trained by SVMs and ANN perform equally well on a small testing set of 6000 cases. As the number of cases increased to 60000, SVMs outperforms ANN. The authors also examine the effect of different class distribution by varying the class ratio in the training set. They obtain the best results with the training set consisting of 50% churners and 50% non-churners.

Paying more attention on selecting the most significant features, (Yi & Guo-en, 2010) used SVM-RFE (Recursive feature selection based on support vector machine). Although SVMs are black boxes, linear SVMs return feature weights which indicates how important each feature is. Choosing features with highest weights to pass into SVMs classifier, the authors obtain an explainable prediction model or so as they claim. Performance of SVMs is found to be better than other common models such as decision tree, ANN, Bayes net, logistic regression and Naïve Bayes (Yi & Guo-en, 2010), (Guo-en & Wei-dong, 2008).

Churn prediction involves the search and identification of churn indicators. The most common approach is to consider each customer individually by studying their personal and business profile. Recent research opens a new approach in churn prediction by taking into account the social influence and studying customers' interactions. This new approach is based on the hypothesis that a churn customer influences other customers to churn as well. As pointed out by (Richter, Yom-Tov, & Slonim, 2010), the churn decision is formed by two aspects of reasoning: the social aspect and the economical aspect. A customer often belongs to one or several social groups such as a family, a group of friends or colleges. Within each group he is constantly receiving from and passing information to others, being influenced and being an influence. When a group member churns due to some dissatisfaction with the current mobile operator or a better deal from a competitor, this information is spread over the group, affecting other group members' opinion and action concerning churn. Another driving factor is that most customers are members of relatively strongly connected groups. Within each group the amount of calls is quite high between group members. Therefore, they prefer to maintain their subscriptions to the same service provider to enjoy the low calling rate inside mobile network.

Researches in this direction apply social network analysis where social relationships are viewed in terms of network theory. In a mobile network context, customers form a network which consists of two main elements: nodes and edges. Customers are considered as nodes, and edges are the connections between them. The weight on an edge indicates the strength of a connection. It can be quantified by call amount. High call amount implies strong connection between individuals and vice versa. When two nodes are connected directly, they are called neighbors or friends. The edge between them can be directed edge (points to only one direction) or undirected edge (points to both directions).

(Dasgupta, Singh, Viswanathan, Chakraborty, Mukheejea, & Nanavati, 2008) try to establish a relationship between the churn likelihood of a customer and the number of his friends that have already churned. The network of customers is modeled as a call graph where customers are nodes and their social connections in the form of phone calls are edges. The term friend refers to those who call each other during a certain period. The strength of their relationship is indicated by the call frequency and volume. (Dasgupta, Singh, Viswanathan, Chakraborty, Mukheejea, & Nanavati, 2008) propose a spreading activation-based approach where churn is spread over the network as a diffusion process. When some key individuals in the network churn, they influence their friends to churn, who in turn spread the churn epidemic to others, and so forth. The proposed model obtains a correct classification rate of 62% for churners. Furthermore, decision tree is tested by using three different groups of features. The usage features that are based purely on information extracted from CDR data. The connectivity features are based on the social ties of an individual with existing churners. And the interconnectivity features are derived from the structural ties between these churners. The results show that the decision tree that uses only usage features performs the worst. The decision tree uses connectivity features along with usage features performs slightly better. By

adding the interconnectivity features the performance of decision tree is significantly improved.

Following the social network analysis direction, (Richter, Yom-Tov, & Slonim, 2010) observe groups' behavior in a network instead of individuals' behavior. They called it Group-First Churn Prediction. By keeping only the strongest connections, they partition the network into a collection of small clusters which represent dense social groups. Machine learning techniques are used to establish a model that correlates group's explaining features with group's churn. The interactions within each group are also analyzed to identify the importance of each group member. The constructed model outputs group's churn score. And each customer is assigned an individual churn score based on the churn score of his social group as well as his personal characteristics.

Social influence is a driving factor of customer churn. In the scope of this thesis, the application of social network analysis in churn prediction is utilized mainly for extracting features that will be used as inputs in learning algorithms. The two quantities that will be calculated for each customer in the network are centrality degree and PageRank. Centrality degree is a term originally from graph theory. A degree is a number of connections that a node has so it indicates how tight a node is connected to the network. Centrality degree and PageRank are both connectivity measures of how well a node is connected to the network. The difference between them is that centrality measures the quantity while PageRank measures the quality of the connections.

PageRank is a method developed by the founder of Google which measures the relative importance of a web page based on the graph of the web. A webpage in the web in general can be viewed as a node in the network which consists of N nodes.

$$P(s) = \frac{(1-d)}{N} + d \sum_{t \in M(s)} \frac{P(t)}{C(t)} \quad \textbf{(2.1-1)}$$

$P(s)$ is a PageRank of s, $C(s)$ is a number of out-going links from s, $M(s)$ is a set of nodes that point to s, d is a damping factor varies between 0 and 1, usually d is set around 0,85. PageRanks of all web pages sum up to one (Brin & Page, 1998). The probability that a random surfer visits a page is its PageRank. A random surfer is the one who visits any web page by random and clicks on links randomly. The probability that he continues clicking on the same web page is d, and $(1-d)$ is the probability that he get bored and starts on a new random webpage.

PageRank is calculated using iterative algorithm. (Easley & Kleinberg, 2010) describe the PageRank computation briefly. Initially, all nodes have the same PageRank equal to $1/N$ where N is the number of nodes in a network. Next step, an update of PageRank values is performed as follows.

i. Each node divides its Pagerank into equal proportions and passes them to all nodes through its out-going links.
ii. Each node adds to its current Pagerank the new amount it receives from other nodes through its in-coming links.
iii. The Pagerank of each node is then scaled down by a factor of d.
iv. The residual $(1-d)$ is divided equally to all nodes so each node receives an addition of $(1-d)/N$ to its PageRank.

2.2 The Data Mining Process

The CRISP-DM model as it was claimed to be "The new blueprint for data mining" (Shearer, 2000) stands for Cross-Industry Standard Process for Data Mining. It is a guideline of how to conduct a data mining project, whose life cycle consists of six phases as shown in figure 2-1.

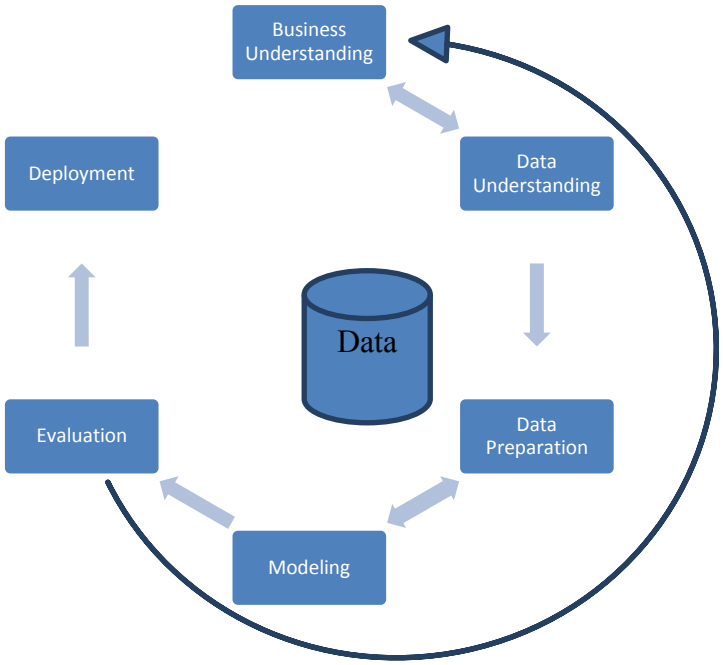

Figure 2-1. The phases of the CRISP-DM process model (CRISP-DM - Process Model)

The phases are not meant to be followed strictly sequentially in the right order. Through a project, one may have to repeatedly move back and forth between different phases. If the phases are sorted in the right order, the result of one phase is the input for the next phase. Start from the first phase, the goal is to reach the end phase and complete the project. But an outcome of each phase will decide if it is sufficient to move forward or revision is needed. As the analyst move along the project, he gains deeper and broader knowledge about the subject matter. Therefore, the analyst should go back and improve the previous implementation of foregone phases to obtain better outcome because it will affect the final outcome of the whole process. According to (Shearer, 2000), the phases are as follows:

- **Business understanding**. This is where the life cycle starts. The project objective is determined. The background, business perspective and resources are considered. The project objective is translated into a realistic data mining problem which is possibly solvable with respect to foreseen limitations. A project plan is outlined. This study is triggered by business motivation as described in chapter 1.1. Customer churn need to be prevented by churn prediction. The characteristics of the mobile telephony market call for certain desirable properties that a churn prediction solution must have as listed in chapter 1.2. The project objective is therefore to develop a churn prediction solution having these properties in mind. From the data mining perspective, churn prediction will be formulated as a classification task of churners and non-churners. Learning algorithms are applied on training data to build classifiers. Available resources are customer data in the data warehouse of the mobile operator and data mining software. The primary constraints are time and computational capacity of the analytical software.

- **Data understanding**. A sample of data must be examined to acquire insights into the data domain. In this phase the analyst gets to know which data are available and how to collect them. Exploration of data gives a basic understanding. Beyond that it can produce a discovery of interesting subsets from which the analyst can form initial hypotheses. In the business context, it is essential that an analyst gains his knowledge not only from the data itself but also from the domain experts. The mobile operator provides the author of this thesis full access to their data warehouse and initially a sample of available data that may benefit churn prediction such as demographics, revenue, number porting and call detail records. Time and effort is spent mainly on studying the architecture of the data warehouse and how to query the data from different sources.
- **Data preparation**. In short, this phase covers all tasks to make the final dataset from the raw data. It includes selection, construction, cleansing and transformation of data. For this purpose, a special database is built from which data needed for churn modeling can be generated. The structure of this database is described in chapter 4.1 and 0. The extracted data is then transferred to the data mining platform, where it can be transformed and sampled. Chapter 4.3 reports why and how data sampling is carried out. Several datasets are created from this data using different feature selection methods. Feature selection methods reduce the horizontal dimension of the data by selecting only the most relevant data related to churn. Results from feature selection are presented in chapter 4.4.
- **Modeling**. Knowledge and experience help the analyst to decide which modeling techniques and methodology to use. Some models require data in specific form which demand the analyst to have a visit to the data preparation phase again. Based on the literature reviewed in the previous chapter, four machine learning algorithms are chosen for modeling due to their proven efficiency and simplicity. After the model is built, it must be tested to determine its accuracy and generality. The performances of different models are compared according to predefined evaluation criteria. Model building, evaluation and comparison, all are reported in chapter 5.
- **Evaluation**. In this phase, conclusion can be drawn about the usefulness and reliability of the results and how well it solves the project objective. It is also about review throughout the whole process to assure that there isn't any important factor or task that has been missed out. Decision is taken here to move forward to the last phase, backward to some phases or even to the start point. Chapter 6 of this thesis sums up the work and draws conclusion about the main findings of this study. Works that are left undone are pointed out and ideas about future works are suggested.
- **Deployment**. Having in mind that it is the final user who will use the results of the data mining project. Therefore, the knowledge that has been obtained needs to be organized and presented in a way so that it can possibly be utilized by the user. The final product can be a presentation, a report or a computer program. This thesis is written not only to represent the results of the study but also to report in details what has been done so that this data mining process can be repeated and practiced by other experts within the industry.

Experiences in the data mining industry show that in general Data Preparation is the most resource consuming phase in a data mining project. It has been estimated that about 50 to 70 percent of the time and effort is put into this phase. The remaining 20 to 30 percent is spent in the Data Understanding phase, 10 to 20 percent for each of the Modeling, Evaluation and Business Understanding phases and about 5 to 10 percent for the Deployment phase (Shearer, 2000).

2.3 Summary

This chapter reviews how data mining and machine learning is applied to churn prediction in past and current research. Churn prediction belongs to analytical CRM and its goal is to estimate the probability that a currently active customer will churn in the near future. Studies focus on the search for methods and features that are the most effective in predicting churn. The most popular methods that have been used for churn prediction are such as: decision tree, regression, Naïve Bayes, artificial neural network and support vectors machine. Many studies attempt to show that complex black-box models such as artificial neural network and support vectors machine outperform white-box models like decision tree, regression, Naïve Bayes which are known to be simpler regarding structure and implementation. However, when those studies are compared, no conclusion can be drawn about which particular method is the best and which one is the worst of all. The achieved overall accuracy ranges from 50% to 90% . The features are extracted from a broad range of sources, from customer demographics and value, calling pattern, invoicing data, usage data, contractual data, data relating to subscriptions and cancellation of services, customer complaints and repairs interactions. Some studies also address the problem of skewed class distribution due to the fact that churn cases are often many times fewer than non-churn cases.

Along with the traditional machine learning approach, recent studies pursuing new research direction where social network analysis is employed for churn prediction. Some research in this promising area was introduced. This chapter also explains how social network analysis is utilized in this study to extract features based on centrality degree and PageRank. Centrality degree and PageRank are both connectivity measures of how well a node is connected to the network. The difference between them is that centrality measures the quantity while PageRank measures the quality of the connections.

The second part of this chapter reviews how the work of this study fits into the standard process of a data mining project defined by the CRISP-DM process model. The purpose of this review is to give an overview and better understanding of what has been done in this study and what lies ahead in this thesis.

3 The Classification Task

Supervised learning is a branch in machine learning of which the goal is to establish models that describe the relationship between training examples whose target concept values are known and the target concept. The constructed models are employed to identify the target concept of an unseen example. Machine learning algorithms are applied on data set to build learners. When the task is classification, these learners are called classifiers. The input to the classifier is a set of instances $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$. Each instance $\mathbf{x} \in \mathbb{R}^d$ is presented in the form of d-dimensional feature vector $\mathbf{x} = [x_1, x_2, \dots, x_d]$ and the class $y \in \{-1, +1\}$ in the case of binary classification. All discussions in this thesis about classification are referred to binary classification and the above notation is used throughout. In the context of churn prediction, the instances are customers and the class labels are churn and non-churn. Two ingredients for the recipe of a classifier are the data and the learning algorithms. A training set is required to build the model and estimate its parameters given an algorithm. As it is pointed out in (Alpaydin, 2010), the training error is not appropriate metric for deciding which classifier is better because the more complex model will in most of the cases give lower training error than the simpler one. Therefore a validation set is needed to tune the hyper-parameters and select the best model. Finally, the performance of a classifier is evaluated on a testing set.

The content of this chapter provides theoretical background for the classification task of this study as illustrated in figure 3-1. First of all, the sampled training data is fed into feature selection phase where irrelevant and redundant features are filtered out. Feature selection methods are introduced in chapter 3.1. The result from feature selection is dimensionally reduced data containing only the most relevant features. Afterwards, machine learning algorithm is applied on this reduced data to build classifier. Four learning algorithms C4.5 decision tree, alternating decision tree, Naïve Bayes and logistic regression are described in chapters 3.2, 3.3, 3.4 and 3.5 respectively. The performance of a classifier is evaluated using evaluation criteria represented in chapter 3.6.

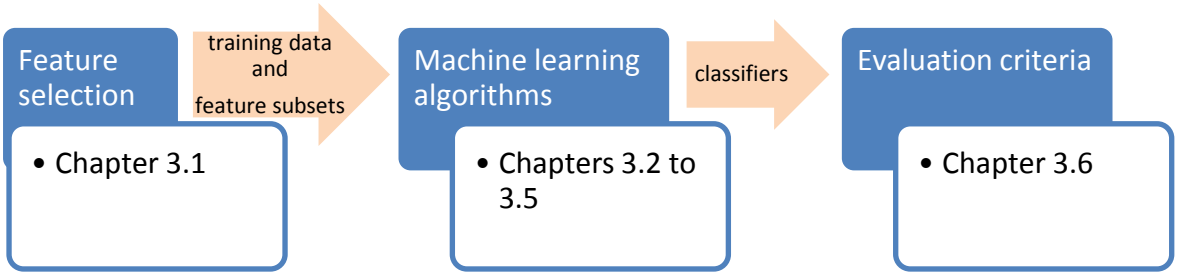

Figure 3-1. The content organization of chapter 3 - The Classification Task

3.1 Feature Selection

Feature selection refers to the process of selecting a subset of relevant features from a pool of features. First of all, this reduces the number of features as input to the model and therefore reduces the data acquisition cost and computational cost. Secondly, it yields not only more accurate but also more compact and interpretable results. As described in (Kira & Rendell, 1992), "Feature selection, as a preprocessing step to machine learning, has been very effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility". Feature selection includes individual or subset selection. Individual feature selection ranks features separately according to a particular metric where the subset selection takes into account the interaction and correlation among features.

A relevant feature is neither irrelevant nor redundant to the target concept. An irrelevant feature does not affect the target concept in any way, and a redundant feature does not add anything new to the target concept (Dash & Liu, 1997), (John, Kohavi, & Pfleger, 1994). Relevant features are the ones that contribute to the target concept. Irrelevant features are unwanted while redundant features are unnecessary. Correlations of relevant features lead to redundant features (Hall M. A., 1999), (Yan, Wolniewicz, & Dodier, 2004) because to describe the target concept only one of them is needed since they are correlated.

The process of any feature selection method includes: first to generate a candidate feature subset and second is to evaluate the generated candidate and calculate relevancy score. Based on relevancy score, the predefined stopping criteria determines whether it is the optimal feature subset. If yes, the process ends, else the generation process will start again to generate the next candidate feature subset. Overall, feature selection is basically a search method of which the four main components are a starting point, a search strategy, an evaluation function and a stopping criterion. (Hall M. A., 1999).

1. **Starting point**. A search needs to start somewhere, in this case a specific point in the feature subset space. One choice is to start with an empty set, proceeding forward and collecting features step by step. This method is known as sequential forward selection. Another choice is to include all features at the beginning, proceeding backward and removing them gradually. This method is known as sequential backwards selection. One can also start the search somewhere in the middle with a random feature subset and combine the two pre-mentioned selection methods into a so called bi-directional search which use both addition and deletion.

2. **Search strategy**. If there is no limitation on time and resources, the best thing to do is to carry out an exhaustive search over the feature subset space where all combination of the features is tested. It guarantees that the global optimal solution will be found but it is too computationally expensive and not practical for commercial uses. A simple math calculation shows that d initial features form 2^d possible subsets. Hence, with a large number of features, heuristic search strategies are preferable. They use a heuristic evaluation function to guide a search to explore the space of feature subsets. They work faster since their search space is smaller. But it is not guaranteed that the optimal feature subset will be found. However, good results can be achieved in a reasonable time. Yet another alternative is random search of which the result's quality depends on the number of trials. A popular and straightforward search strategy is greedy hill climbing. The algorithm considers all possible options of a local change from the current state by adding or deleting a single feature to or from the current

feature subset. The addition or deletion of a single feature that increases the worth of the current subset the most will be carried out. A pseudo code of greedy hill climbing is given in table 3-1.

Table 3-1. A greedy hill climbing search algorithm (Kohavi & John, 1997)

1. Let $S \leftarrow$ initial state; $S_{best} \leftarrow$ initial state
2. Expand S: apply all operators to S, giving S's children
3. Apply the evaluation function f to each child S' of S
4. Let $S =$ the child S' with highest evaluation $f(S')$
5. If $f(S) > f(S_{best})$ then $S_{best} \leftarrow S$; go to 2
6. Return S_{best}

As in greedy hill climbing, BestFirst search keeps moving forward by exploring all new possibilities from the current best subset. In addition, the algorithm maintains two lists. OPEN stores all feature subsets evaluated so far but haven't been chosen. At each iteration, the best feature subset among all candidates is chosen and moved from OPEN to CLOSED. At any time, the search can pick up a not yet chosen candidate in the OPEN list and continue from there if it is a better option than the current expanded search path. A pseudo code of BestFirst is given in table 3-2. (Kohavi & John, 1997).

Table 3-2. The BestFirst search algorithm (Kohavi & John, 1997)

1. Put the initial state on the OPEN list; CLOSED list $\leftarrow \emptyset$; $S_{best} \leftarrow$ initial state
2. Let $S = \arg\ \max_{R \in \text{OPEN}} f(R)$ (get the state from OPEN with maximal $f(R)$)
3. Remove S from OPEN and add S to CLOSED
4. If $f(S) > f(S_{best})$, then $S_{best} \leftarrow S$
5. Expand S: apply all operators to S, giving S's children
6. For each child not in the CLOSED or OPEN list, evaluate and add to the OPEN list
7. If S_{best} changed in the last k expansion (k is a predefined parameter), go to 2
8. Return S_{best}

3. **Evaluation function**. A search has to know what it is searching for. An evaluation function measures the worth of a feature subset. (Dash & Liu, 1997) divide the evaluation functions into five groups: distance, information, dependence, consistency and classifier error rate measure.

   i. Distance measure is based on an assumption that a good feature subset should support instances of the same class to stay close to each other and instances of different classes to stay away from each other. Relief (given in table 3-3) is a feature weight-based algorithm.

Table 3-3. Original Relief algorithm (Liu, Motoda, & Yu, 2004)

Given n- number of sampled instances, and d- number of features,

1. Set all weights $w_i = 0$
2. For $j = 1$ to n do begin
3. randomly select an instance $\mathbf{x}$
4. find nearest hit H and nearest miss M
5. For $i = 1$ to d do begin
6. $w_i = w_i - \text{diff}(x_i, \mathbf{x}, H)/\text{n} + \text{diff}(x_i, \mathbf{x}, M)/\text{n}$
7. end;
8. end;

At the beginning each feature is assigned a weight of zero. A feature weight measures the relevance of a feature to the target concept. Relief randomly samples instances in the dataset. For each selected instance, the algorithm finds its nearest hit H and nearest miss M based on Euclidean distance. Nearest hit is a nearest instance of the same class and nearest miss is a nearest instance of the opposite class. The weight of each feature is scaled up if that feature distinguishes between an instance and its nearest miss. The pre-mentioned weight is scaled down if that feature distinguishes between an instance and its nearest hit. The division by n normalizes all weights to the interval [-1,1]. At the end, the features are sorted by weights in descending order. The chosen feature subset consists of either the top features in the list or those having weights above a specific threshold. Relief doesn't take into account the interdependence among features. According to (Kira & Rendell, 1992), Relief does not help with redundant features. It selects all relevant features which results in unnecessary large feature subset.

The function diff calculates difference between the values of feature x_i in any two instances $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$. The difference between nominal feature values is 0 or 1 depends on whether the values are different or not. For continuous feature, diff is the normalized difference between feature's values.

$$\text{diff}\left(x_i, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \begin{cases} 0 \text{ if } x_i \text{ is nominal, } x_i^{(1)} \neq x_i^{(2)} \\ 1 \text{ if } x_i \text{ is nominal, } x_i^{(1)} = x_i^{(2)} \\ \dfrac{\left|x_i^{(1)} - x_i^{(2)}\right|}{\text{nu}_i} \text{ if } x_i \text{ is continuous} \end{cases} \quad \textbf{(3.1-1)}$$

where nu_i is a normalization unit which normalizes the value of diff function to the interval [0,1].

ii. Information measure estimates the information gain from a feature. Entropy is used as a measure of information content. The feature subset that yields the maximum entropy reduction which is equivalent to maximum information gain is selected. The reduction in entropy of the class given a feature is defined as the difference between the prior entropy and the posterior entropy after observing a feature.

$$\text{InfoGain(Class,Feature)} = \text{Entropy(Class)} - \text{Entropy(Class| Feature)} \quad \textbf{(3.1-2)}$$

The entropy of a discrete random variable Y is defined as follows.

$$\text{Entropy}(Y) = -\sum_{y \in Y} p(y) \log p(y) \quad \textbf{(3.1-3)}$$

The entropy of random variable Y after observing random variable X.

$$\text{Entropy}(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log p(y|x) \quad \textbf{(3.1-4)}$$

The information gain of Y given X.

$$\begin{aligned}\text{InfoGain}(Y,X) &= \text{Entropy}(Y) - \text{Entropy}(Y|X) \\ &= \text{Entropy}(X) - \text{Entropy}(X|Y) \\ &= \text{Entropy}(Y) + \text{Entropy}(X) - \text{Entropy}(X,Y)\end{aligned} \quad \textbf{(3.1-5)}$$

iii. Dependence measure evaluates the correlation between a feature and the class. The one which is more related to the class will be chosen. Correlation or dependence between features, also known as degree of redundancy. If a feature is heavily dependent to other feature, it is redundant and therefore unnecessary. (Hall M. A., 1999) puts forward a definition of a good feature subset "A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other". This definition as a guideline leads to an approach named Correlation-based feature selection CFS. The CFS evaluation function which measures the relevancy and redundancy of a feature subset is expressed as:

$$M_S = \frac{d\overline{r_{cf}}}{\sqrt{d + d(d-1)\overline{r_{ff}}}} \quad \textbf{(3.1-6)}$$

where d is the number of features in the subset S, $\overline{r_{cf}}$ is the average class-feature correlation between features in S and the class, $\overline{r_{ff}}$ is the average feature-feature correlation among the features in S. CFS searches in the feature subset space for a feature subset that maximize M_S therefore, it searches for a group of least correlated features that are most relevant to the class. The correlation in equation (3.1-6) is computed using symmetrical uncertainty introduced in (Press, Flannery, Teukolski, & Vetterling, 1988).

$$\text{symmetrical uncertainty} = 2\left[\frac{\text{InfoGain}}{\text{Entropy}(Y) + \text{Entropy}(X)}\right] \quad \textbf{(3.1-7)}$$

The quantities in equation (3.1-7): the gain in the numerator and the entropy in the denominator have been mentioned in the previous text about information measure and formulated in equations (3.1-3), (3.1-4) and (3.1-5). Substitute (3.1-5) into (3.1-7), the symmetrical uncertainty can be written as:

$$\begin{aligned}\text{symmetrical uncertainty} &= 2\left[\frac{\text{Entropy}(Y) + \text{Entropy}(X) - \text{Entropy}(X,Y)}{\text{Entropy}(Y) + \text{Entropy}(X)}\right] \\ &= 2\left[1 - \frac{\text{Entropy}(X,Y)}{\text{Entropy}(Y) + \text{Entropy}(X)}\right]\end{aligned} \quad \textbf{(3.1-8)}$$

If X and Y are uncorrelated, $\text{Entropy}(X,Y) = \text{Entropy}(Y) + \text{Entropy}(X)$ and the symmetrical uncertainty is equal to zero. When $\text{Entropy}(X,Y) = \text{Entropy}(Y) = \text{Entropy}(X)$, the symmetrical uncertainty reaches its maximum which is one. Since the above measure can only be applied to discrete random variables. All continuous features are converted to nominal using discretization in a preprocessing step.

iv. Consistency measure. In this context, the definition of inconsistency is: two instances are inconsistent if they have matching feature values but group under different classes. It is almost impossible to find the feature subset that doesn't contain any inconsistent instances pair. The goal is to look for a smallest subset

that satisfies a predefined inconsistency rate. In other words, the goal is to approach the min-feature-bias which is a smallest feature subset that defines a dataset so that the proportion of inconsistent instances pairs it contains is below a predefined rate.

Table 3-4. LVF algorithm (Liu & Setiono, 1996)

Input: MAX-TRIES-number of iterations; D-dataset; d-number of features; γ-allowable inconsistency rate;
Output: a feature subset S that satisfies the inconsistency criterion
$C_{best} = d$;
For $i = 1$ to MAX-TRIES
  $S = \text{randomSet(seed)}$;
  $C = \text{numOfFeatures}(S)$;
  If $C < C_{best}$
    If $\big(\text{Inconsistency\_Check}(S, D)\big) < \gamma$
      $S_{best} = S$; $C_{best} = C$;
      $\text{Print\_Current\_Best}(S)$;
    end if;
  else if $C = C_{best}$
    If $\big(\text{Inconsistency\_Check}(S, D)\big) < \gamma$
      $\text{Print\_Current\_Best}(S)$;
    end if;
  end if;
end for;

Liu and Setiono (Liu & Setiono, 1996) proposes an approach to feature selection named filter version of Las Vegas algorithm (LVF). The LVF algorithm (see table 3-4) carries out a certain number of iterations, MAX_TRIES. At each iteration, a random subset S is generated from the feature subset space. If S fulfils two following conditions, the current best subset S_{best} will be replaced by S. The conditions are: S contains fewer features than S_{best} and the inconsistency rate of the reduced data bounded by S is below γ, the allowable inconsistency rate. A group of matching instances is considered inconsistent if their values match for all features except the class. The inconsistency rate of a dataset bounded by a given feature subset is calculated in two steps: (1) The inconsistency count of each group of matching instances is the total number of instances in the group minus the number of instances in the group with the major class; (2) The overall inconsistency rate is the sum of all the inconsistency counts divided by the total number of instances in the dataset.

v. Classifier error rate measure. The classifier itself is used as an evaluation function. The worth of a feature subset is measured by classification error rate on a training set using cross validation. Brief introduction of cross validation is given in chapter 4.3.

The first four evaluation methods belong to an approach called filter. Filters operate independently of a learning algorithm. They ignore the effect of selected subset on the performance of a classifier. The evaluation is performed by a heuristic that is unrelated to the learning algorithm. The last evaluation method belongs to another approach

called wrapper. In wrapper method, the evaluation is performed by the same learning algorithm that will be used for the classification task. The choice between filter and wrapper is about the tradeoff between generality and accuracy. Wrapper often provide higher predictive accuracy since it is optimized for the target learning algorithm. However, filter execution time is considerably shorter than of a wrapper because the calculation of the heuristics functions is in general simpler than the error rate estimation using cross validation. In addition, wrappers need to be run again for each particular algorithm while filters are general and can be re-applied for different learning algorithm.

The difference between filter and wrapper is demonstrated in figure 3-2. A search generates a candidate feature subset from the feature space of the training data. The worth of this subset is evaluated using a specific heuristic function in filter approach and a machine learning algorithm in wrapper approach. The relevancy measure is the heuristic merit if a heuristic function is used and the estimated accuracy if a machine learning algorithm is used. At this point, if the stopping criterion is met, the current best feature subset is returned. Otherwise another search is carried out.

4. **Stopping criterion**. A stopping criterion is decided beforehand under which condition the search will terminate. The choice of stopping criterion is often based on the search strategy and the evaluation function. For example to stop when a predefined number of features or iterations have been reached. Other option is to stop when no further improvement is achieved by making changes to the current best subset.

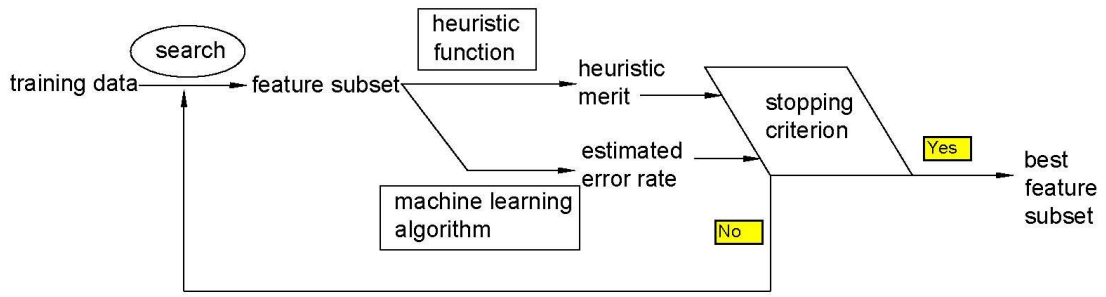

Figure 3-2. The feature selection process – The difference between filter and wrapper approach

3.2 C4.5 Decision Tree

First introduced by (Hunt, Marin, & Stone, 1966), decision tree learning has become one of the most widely used and researched machine learning methods. As white boxes, decision trees generate interpretable and understandable models. Induction of decision tree involves building a tree top-down using divide and conquers strategy. The ultimate goal is recursively partition the training set, choosing one feature to split each time until all or most of instances in each partition belong to the same class.

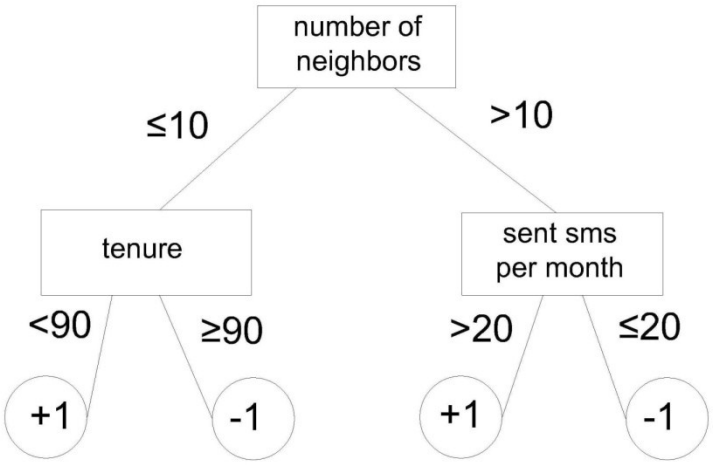

Figure 3-3. An example of a C4.5 decision tree built for churn prediction in the mobile market

A decision tree consists of four main elements: a root; decision nodes indicate features used for splits; branches correspond to possible outcomes of feature value and finally leaves that specify expected value of the class. Each leaf is assigned to the class that has the majority of instances inside it. To classify a new instance, start at the root and follow a path lead by the nodes and branches downward, end at a particular leaf and the instance is assigned a class specified by the leaf.

Figure 3-3 illustrates an example of a decision tree built for churn prediction in the mobile market. Assuming that +1 denotes churner and -1 non-churner, the tree divides customers into four groups. Those who have more than ten neighbors and send more than 20 text messengers per month, or those who have ten or less neighbors and have tenure less than 90 days are likely to churn. The characteristics of those who are less likely to churn can be interpreted from the tree with the same manner. The Hunt's algorithm for decision tree construction called CLS (Concept Learning System) created a foundation for decision tree learning.

Given a set D of training cases and the classes are $\{y_1, y_2, \dots, y_c\}$. The tree is constructed recursively by following three rules:

- If D contains one or more cases which all belong to the same class $y_j$then a leaf node is created representing class y_j
- If D contains no cases then a leaf node is created. The class associated with the leaf node is then assumed to be the most frequent class at the parent node.
- If D contains cases which belong to more than one class then choose a test that split D into subsets of cases so that each subset is or nearly is a single-class collection of cases. The test is based on a single feature x_i that has one or more possible outcomes $\{x_{i1}, x_{i2}, \dots, x_{im}\}$. D is split into subsets $\{D_1, D_2, \dots, D_m\}$ where each subset contains all cases in D that have the same outcome of x_i. A decision node that represents the test is created and branches grow out of this node denote the possible outcomes.

The core of a decision tree learning algorithm is the evaluation metric which measures the goodness of a split. Employing the fundamental rules of CLS, (Quinlan, 1986) developed ID3 algorithm in which information gain heuristic is used as the test to choose the best split. C4.5, an improved and extended version of ID3 is presented in (Quinlan, 1993). The purpose of each split is to end up with child nodes that are purer than a parent node. Inspired by entropy measure in information theory (Quinlan, 1986) applied it to estimate the impurity of a group

of instances. Adopting the general definition of entropy in equation (3.1-3) in this context results in:

$$\text{Entropy}(D) = -\sum_{j=1}^{c} p_j \log_2 p_j \qquad \textbf{(3.2-1)}$$

Where c is the number of different classes and p_j is the proportion of D (a group of instances) belonging to class j. According to this definition, a completely pure node in which all instances belong to the same class has entropy equal to zero. In the case of only two classes, a positive and a negative one, the highest entropy occurs in a node where numbers of positive and negative instances are equal.

In order to evaluate the goodness of a split, ID3 compares the entropy of a parent node to the weighted sum of its child nodes' entropies after the split. A function called gain criterion is used to compute the entropy reduction cause by a split according to a particular feature.

$$\text{Gain}(D, F) = \text{Entropy}(D) - \sum_{v \in \text{Values}(F)} \frac{|D_v|}{|D|} \text{Entropy}(D_v) \qquad \textbf{(3.2)}$$

Where $\text{Values}(F)$ is the set of all possible values for feature F, D_v is a subset of D which contains instances whose value of F equals to v (Mitchell, Machine Learning, 1997). So at each step, we choose a split that causes the largest entropy reduction which is equivalent to the highest information gain.

However, a shortcoming of gain criterion is that it favors features with many possible values. An extreme example is a data set of customers with many features and one feature denotes customer's ID. Since customer's ID is unique for each customer, choosing this feature to split the data set results in a large number of subsets, each contains only one instance and each has the entropy of zero. The gain is maximized but the split is useless and the tree will have no prediction power. (Quinlan, 1993) adopts a function to penalize such feature called split information which measures the entropy of set D with respect to feature F.

$$\text{SplitInfo}(D, F) = -\sum_{v \in \text{Values}(F)} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|} \qquad \textbf{(3.2-2)}$$

Incorporating with gain criterion, C4.5 uses a so-called gain ratio criterion that balances between the two targets of achieving the purest child nodes and using the fewest branches.

$$\text{GainRatio}(D, F) = \frac{\text{Gain}(D, F)}{\text{SplitInfo}(D, F)} \qquad \textbf{(3.2-3)}$$

Back to an extreme example above, the entropy of a training set with respect to this ID feature is larger than of all other feature that has fewer values. If there are n instances with n customer ID then the SplitInfo is equal to $\log_2 n$. It follows that this ID feature will be eliminated in the selection of split according to the GainRatio function.

However, another problem arises in the case of a feature that varies narrowly where almost all instances have the same feature's value. The SplitInfo is then approximately equal to zero and the GainRatio approaches infinity or is undefined. The solution is first to select only features for split having the Gain values that are above the average Gain of all tested features.

GainRatio is then calculated for features that survived the first test and find the single feature that has the maximum GainRatio among them.

The discussion so far is restricted to nominal features. Since the task is classification, the class feature has to be nominal. Otherwise, regression tree is used which is beyond the scope of this thesis. C4.5 can handle continuous features other than the class feature. It looks for a threshold value v of a continuous feature F which gives the best split. Two child nodes are formed after the split. One contains instances whose value of F is above v while the other contains instances whose value of F is equal or below v.

Yet another concern that needs attention is the existence of instances with missing feature values. C4.5 provides separated solutions for three phases in the tree construction process: calculation of GainRatio, instances partition and instances classification. When only a fraction of instances in set D have known value for feature F and value of F is missing in the rest of instances, a modification is made in information gain calculation.

$$\text{Gain}(D,F) = r * \left\{ Entropy(D) - \sum_{v \in Values(F)} \frac{|D_v|}{|D|} Entropy(D_v) \right\} \quad \textbf{(3.2-4)}$$

Where r is the proportion of instances in D with known value for feature F. Unknown value is treated as an additional possibility of outcome in the estimation of the SplitInfo. After a feature for split has been chosen, the task is where to send instances with unknown value for this feature downward the child nodes. A weight is assigned to each instance represents the probability that it belongs to a child node. An instance with a known value v for feature F is sent to a corresponding child node. The probability that it belongs to this node is one so it receives the weight $w = 1$. Only a fraction of an instance with unknown value is sent to each child node $w = w * p$. The fraction p is estimated as the sum of the weights of instances in a child node divided by the sum of the weights of instances with known value for feature F. A similar method is used to classify a new instance whose feature value is missing. At a decision node, since an outcome is unknown, the algorithm explores all possible paths which lead to leaves with different classes. An instance is classified as the class with the highest probability.

In C4.5 algorithm a tree is grown fully and it continues to split until a node is completely pure. However, a node with too few instances has almost no prediction power. So there is a trade-off between accuracy and generalization, a balance between the pureness and the size of a node. Aiming to simplify a tree in the last phase of tree construction, tree pruning reduces the complexity, avoids over-fitting and increases the accuracy. There are two ways of implement tree pruning: pre-pruning and post-pruning. Pre-pruning is carried out while the tree is being induced. It prevents the formation of node which contains too few instances therefore has insignificant generalization ability. The choice of the minimum allowable number of instances per leaf depends on each data set, often found by experiments.

Post pruning works on a fully induced tree. After the full-size tree has been built, it will be pruned backward. Each node in the tree is considered for pruning whether the node including its sub-tree can be replaced with one leaf. All instances that belong to the sub-tree are transferred to the substituted leaf. A separated data set from the training set used for tree building called pruning set is prepared. If the pruned tree performs as good as or better on the pruning set, then the node and its sub-tree are removed. When a separated data set is used to estimate the classification error rate, it is called reduced-error pruning. Another approach adopted in C4.5 named pessimistic pruning. Given a leaf contains N instances, E of them are

misclassified so the observed error rate for this leaf is $\mathrm{E/N}$. Assuming that the error rate follows binomial distribution, its expected value lies inside a confidence interval bounded by two limits. For a confidence factor α (also known as significance level), the upper limit $\mathrm{U}_\alpha(\mathrm{E},\mathrm{N})$ is used as an estimate of the error rate. Therefore the expected number of misclassified instances in a leaf is $\mathrm{e} = \mathrm{N} * \mathrm{U}_\alpha(\mathrm{E},\mathrm{N})$. And the estimate of the error rate of a tree is calculated as follows.

$$e_t = \frac{\sum_{i=1}^{l} N_i * U_\alpha(E_i, N_i)}{\sum_{i=1}^{l} N_i} \quad \textbf{(3.2-5)}$$

Where l is number of leaves in the tree. If an estimated error rate of an un-pruned tree is higher than of a pruned tree then pruning is carried out.

3.3 Alternating Decision Tree

Boosting is a well-known machine learning technique inspired by the idea of making one better classifier by combining multiple classifiers. In boosting algorithms, classifiers are built, assessed and added iteratively one by one to the ensemble. After each addition, the training data is re-weighted. The instances that are correctly classified lose weight and the instances that are misclassified gain weight. The weight of instances contributes to the error calculation of the algorithms. Therefore, in the next iteration, a generated classifier will concentrate on those instances with higher weight in order to minimize the error. In this manner, classifiers in the ensemble complement one another. A posterior classifier try to improve the classification of instances that preceding classifiers have failed. One of the disadvantages of boosting is the final classifiers ensemble is large, complex and difficult to interpret (Freund & Mason, 1999).

Alternating decision (AD) tree learning algorithm exploits the concept of boosting and decision tree to create a new type of classifier. An AD tree is an ensemble of units called base rules. A base rule r consists of four main elements: a precondition c_1, a condition c_2 and two prediction values a and b. Similar to conventional decision tree, each condition is stored in one decision node. In addition, in AD tree, each decision node leads to two prediction nodes which store prediction values as real numbers. A base rule r maps each instance to a prediction value that is defined as follows.

$$r(\mathbf{x}) = \begin{cases} a, & \text{if } c_1 \wedge c_2 \\ b, & \text{if } c_1 \wedge \neg c_2 \\ 0, & \text{if } \neg c_1 \end{cases} \quad \textbf{(3.3-1)}$$

To classify a new instance, one follows a multi-path associated with that instance and collects the sum of prediction values stored in prediction nodes. The sign of the final sum decides to which class the instance belongs. Figure 3-4 demonstrates an example of an AD tree built for churn prediction in the mobile market. Take an example of a customer with the following characteristics: his tenure is less than 90 days, he has more than ten neighbors and sends less than 20 text messengers per month. His total prediction value according to figure 3-4 will be $0{,}8 + 0{,}4 - 0{,}6 = 0{,}6$. A positive value denotes that this customer is classified as a churner.

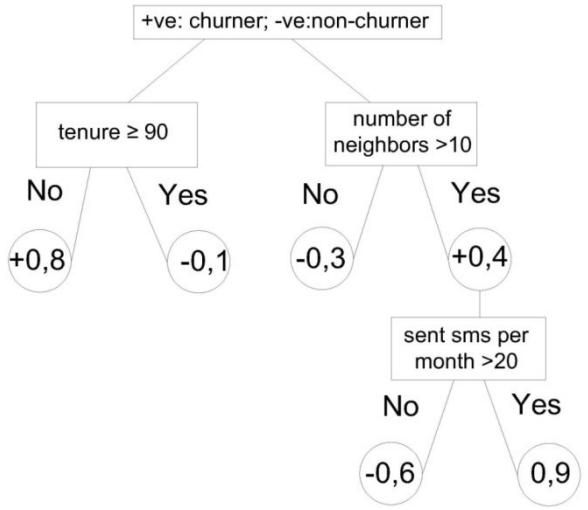

Figure 3-4. An example of an AD tree built for churn prediction in the mobile market

An AD tree is grown iteratively by boosting iterations. At each iteration, the algorithm maintains two sets, a set of preconditions $\mathcal{P}_t$ and a set of rules $\mathcal{R}_t$ where t is the iteration index. After each iteration, a new chosen base rule is added to the tree. Number of boosting iterations nb is a hyper-parameter, often tuned by experiments on each particular data set. (Pfahringer, Holmes, & Kirkby, 2001) describe the AD tree learning algorithm in four main steps.

Initialize. Set the weight of all instances to 1. Set the first rule $\mathcal{R}_1$ to consist of a base rule whose both precondition and condition are true. The prediction value is $a = \frac{1}{2}\ln\frac{W_+(c)}{W_{-(c)}}$ where $W_+(c)$ and $W_-(c)$ are the total weight of the positive and negative instances in the training data that satisfy condition c. Initially, c is set to be true.

Pre-adjustment. All instances are re-weighted $w_{i,1} = e^{-ay^{(i)}}$ where $i = 1,2,\dots,N$ is the instance index.

Do for t= $1,2,\dots,nb$

1. Generate the set $\mathcal{C}$ of conditions using weights associated with each training instance $w_{i,t}$

2. For each precondition $c_1 \in \mathcal{P}_t$ and each condition $c_2 \in \mathcal{C}$ calculate
$$Z_t(c_1,c_2) = 2\left(\sqrt{W_+(c_1 \wedge c_2)W_-(c_1 \wedge c_2)} + \sqrt{W_+(c_1 \wedge \neg c_2)W_-(c_1 \wedge \neg c_2)}\right) + W_+(\neg c_1)$$

3. Select c_1, c_2 which minimize $Z_t(c_1,c_2)$ and set $\mathcal{R}_{t+1}$ to be $\mathcal{R}_t$ with the addition of the rule r_t whose precondition is c_1, condition is c_2 and two prediction values are:
$$a = \frac{1}{2}\ln\frac{W_+(c_1 \wedge c_2)+1}{W_-(c_1 \wedge c_2)+1} \quad \textbf{(3.3-2)}$$
and
$$b = \frac{1}{2}\ln\frac{W_+(c_1 \wedge \neg c_2)+1}{W_-(c_1 \wedge \neg c_2)+1} \quad \textbf{(3.3-3)}$$

4. Set $\mathcal{P}_{t+1}$ to be $\mathcal{P}_t$ with the addition of $c_1 \wedge c_2$ and $c_1 \wedge \neg c_2$

5. Update the weights of each training instances:

$$w_{i,t+1} = w_{i,t} e^{-r_t(\mathbf{x}^{(i)})y^{(i)}}$$

Output. The classification rule is the sign of the sum of all base rules in $\mathcal{R}_{T+1}$

$$\text{class}(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} r_t(\mathbf{x})\right)$$

3.4 Naïve Bayes

Naïve Bayes learning generates a probabilistic model of the observed data. Despite its simplicity, Naïve Bayes has been verified to be competitive with more complex algorithm such as neural network or decision tree in some domains (Mitchell, Machine Learning, 1997), (George & Langley, 1995). Given a training set of instances, each is represented as a vector of features $[x_1, x_2, \dots, x_d]$, the task is learning from the data to be able to predict the most probable class $y_j \in \mathbb{C}$ of a new instance whose class is unknown. Naïve Bayes employs the Bayes's theorem to estimate the probabilities of the classes.

$$P(y_j|x_1, x_2, \dots, x_d) = \frac{P(y_j)P(x_1, x_2, \dots, x_d|y_j)}{P(x_1, x_2, \dots, x_d)} \quad \textbf{(3.4-1)}$$

Where $P(y_j)$ is the prior probability of class y_j which is estimated as its occurrence frequency in the training data. $P(y_j|x_1, x_2, \dots, x_d)$ is the posterior probability of class y_j after observing the data. $P(x_1, x_2, \dots, x_d|y_j)$ denotes the conditional probability of observing an instance with the feature vector $[x_1, x_2, \dots, x_d]$ among those having class y_j. And $P(x_1, x_2, \dots, x_d)$ is the probability of observing an instance with the feature vector $[x_1, x_2, \dots, x_d]$ regardless of the class. Since the sum of the posterior probabilities over all classes is one $\sum_{y_j \in \mathbb{C}} P(y_j|x_1, x_2, \dots, x_d) = 1$, the denominator on equation (3.4-1) 's right hand side is a normalizing factor and can be omitted.

$$P(y_j|x_1, x_2, \dots, x_d) = P(y_j)P(x_1, x_2, \dots, x_d|y_j) \quad \textbf{(3.4-2)}$$

An instance will be labeled as the particular class which has the highest posterior probability y_{MAP}.

$$y_{MAP} = \arg\max_{y_j \in \mathbb{C}} P(y_j)P(x_1, x_2, \dots, x_d|y_j) \quad \textbf{(3.4-3)}$$

In order to estimate the term $P(x_1, x_2, \dots, x_d|y_j)$ by counting frequencies, one needs to have a huge training set where every possible combinations $[x_1, x_2, \dots, x_d]$ appear many times to obtain reliable estimates (Mitchell, Machine Learning, 1997). Naïve Bayes solves this problem by its Naïve assumption that features that define instances are conditionally independent given the class. Therefore the probability of observing the combination $[x_1, x_2, \dots, x_d]$ is simply the product of the probabilities of observing each individual feature value $P(x_1, x_2, \dots, x_d|y_j) = \prod_{i=1}^{d} P(x_i|y_j)$. Substituting this approximation into equation (3.4-3) to derive the Naïve Bayes classification rule.

$$y_{MAP} = \arg\max_{y_j \in \mathbb{C}} P(y_j) \prod_{i=1}^{d} P(x_i|y_j) \quad \text{(3.4-4)}$$

As discussed above, for nominal feature, the probability is estimated as the frequency over the training data. For continuous feature, there are two solutions. The first one is to perform discretization on those continuous features, transferring them to nominal ones. The second solution is to assume that they to follow a normal distribution.

The term $P(x_i|y_j)$ is estimated by the fraction $\frac{\#D(x_i|y_j)}{\#D(y_j)}$, where $\#D(y_j)$ is the number of instances in the training set having class y_j, and $\#D(x_i|y_j)$ is the number of these instances having feature value x_i and class y_j. If the training data doesn't contain any instance with this particular combination of class and feature value, $\#D(x_i|y_j)$ is zero. The estimate probability according to equation (3.4-4) will be zero for every similar cases. To avoid this, a correction called the m-estimate is introduced (Gutkin, 2008), (Cetnik, 1990).

$$P(x_i|y_j) = \frac{\#D(x_i|y_j) + mP(x_i)}{\#D(y_j) + m} \quad \text{(3.4-5)}$$

If the prior probability $P(x_i)$ is unknown, uniform distribution is assumed, i.e. if a feature has k possible values, then $P(x_i) = 1/k$. The parameter m can be regarded as the additional m dummy instances appended to the training set.

3.5 Logistic Regression

Naïve Bayes is a generative classifier. Given the data $\mathbf{x} \in \mathbb{R}^d$ and the class $y \in \{-1, +1\}$, it learns a model of the conditional probability $P(\mathbf{x}|y)$ and the prior probability $P(y)$ to predict the most probable class $P(y|\mathbf{x})$. Meanwhile, logistic regression is a representative of discriminative classifier. It learns a direct map from input $\mathbf{x}$ to output y by model the posterior probability $P(y|\mathbf{x})$ directly (Ng & Jordan, 2002). The parametric model proposed by logistic regression is of the form.

$$P(y = -1|\mathbf{x}) = \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)} \quad \text{(3.5-1)}$$

And

$$P(y = 1|\mathbf{x}) = \frac{\exp\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)}{1 + \exp\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)} \quad \text{(3.5-2)}$$

The main task of logistic regression is adjusting the weights so that the model fits the data as well as possible.

$$\mathbf{w} = [w_0, w_1, w_2, \ldots, w_d] \leftarrow \arg\max_{\mathbf{w}} \prod_{k} P\left(y^{(k)}|\mathbf{x}^{(k)}, \mathbf{w}\right) \quad \text{(3.5-3)}$$

where $\mathbf{w}$ is the vector of parameters, $y^{(k)}$ is the observed value of y and $\mathbf{x}^{(k)}$ the observed value of $\mathbf{x}$ in the k^{th} training instance. The maximization of equation (3.5-3) is known as the maximum likelihood estimation (MLE).

$$\mathbf{w}=[w_0, w_1, w_2, \ldots, w_d] \leftarrow \arg\max_{\mathbf{w}} \sum_{k} \ln P\left(y^{(k)}|\mathbf{x}^{(k)},\mathbf{w}\right) = \arg\max_{\mathbf{w}} \mathrm{L}(\mathbf{w}) \qquad \textbf{(3.5-4)}$$

where $\mathrm{L}(\mathbf{w})$ is called the conditional log-likelihood of the class (Mitchell, 2005). Maximization of $\mathrm{L}(\mathbf{w})$ can be achieved for example by using gradient ascent.

Designed for continuous feature but logistic regression can still handle nominal feature and missing values. Nominal features are converted to binary features and missing values are replaced by the mean (continuous features) or the mode (binary features) of the training data.

3.6 Model Performance Evaluation

No Free Lunch theorem concludes that no algorithm is superior to another on average over all domains. Some works well in particular domains while the other perform better elsewhere. In this study, learning algorithms are applied on training data to build classifiers. The performances of the classifiers are compared or in other words, the performances of different learning algorithms on this particular data are compared. So what are the indicators of a "good" classifier? The desired properties are accuracy, generality and confidence level of prediction. Confusion matrix (see figure 3-5) presents four possible outcomes when a classifier is applied on a set of instances.

		Actual Class	
		p (+)	n (-)
Hypothesized class	p (+)	**True Positive**	**False Positive**
	n (-)	**False Negative**	**True Negative**
Column totals		P	N

Figure 3-5. The confusion matrix (Fawcett, 2006)

A correctly classified instance is counted as a true positive (TP) or a true negative (TN) if its actual class is positive or negative respectively. A positive instance which is misclassified as negative is counted as a false negative (FN). And a negative instance which is misclassified as positive is counted as a false positive (FP). The total number of positive instances in the data set is $\mathrm{P} = \mathrm{TP} + \mathrm{FN}$, and the total number of negative instances is $\mathrm{N} = \mathrm{TN} + \mathrm{FP}$. Based on a confusion matrix, the most common evaluation metrics are overall accuracy, true positive rate and false positive rate.

The overall accuracy (OA) is the proportion of the correctly classified instances.

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} \qquad \textbf{(3.6-1)}$$

The true positive rate (also known as hit rate) is the proportion of positive instances that a classifier captures.

$$\text{TP rate} = \frac{\text{TP}}{\text{P}} \quad \textbf{(3.6-2)}$$

And the false positive rate (also known as false alarm rate) is the proportion of negative instances that a classifier wrongly flagged as positive.

$$\text{FP rate} = \frac{\text{FP}}{\text{N}} \quad \textbf{(3.6-3)}$$

When TP rate is plotted as y against FP rate as x, one obtains a receiver operating characteristics (ROC) graph. Each classifier is represented by a point on ROC graph. A perfect classifier is represented by point (0, 1) on ROC graph which classifies all positive and negative instances correctly with 100% TP rate and 0% FP rate. The diagonal line $y = x$ demonstrates classification that is based completely on random guesses (Fawcett, 2006). In that case, one can achieve the desired TP rate but unfortunately also gain equally high FP rate. The major goal of churn prediction is to detect churn. Therefore, a suitable classifier is the one having high TP rate and low FP rate given that churn is the positive class. Such classifier is located at the upper left corner of ROC graph.

A classifier provides output in probabilistic form $P(y = +1|\mathbf{x})$, the probability that an instance belongs to the positive class. If this probability is above the predefined threshold $P(y = +1|\mathbf{x}) > \theta$, an instance is classified as positive, otherwise negative. A classifier using high value for θ is considered "conservative". It classifies positive instances only with strong evidence so it makes few FP mistakes but at the same time has low TP rate. A classifier using low value for θ is considered "liberal". It classifies positive instances with weak evidence so it achieve high TP rate but also makes many FP mistakes (Fawcett, 2006). When the performance of a classifier is plotted on ROC graph with value of θ varied from 0 to 1, a ROC curve will be formed. It demonstrates the trade-off between TP rate and FP rate.

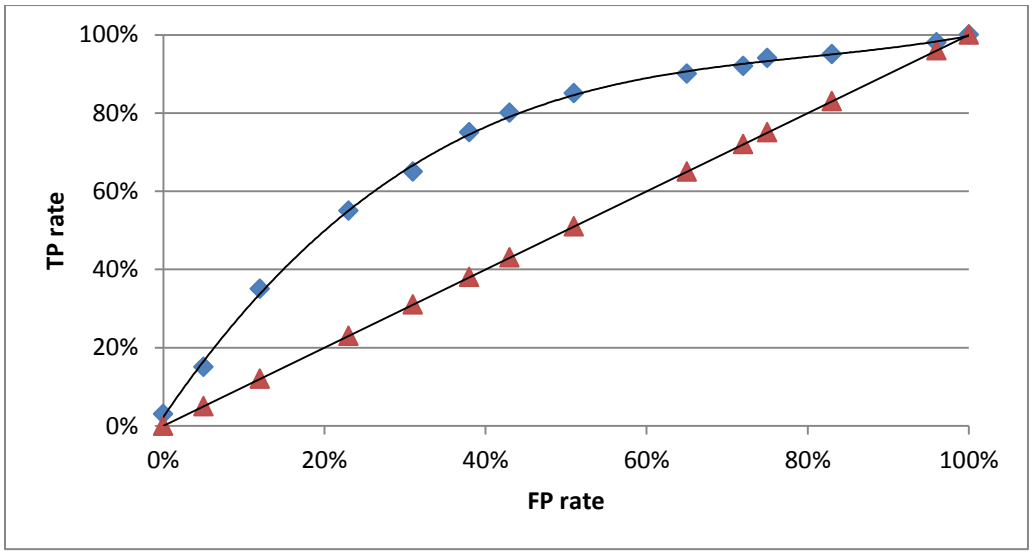

Figure 3-6. A sample ROC curve. The red points are random guess classifiers. The blue points obtained by varying the value of the threshold θ of a classifier from 0 to 1.

Figure 3-6 shows an example of ROC curve. The red points are random guess classifiers. The blue points represent the performance of a classifier using different values of θ. The higher θ is, the more "conservative" a classifier becomes. Conservative classifiers locate at the lower part of ROC graph. In contrast, the lower θ is, the more "liberal" a classifier becomes. Liberal classifiers locate at the upper part of ROC graph.

Given two ROC curves, the one that is further to the left of the random diagonal is preferred. For this reason, area under ROC curve (AUC), a quantity that measures the overall average performance of a classifier is introduced. The advantage of AUC is unlike many other evaluation metrics such as the overall accuracy, AUC is not affected by the class distribution, the P/N ratio. In the case of unbalanced class distribution such as in churn prediction data AUC gives a fair measure for model comparison.

Based on the classification result, mobile operator focuses on customers that are classified as positive or churners. However, the operator will probably be unable to react to all positive classified instances due to the lack of resources. Besides that, quality is more important than quantity. The question is not only how many percent of churners a model can covers but also with how much reliability. A classifier which covers 30% of churners with 90% reliability may be more preferable than the one which covers 50% of churners with 60% reliability? The choice is up to each company to evaluate the cost of ignoring customers in churn risk versus the cost of offering unnecessary special treatment for customers that will not churn. (Tong, Xie, Hong, Shi, Fang, & Perkins, 2004) suggests a formula to calculate the confidence level of a prediction.

$$\text{confidence level for class } y_j = \frac{P(y_j|\mathbf{x}) - 0{,}5}{0{,}5} \qquad \textbf{(3.6-4)}$$

A high confidence prediction $P(y_j|\mathbf{x}) > 0{,}7$ is equivalent to confidence level $> 0{,}4$. A random guess $P(y_j|\mathbf{x}) = 0{,}5$ results in confidence level of zero. The overall confidence level of a classifier on a testing set is:

$$\text{OCL} = \frac{1}{N_{\text{churners}}} \sum_{\mathbf{x}^{(k)} \in \Phi} \frac{P(y^{(k)} = +1|\mathbf{x}^{(k)}) - 0{,}5}{0{,}5} \qquad \textbf{(3.6-5)}$$

where N_{churners} is the total number of churners in the data set, Φ is a set of all true positive instances given that churn is the positive class. OCL is maximum and equals to one when all churners in the data set are captured with confidence levels of one. If only 30% of churners are captured with confidence levels of one, $\text{OCL} = 0{,}3$. Hence OCL not only reflects the quality of the prediction but also the accuracy of the classification.

3.7 Summary

This chapter provides theoretical background for the classification task of churners and non-churners in this study. In order to lessen the complexity and computational cost of the models, feature selection is employed prior to the modeling phase. Feature selection keeps only the most relevant features by filtering out irrelevant and redundant features. Overall, feature selection is basically a search method of which the four main components are a starting point, a search strategy, an evaluation function and a stopping criterion. Based on the evaluation functions, feature selection methods can be divided into five groups: distance, information,

dependence, consistency and classifier error rate measure. The first two methods evaluate each feature individually and give them scores. The next three methods search for the best subset of features with respect to the heuristic metrics. Thereof, the last method is a wrapper approach which uses a base classifier to assess subsets of features using cross validation.

After feature selection, machine learning algorithm is applied on the reduced data to build classifier. Four learning algorithms C4.5 decision tree, alternating decision tree, Naïve Bayes and logistic regression are described in this chapters. C4.5 is a decision tree builder using entropy for splitting nodes and adopting pruning to increase generality power. Alternating decision tree enhances the performance and interpretability of decision tree by adopting boosting techniques. Naïve Bayes is a probability approach based on Bayes's theorem. Logistic regression is a parametric model which outputs continuous probability value.

The last section of this chapter introduces different evaluation criteria that are used to evaluate the performance of a classifier. Having the main objectives and characteristics of churn prediction in mind, the author of this thesis selects the overall accuracy, the area under ROC curve and the overall confidence level of prediction for model comparison in this study.

4 Data Preparation

Being one of the two main ingredients of a classifier, the data takes a crucial role in the success of model construction. High performance data warehouse is run by the mobile operator making it possible to access, extract and manipulate the data needed for churn modeling. Information about customers that is useful for churn prediction are demographics, revenue, number porting and call detail records. From these sources, a special database is built for the purpose of data acquisition. The executed steps for building a data set from this database are listed in Appendix A. The database is defined by a relation model which is presented in chapter 4.1. The data can be extracted according to the timeline described in chapter 4.2. Churn data has highly skewed class distribution due to the fact that churn cases are often many times fewer than non-churn cases. Chapter 4.3 discusses about the sampling methods used to overcome this problem. General statistics of the data used in this study are also given in this chapter. After the work of data transforming and sampling has been done, different feature selection methods are carried out to search for the most significant features that influence churn. The results from the feature selection phase are given in chapter 4.4.

4.1 The Relation Model

Figure 4-1 illustrates the relation model of the customized churn prediction database. It consists of sources tables (the yellow ones), target tables (the blue ones) and a center view (the red one). The yellow tables are sources tables which can be generated from available data in the data warehouse. The blue tables are target tables. They are built by executing the corresponding procedures whose names start with the word *build* followed by the tables' names. At the center of this relation model is a view in the database named *Cust_signature*. Each line in *Cust_signature* is an instance of customer and the columns contain all features associate with this particular customer. *Cust_signature* can be created with the union of six target tables which form the star schema around it: *cust_demographic*, *calling_pattern*, *refill_history*, *cdr_billed*, *billing_data*, *sna_network*. Similar to *Cust_signature*, each target table contains records of customers with features stored in the columns. The features of these tables belong to six following categories.

- Demographics. This group of features contains primary characteristic of customers such as their age, gender, family size, marital status, residence. In addition are features which describe general characteristic of their subscriptions such as if a customer is the payer of his usages or not, his rate plan and tenure. Demographics features explain the behavior of customers. Younger customers may send more text messengers and older customers with a big family may call more often. Long-tenure customers are often considered to be more loyal to the service provider.

- Billing data. Features in this group are generated for postpaid customers only. High bills may lead to churn decision of a customer. On the other hand, discounts may please customers. Besides mobile service, many customers exploit other services that the operator provides as well. In that case, a customer often prefers to buy all services from the same operator. So if the mobile usage is just a minor part of his total billing,

he may not be as sensitive to the price. In other words, he focuses more on the convenience of having all services at the same place than the price.

- Refill history. This group describes the general behaviors of prepaid customers. Such as how frequently and how much one has spent on refills. Customers choose prepaid subscription for economical reason so they are especially sensitive to the price. Spending too much on refills in a short period can encourage them to look for a cheaper alternative.

- Calling pattern. Features of this group attempt to capture calling behaviors and changes in calling pattern of customers such as the volume and frequency of inside, outside network and international calls or amount of sending and receiving text messengers.

- CDR billed. As the consequence of calling behavior, this group computes the costs due to different usage types.

- Network. Based on the hypothesis that people that are connected to customers influence their churn decision, this group includes features such as number of neighbors and churn neighbors, call volume to and from the neighbors, centrality degree and PageRank. A brief introduction of centrality degree and PageRank can be found at the end of chapter 2.1.

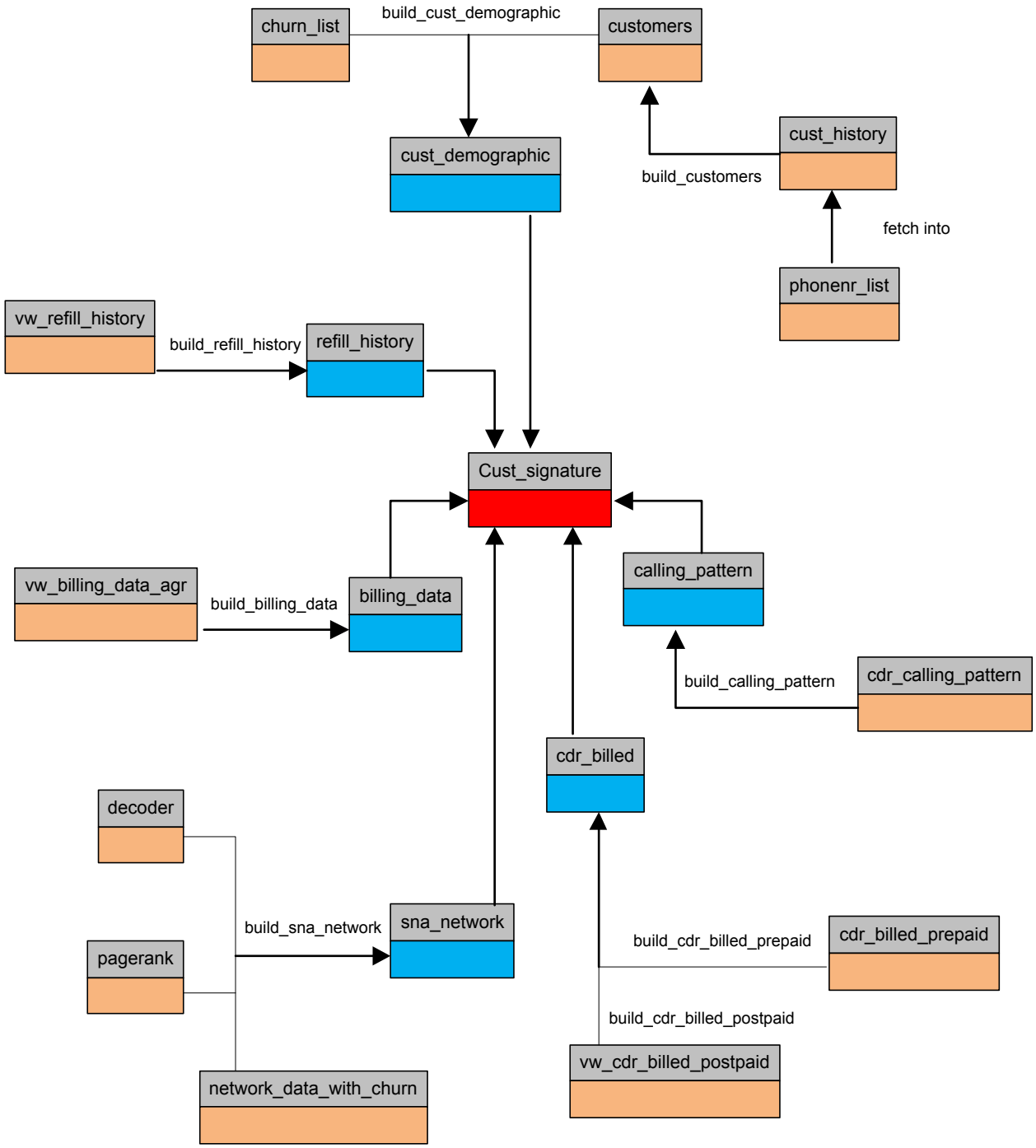

Figure 4-1. The relation model of the customized churn prediction database.

4.2 The Time Aspect

Churn is a causal event hence extracted features have to capture changes in the characteristics and behaviors of customers. The role of a prediction model is then to detect noticeable changes that lead to churn. The classification task of this study requires at least two time periods. They however do not have to be distinct. One period defines the target window from which features are extracted for each customer. The other one is used to label the customers as churn or non-churn.

(Wei & Chiu, 2002) propose a data extraction strategy which consists of not only two but three time periods. They call the foremost time period an observation period *T*. Features that describe the changes in calling pattern of customers are extracted from this observation period. The length of *T* is set to 30 days and divided into a certain number of sub-periods. Following *T* is a retention period *R* which is prior to the prediction period. When the model is applied in reality, the retention period provides the service provider an amount of time to perform retention actions. 14 days are assumed for retention period. After *R*, a prediction period *P* of 7 days is used to define the churn status of customers. Those who are disconnected after this period are churners and those who are still remained in services are non-churners.

Figure 4-2 shows a proposed timeline for this study which consists of a target window and an observation period. All customers that are active at the beginning of the observation period are included in the data set. Monthly aggregated features are extracted for each customer in the target window of three months prior to the observation period. Prepaid customers often refill their calling credit on a monthly basis hence too narrow window will not be able to capture significant changes in their behaviors. Therefore, the target window in this study which is equivalent to the *T* period in (Wei & Chiu, 2002) is lengthened to three months instead of only 30 days. The observation period is used for labeling customers as churn or censoring. Those who churn in this period are labeled as churn and those who don't churn after this period are labeled as censoring. The term *censoring* underlines the fact that the timeline is censored. What happens after that is unknown. A censoring customer will remain in service until he becomes a churner.

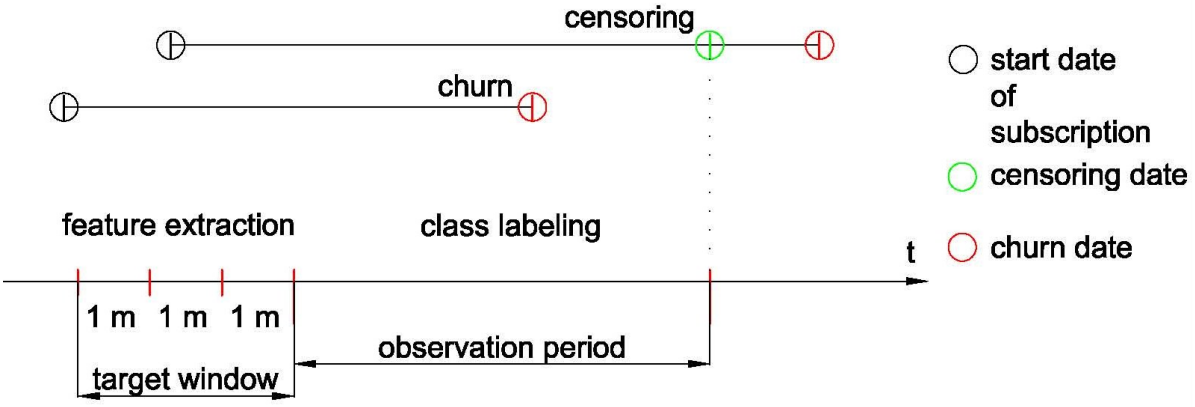

Figure 4-2. The proposed timeline for feature extraction from the database

The advantage of this timeline is that the target window is the same for all customers which makes it convenient for the mobile operator to extract features. The operator can also adjust how many months in advance the prediction is made by changing the observation period. Due

to the rapid changes in the market environment, the prediction models needs to be updated on a regular basis. For example, data collected in the first quarter is used to train a classifier. This classifier is then employed to predict churn in the second quarter. Five months observation period is used for this study. Normally, a shorter period may be more preferable. However the main purpose here is to collect as many churn cases as possible which can only be achieved by lengthening the observation period.

Except for demographics features, continuous features in the other five categories are extracted for each month in the target window. In addition, derived features such as the maximum, the month when the maximum occurs, total, monthly average, and ratio are generated. List of all features can be found in Appendix C. Here below is a brief summation of the features. Table 4-1 sums up the number of features in each category where calling_pattern is the biggest category.

1. ***Demographics***

- Customer's age
- Family size
- Gender
- Postcode
- Land area
- Marital status
- Rate plan
- Subscription type (postpaid or prepaid)
- Payer's age
- Is payer? (customer is the registered payer or not)
- Tenure

2. ***Billing data***

- Number of billed services
- Number of billed products
- Billed amount due to mobile usage
- Discount amount
- Total billed amount
- Ratio of mobile usage w.r.t. total billed amount
- Ratio of discount w.r.t. total billed amount

3. ***Refill history***

- Refill frequency/amount
- Maximum refill frequency/amount
- The month when the maximum refill frequency/amount occur
- The total/average refill frequency/amount

4. ***Calling pattern***

- Inside/outside network & abroad call volume/frequency
- Total originating/terminating call volume/frequency
- Total sending/receiving sms
- Ratio of inside/outside network & abroad w.r.t. total originating call volume/frequency
- Ratio of originating w.r.t. terminating call volume
- Ratio of sending w.r.t. receiving sms

5. CDR billed

- Charged amount due to inside/outside network & abroad calls
- Ratio of inside/outside network & abroad calls w.r.t. total charged amount
- Charged amount due to sms sending inside/outside network & abroad
- Ratio of inside/outside network & abroad sms sending w.r.t. total charged amount
- Total charged amount

6. Network

- Number of neighbors/churn neighbors
- Ratio of churn neighbors w.r.t. neighbors
- Degree centrality
- Total degree centrality of neighbors/churn neighbors
- Ratio of degree centrality of churn neighbors w.r.t. that of neighbors
- Page rank
- Total page rank of neighbors/churn neighbors
- Ratio of page rank of churn neighbors w.r.t. that of neighbors
- Total call volume to/from neighbors/churn neighbors
- Ratio of call volume to/from churn neighbors w.r.t. that of neighbors

Table 4-1. Number of features in each feature category

Group	Number of features
calling pattern	124
network	109
cdr billed	79
billing data	21
demographics	11
refill history	14

4.3 Sampling

The class distribution of the data is highly skewed towards censoring cases. Churn is the class of interest but it is the minority class in the data set. As the consequence, classifiers will predict all instances as the major class in order to maximize the classification accuracy which leads to null prediction system. To overcome this problem, either over- or under-sampling can be applied. Neither of them is consistently better than the other one so the choice depends on the nature of the data. In over-sampling, instances belong to the minority class are replicated, adjusting the class distribution to a desired ratio. In under-sampling, instances belong to the majority class are discarded, reducing the distribution bias.

Under-sampling is used in this study due to two reasons. First is because WEKA has computational limitation and allows only a limited amount of training data. Secondly, oversampling means making copies of instances hence it doesn't provide any new information and it can lead to over-fitting (Burez & Van den Poel, 2009). The classifier learns nothing new via these copies and the estimated accuracy is optimistic. Under-sampling provides more realistic accuracy estimate.

Table 4-2. Statistics of the training and testing sets

Dataset	Number of churn instances	Number of censoring instances
Postpaid training set	2190	2190
Postpaid testing set	828	27932
Prepaid training set	4234	4234
Prepaid testing set	1355	36409

Table 4-2 summarizes the statistics of training and testing sets for post- and prepaid. The amount of training data for prepaid is two times more than for postpaid. A classifier can be tuned so it provides maximum accuracy on the training set by memorizing the data. However it has low generalization capability and will perform poorly on new data (Witten & Frank, 2005). This phenomenon is known as over-fitting. In order to prevent over-fitting, a separated validation set is needed. However often due to the limited amount of available data, cross validation is applied. The role of cross validation is to optimize the model parameters with respect to accuracy and generalization ability.

In this study, the training set is utilized for both training and validating classifiers using k-fold cross validation. The data is divided into k equal folds. Afterwards, k phases of training and validating are carried out. At each phase, one fold is held-out for validation and the rest $k-1$ folds are used for training. The overall performance of a classifier can be estimated as its average performance in k validations. In general, 10-folds cross validation is the most common strategy. When instances are sampled into folds so that the class distribution in each fold remains the same as in the original training set, it is called stratified cross-validation.

After training and validation, the classifiers' performances are evaluated using a testing set. As given in table 4-2, no sampling is applied on the testing sets. This is done purposely so that the testing sets reflect the real world data with highly skewed class distribution. Hence the performance evaluation is realistic but not optimistic. Two data sets are prepared for two observation periods: from 1st July 2010 to 1st December 2010 and from 1st December 2010 to 1st May 2011. Therefore, the target window of the first period includes April, May and June 2010 and of the second period includes September, October and November 2010. The first and half of the second data set are combined and under-sampled into training set. The remaining half of the second data set is used for testing and no sampling is carried out as reported above.

4.4 Feature Selection

As discussed in chapter 3.1, features can be selected using different evaluation functions: distance, information, dependence, consistency and classifier error rate measure. By choosing one representative for each measure, six feature selection methods are examined:

i. Distance measure: Relief
ii. Information measure: InfoGain
iii. Dependence measure: correlation-based feature selection (CFS)
iv. Consistency measure: filter version of Las Vegas algorithm (LVF)
v. Classifier error rate measure: C.4.5 decision tree based wrapper (wr-J48tree), alternating decision tree based wrapper (wr-ADtree), Naïve Bayes based wrapper (wr-Bayes) and logistic regression based wrapper (wr-logistic).

The first two methods evaluate each feature individually and give them scores. Features are ranked according to their scores and the top 20 are chosen. The next three methods search for the best subset of features with respect to the adopted metrics. Thereof, the last method is a wrapper approach which uses a base classifier to assess subsets of features using cross validation. The first four methods output a total of four reduced data sets. The wrapper method alone outputs four reduced data sets since there are four base classifiers. Hence, eight reduced data sets for postpaid and eight for prepaid are generated after the feature selection phase.

Among the original 358 features, the number of selected features by all feature selection methods under each feature category is summed up in tables 4-3 and 4-4 for postpaid and prepaid sector respectively. The category with highest proportion of chosen features is demographics since 47% of its features are chosen in both sectors. Calling pattern has the most chosen features since it is the biggest category of all. Billing data and refill history are equally important in postpaid and prepaid since six features of each category are selected. CDR billed information explains better churn in postpaid than churn in prepaid as 29% of its features are chosen in postpaid but only 14% in prepaid. On the opposite, network features weight heavier in prepaid than in postpaid, 27% of its features are chosen in prepaid, only 13% in postpaid.

Table 4-3. Number and proportion of chosen features in postpaid

Category	Number of chosen features	Proportion of chosen features
demographics	8	47%
calling pattern	39	31%
cdr billed	23	29%
billing data	6	29%
Network	14	13%

Table 4-4. Number and proportion of chosen features in prepaid

Category	Number of chosen features	Proportion of chosen features
demographics	8	47%
refill history	6	43%
calling pattern	41	33%
Network	29	27%
cdr billed	11	14%

Filtering out features that are chosen only once, tables 4-5 and 4-6 list those features which are chosen by two or more methods for postpaid and prepaid sector respectively. Note that the full tables can be found in Appendix B. The top features, LAND_AREA in postpaid and CUST_AGE in prepaid are chosen by all methods except InfoGain. Features that appear in both tables are ten: LAND_AREA, CUST_AGE, RATEPLAN, PAYER_AGE, MARITALSTATUS, GENDER, TENURE, SUM_SMSIN, SUM_TOTAL_DC_NEIBOR, and SUM_TOTAL_PR_NEIBOR. However, they are not chosen by the same methods in post- and prepaid. Seven of them are demographics, SUM_SMSIN belongs to calling pattern category and the last two are network features. It is noticeable that plenty of features connected to out-net call volume and charge appear in both tables. This underlines that customers who make more frequently out-net calls will be quite likely to churn. The increase in out-net calls frequency may be the warning signal first, then later become the cause of churn.

Table 4-5. List of features that are chosen by two or more feature selection methods in postpaid

Feature	Category	CFS	LVF	InfoGain	Relief	wr-Adtree	wr-Bayes	wr-J48tree	wr-logistic	Total
LAND_AREA	demographics	1	1		1	1	1	1	1	7
CUST_AGE	demographics	1			1	1	1	1	1	6
RATEPLAN	demographics	1	1	1	1				1	5
SUM_SMSIN	calling pattern	1		1		1		1		4
PAYER_AGE	demographics	1	1		1					3
IMAX_OUTNET_TCHARGE_RAT	cdr billed				1		1			2
MARITALSTATUS	demographics		1		1					2
SUM_TOTAL_PR_NEIBOR	network			1					1	2
IMAX_S_OUTNET_CHARGE	cdr billed				1				1	2
MAX_OUTNET_FREQ	calling pattern			1					1	2
NUM_NEIBOR2	network		1						1	2
OUTNET_FREQ	calling pattern		1	1						2
GENDER	demographics		1		1					2
AVG_TCHARGE	cdr billed	1		1						2
IMAX_SMSIN	calling pattern		1		1					2
SUM_TOTAL_DC_NEIBOR	network			1			1			2
INNET_TCHARGE_RAT2	cdr billed		1					1		2
AVG_OUTNET_FREQ	calling pattern	1		1						2
TENURE	demographics				1		1			2
MAX_OUTNET_CHARGE	cdr billed	1		1						2

Table 4-6. List of features that are chosen by two or more feature selection methods in prepaid

Feature	Category	CFS	LVF	InfoGain	Relief	wr-Adtree	wr-Bayes	wr-J48tree	wr-logistic	Total
CUST_AGE	demographics	1	1		1	1	1	1	1	7
AVG_TOTALIN_FREQ	calling pattern		1	1		1		1		4
MARITALSTATUS	demographics	1	1		1		1			4
RATEPLAN	demographics	1	1		1			1		4
PAYER_AGE	demographics		1				1		1	3
LAND_AREA	demographics		1		1		1			3
TENURE	demographics		1		1				1	3
GENDER	demographics		1		1		1			3
MAX_OUTNET_TCHARGE_RAT	cdr billed	1					1		1	3
SUM_TOTALIN_FREQ	calling pattern			1				1		2
REFILL_FREQ1	refill history	1				1				2
PAGERANK_RAT2	network							1	1	2
AVG_TOTAL_DC_NEIBOR	network			1					1	2
SUM_SMSIN	calling pattern		1	1						2
IMAX_OUTNET_VOL_RATIO	calling pattern				1				1	2
OUTNET_FREQ_RATIO	calling pattern	1							1	2
IMAX_TOTALIN_FREQ	calling pattern		1		1					2
REFILL_FREQ	refill history	1	1							2
INNET_TCHARGE_RAT1	cdr billed		1		1					2
SUM_DEGREE_CENTRALITY	network		1	1						2
MAX_NUM_CHURN_NEIBOR	network					1			1	2
SUM_TOTAL_DC_NEIBOR	network			1			1			2
SUM_TOTAL_PR_NEIBOR	network		1	1						2

Take a closer look at two of the ten above-listed features by construct a one-level decision tree also known as a decision stump for each feature. Tables 4-7 and 4-8 are example of two decision stumps built for post- and prepaid using the top feature in tables 4-5 and 4-6: LAND_AREA and CUST_AGE. According to table 4-7, among all land areas in the country churn rate in postpaid is the lowest in land area with code 4. Table 4-8 shows that the prepaid customer age 55,5 is a threshold below which the churn likelihood is high and above which the churn risk reduces apparently. Demographics of prepaid customers are often unrecorded. In table 4-8, statistic is also given for the cases when customers' ages are missing. It turns out that churn is minority in that group.

Table 4-7. Decision stump for LAND_AREA, the top feature in the list of chosen features in postpaid

Class distributions	
LAND_AREA = 4	
censoring	churn
0.76	0.23
LAND_AREA != 4	
censoring	churn
0.48	0.51
LAND_AREA is missing	
censoring	churn
0.5	0.5

Table 4-8. Decision stump for CUST_AGE, the top feature in the list of chosen features in prepaid

Class distributions	
CUST_AGE <= 55.5	
censoring	churn
0.33	0.66
CUST_AGE > 55.5	
censoring	churn
0.67	0.32
CUST_AGE is missing	
censoring	churn
0.64	0.35

Demographics features are popular among all methods except InfoGain. Tables 4-9 and 4-10 count the number of times features in each category are chosen by each method. In the top 20 features selected by InfoGain for postpaid sector only one is demographic and none in prepaid. Similar relation is between network features and Relief. In the top 20 features selected by Relief for postpaid sector only two are network features and none in prepaid. Among all methods, InfoGain favors network features the most. It selects 5 and 13 network features in top 20 in post- and prepaid respectively. This implies that network features are those which provide the most information gain with respect to the class. Meanwhile, Relief favors demographics and calling pattern features. LVF tails Relief regarding this. LVF and Relief are similar in the way that they both measure the differences between instances in the data set. Relief actually measures the differences by distances while LVF counts the number of differences. This relation between these two methods and these feature categories indicates that demographics and calling pattern features distinguish instances of different classes the best while keeping instances of the same class stay close to each other.

Table 4-9. Number of times features in each category are chosen by each method in postpaid

	CFS	LVF	InfoGain	Relief	wr-Adtree	wr-Bayes	wr-J48tree	wr-logistic	Total
billing data	2	2		1		1			**6**
calling pattern	11	7	9	7	2	1	6	3	**46**
cdr billed	7	3	5	3		1	6	3	**28**
demographics	4	6	1	7	2	3	2	3	**28**
network	3	1	5	2	1	3		2	**17**
Total	**27**	**19**	**20**	**20**	**5**	**9**	**14**	**11**	**125**

Table 4-10. Number of times features in each category are chosen by each method in prepaid

	CFS	LVF	InfoGain	Relief	wr-Adtree	wr-Bayes	wr-J48tree	wr-logistic	Total
refill history	2	3			1		1	1	**8**
calling pattern	11	9	7	8	2		5	7	**49**
cdr billed	2	1		5		1	3	2	**14**
demographics	3	7		7	1	5	2	3	**28**
network	5	3	13		1	3	5	5	**35**
Total	**23**	**23**	**20**	**20**	**5**	**9**	**16**	**18**	**134**

The calling pattern features are favored by CFS the most since 11 features each are chosen by CFS in post- and prepaid. CFS and LVF are the only two that select features from all categories in both post- and prepaid. As can be seen in equation (3.1-6), the value of the CFS heuristic function increases as the correlation between features decreases. In general, features from different categories are less correlated to each other than those belong to the same category. Therefore, CFS tends to choose features from as many categories as possible in order to maximize its heuristic function. Meanwhile, LVF's goal is to find the smallest subset which fulfils a predefined inconsistency rate. Adding a redundant feature that is correlated to another feature in the subset will not help to decrease the inconsistency rate and also make the subset bigger. Hence similar to CFS, LVF exploits as many different feature categories as possible, searching for relevant but uncorrelated features. Besides that, CFS and LVF are the methods that utilize billing data and refill history the most. They take a total of 4 and 5 out of 6 and 8 selected billing data and refill history features respectively.

According to CFS, the relevancy merit of the best subset found is 0,118 in postpaid and 0,188 in prepaid. The higher merit of the prepaid subset indicates that the features are more correlated to the class and less correlated to each other. The consistency rate of the best subset found according to LVF is 1 in postpaid and 0,979 in prepaid. The consistency rate equals to 1 of the postpaid subset means that the training data it defines is completely consistent and doesn't contain any inconsistent instance pair. In other words, if it exists any two customers whose values of all features are matched then their class labels are also the same.

Table 4-11. The merit of the wrappers in both post- and prepaid

	Classification error	
	Postpaid	Prepaid
wr-Adtree	0,333	0,269
wr-Bayes	0,327	0,267
wr-J48tree	0,333	0,263
wr-logistic	0,316	0,260

As can be observed in tables 4-9 and 4-10, wrappers tend to choose smaller subsets than the filters. It is curiously that wr-Bayes doesn't choose any calling pattern features in prepaid although this is the largest category and the most popular among other feature selection methods. Table 4-11 reveals the merit of the wrappers in the form of classification error after 5-folds cross-validation. It is understandable that the classification error in prepaid is lower than in postpaid because the training data is approximately two times more in prepaid than in postpaid.

4.5 Summary

This chapter describes the data preparation process, from data acquisition, data sampling to data dimension reduction. The data can be generated from a special database built for the purpose of churn prediction in this study. This database is defined by a relation model that connects different sources in the operator's data warehouse into one center table which contains customers and their associated features. The features belong to six categories: demographics, billing data, refill history, calling pattern, CDR billed and network features. The data can be extracted according to the timeline demonstrated in figure 4-2. Four data sets are prepared. One training set and one testing set each for post- and prepaid. The data is highly skewed towards censoring cases. Hence under-sampling is conducted to create balanced training set of 4380 instances for postpaid and 8464 instances for prepaid. They are also utilized for validation using 10-folds cross validation. The testing sets contain about 30000 instances for postpaid and 40000 instances for prepaid. No sampling is applied on the testing sets. This is done purposely so that the testing sets reflect the real-world data with highly skewed class distribution.

After the work of data transforming and sampling has been done, different feature selection methods are carried out to search for the most critical indicators that influence churn. Eight reduced data sets each for post- and prepaid are created using two individual feature selection filters, two feature subset selection filter and four wrappers. The results of feature selection reveal the relevancy of each feature category and each individual feature to the target concept which is churn.

The category with highest proportion of chosen features is demographics since 47% of its features are chosen in both sectors. Calling pattern has the most chosen features since it is the biggest category of all. Billing data and refill history are equally important in postpaid and prepaid since six features of each category are selected. CDR billed information explains better churn in postpaid than churn in prepaid as 29% of its features are chosen in postpaid but only 14% in prepaid. On the opposite, network features weight heavier in prepaid than in postpaid, 27% of its features are chosen in prepaid, only 13% in postpaid. It is noticeable that plenty of features connected to out-net call volume and charge are chosen in both post- and prepaid. This indicates that customers who make more frequently out-net calls will be quite likely to churn. The increases in out-net calls frequency may be first the warning signal, and then later become the cause of churn.

Among all methods, InfoGain favors network features the most while Relief favors demographics features. CFS and LVF utilize the feature categories the best since they select features from all categories. Wrappers tend to choose smaller subsets than the filters. The merits of the wrappers are estimated in the form of classification error after 5-folds cross-validation. The classification error in prepaid is lower than in postpaid which may explained by the fact that the training data is approximately two times more in prepaid than in postpaid.

5 Modeling and Evaluation

In this chapter, churn prediction models in the form of classifiers are built by employing four machine learning algorithms: C.45 decision trees, alternating decision tree, Naïve Bayes and logistic regression. C4.5 is a decision tree builder using entropy for splitting nodes and adopting pruning to increase generality power. Alternating decision tree enhances the performance and interpretability of decision tree by adopting boosting techniques. Naïve Bayes is a probability approach based on Bayes's theorem. Logistic regression is a linear parametric model which outputs continuous prediction value.

The data prepared as reported in the previous chapter is used for three purposes: classifier training, parameter tuning and performance evaluation. The training data is used for classifier building. 10-folds cross validation is applied on the training data to find the optimal parameter values and feature subsets. Chapters from 5.1 to 5.4 give and discuss the results for each learning algorithm one after another. In chapter 5.5, the best representative of each learning algorithm is tested on the prepared testing set in order to find out which one is the best of all. The classifiers' performances are evaluated according to three metrics: the overall accuracy, the area under ROC curve and prediction confidence level.

The analysis part of this study is conducted using MATLAB (MATLAB, 2009) and WEKA (Hall, Frank, Holmes, Pfahringer, Reutemmann, & Witten, 2009)

5.1 C4.5 Decision Tree

WEKA implements a commercial version of C.4.5 decision tree called J48tree. After a J48 fully grown decision tree is constructed, pruning is carried out using 10 folds cross-validation on training sets. Post-pruning is tested with the confidence factor cf ranging from 0,0005 to 0,15. Meanwhile, the minimum number of instances per node min was held at 2. The lower the confidence factor is, the more pruning it will be. The confidence factor relates to how the error of a node is estimated as described in chapter 3.2. High confidence factor gives a narrow confidence interval for the error estimate and low confidence factor gives a large confidence interval. Therefore, it is more likely that a node will be pruned due to high error when the confidence factor is low. Pre-pruning is tested by varying the minimum number of instances per node min between 2 and 96. Meanwhile, the confidence factor cf was held constant at 0,25 to minimize post-pruning. The higher the minimum number of instances per node is, the more pruning will be performed.

Figures 5-1 and 5-2 display the effect of post- and pre-pruning on the performances of J48tree classifiers. The performances are based on the overall accuracy (OA) and area under ROC curve (AUC). J48tree classifiers are trained on the full training set for postpaid and reduced sets after five feature selection methods: Relief, InfoGain, LVF, CFS and wr-J48tree. Figures 5-3 and 5-4 displays the same results for prepaid data.

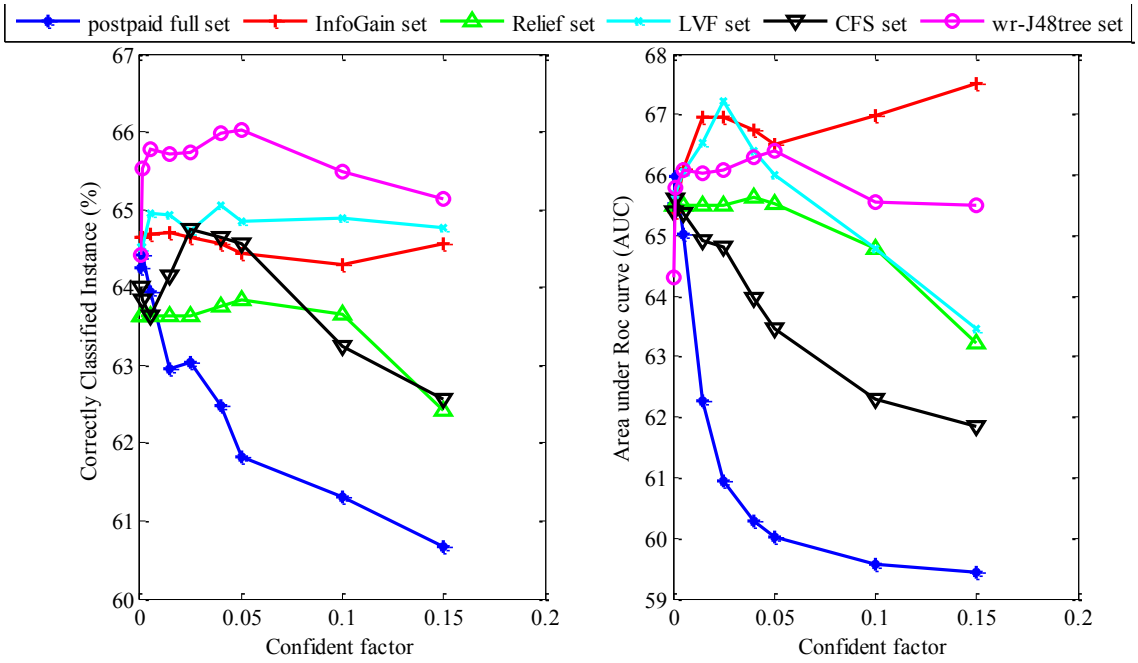

Figure 5-1. Cross-validation accuracy and AUC of J48tree on postpaid data as the confident factor is varied

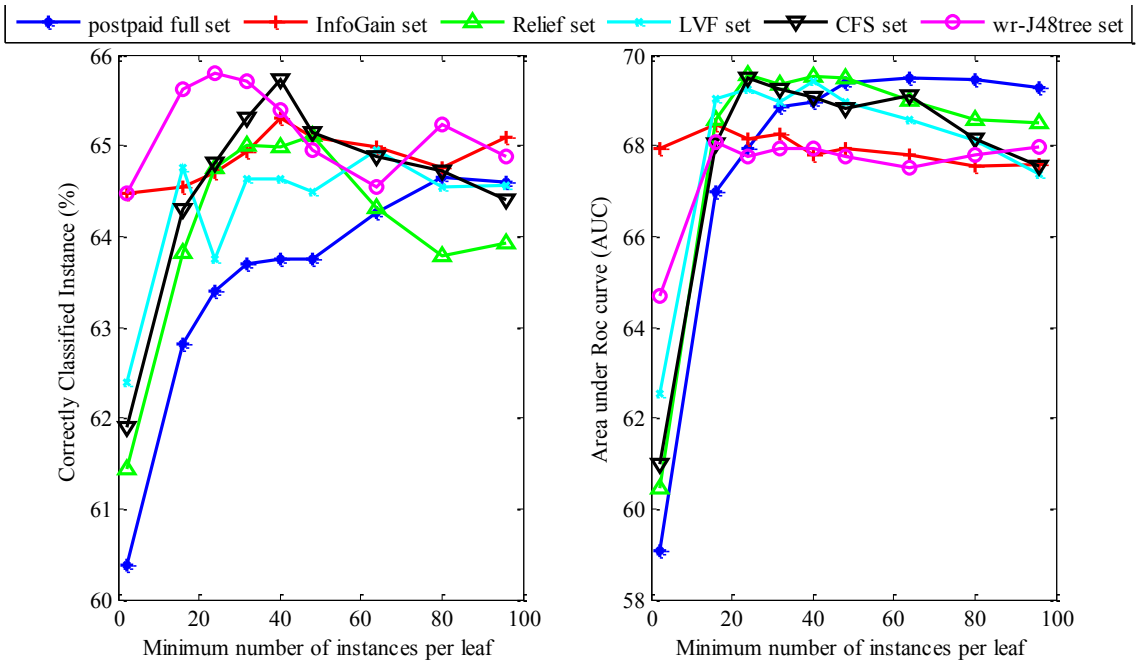

Figure 5-2. Cross-validation accuracy and AUC of J48tree on postpaid data as the minimum number of instances per leaf is varied

According to figure 5-1, the highest OA of 66,03% is achieved by J48tree classifier trained on wr-J48tree set at $cf = 0{,}05$. The highest AUC of 67,5% is achieved by J48tree classifier trained on InfoGain set at $cf = 0{,}15$. The full postpaid set produces worst performances with respect to both OA and AUC. The wr-J48tree classifier like the other classifiers also undergoes the effect of over-fitting. Its performance increases as the confidence factor increased up to 0,05 at a peak of 66,03% OA and 66,41% AUC. Above 0,05, both OA and AUC of the classifier decrease due to over-fitting. When the minimum number of instances varies as showed in figure 5-2, J48tree classifier trained on postpaid CFS set performs the best with 65,73% OA and 69,08% AUC at $min = 40$. Below $min = 24$, all classifiers experiences over-fitting. Above $min = 24$, their performances based on AUC are stable.

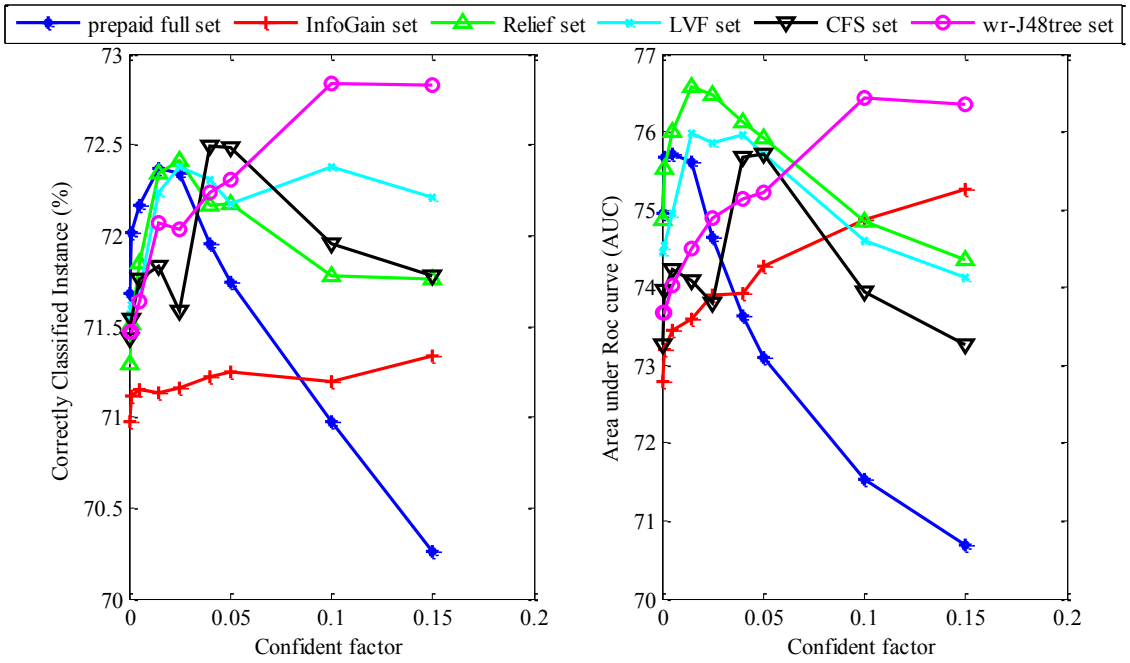

Figure 5-3. Cross-validation accuracy and AUC of J48tree on prepaid data as the confident factor is varied

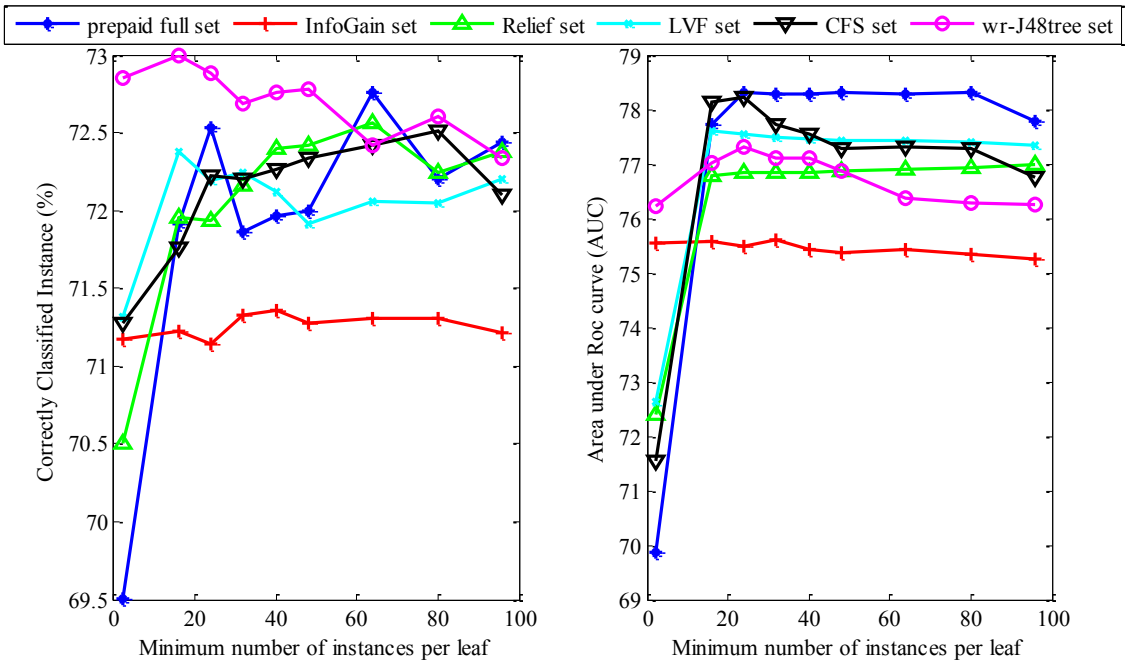

Figure 5-4. Cross-validation accuracy and AUC of J48tree on prepaid data as the minimum number of instances per leaf is varied

Figure 5-3 displays the effect of over-fitting on classifiers trained on prepaid training sets. The effect on wr-J48tree classifier is not clearly. However, values of OA and AUC are similar in both points $cf = 0{,}1$ and $cf = 0{,}15$ so they are probably at the top and heading downward. Above $cf = 0{,}05$, all classifiers undergo over-fitting effect. Overall, wr-J48tree classifier is the best with 72,84% OA and 76,44% AUC at $cf = 0{,}1$. It can be seen in figure 5-4 that below $min = 20$, all classifiers trained on prepaid data experience over-fitting. Above $min = 20$, their performances based on AUC are stable. J48tree classifier trained on InfoGain set is clearly the worst. The highest OA 72,99% is achieved by J48tree classifier trained on wr-J48tree set at $min = 16$. The highest AUC 78,31% is achieved by J48tree classifier trained on the full prepaid set at $min = 24$.

The OA and AUC of postpaid classifiers range from 58% to 70% while of prepaid from 69% to 79%. The difference is about 10%. In both sectors higher level of accuracy for all classifiers is achieved by selecting the right minimum number of instances per leaf. Two best J48tree classifiers for post- and prepaid are chosen. Since the higher value of AUC is more desirable, the selected classifiers have the overall best performances based on AUC. The parameter for each classifier is chosen at the point where the value of OA is the highest. In postpaid, J48tree classifiers trained on postpaid CFS set performs the best with 65,73% OA and 69,08% AUC at $min = 40$. In prepaid, J48tree classifiers trained on prepaid full set performs the best with 72,76% OA and 78,29% AUC at $min = 64$.

Table 5-1. J48tree classifier trained on postpaid CFS set with $min = 40$ and $cf = 0,05$

```
SUM_SMSIN <= 35
|  AVG_OUTNET_FREQ <= 0: censoring (531.0/66.0)
|  AVG_OUTNET_FREQ > 0
|  |  CUST_AGE <= 46: churn (141.0/59.0)
|  |  CUST_AGE > 46: censoring (983.0/349.0)
SUM_SMSIN > 35
|  AVG_OUTNET_FREQ <= 30.333333
|  |  LAND_AREA = 0: churn (6.0/2.0)
|  |  LAND_AREA = 1: churn (584.0/240.0)
|  |  LAND_AREA = 2
|  |  |  AVG_OUTNET_FREQ <= 11.666667: censoring (42.0/12.0)
|  |  |  AVG_OUTNET_FREQ > 11.666667: churn (68.0/26.0)
|  |  LAND_AREA = 3
|  |  |  AVG_OUTNET_FREQ <= 9.333333: censoring (48.0/14.0)
|  |  |  AVG_OUTNET_FREQ > 9.333333
|  |  |  |  IMAX_NUM_NEIBOR <= 2
|  |  |  |  |  MYSUM1 <= 2320.43: censoring (57.52/22.52)
|  |  |  |  |  MYSUM1 > 2320.43: churn (53.48/18.0)
|  |  |  |  IMAX_NUM_NEIBOR > 2: churn (40.0/13.0)
|  |  LAND_AREA = 4: censoring (65.0/16.0)
|  |  LAND_AREA = 5: censoring (200.0/98.0)
|  AVG_OUTNET_FREQ > 30.333333: churn (1561.0/483.0)
```

The advantage of decision trees is they can be presented graphically and interpreted. Table 5-1 displays a J48tree classifier trained on postpaid CFS set with $min = 40$ and $cf = 0,05$. The largest node of this tree with 1561 instances and 70% of them are churners. Customers belong to this node are those who receive more than 35 text messengers on the most recent month and make on average more than 30 out-net phone calls per month in the latest three months. In contrast, customers who receive less than 35 text messengers on the most recent month and don't make any out-net phone calls in the latest three months form a group of 531 non-churners with the probability of 88%.

Statistics of the training data show that customers receive on average 136 text messengers and make 37 out-net phone calls per month. Hence the threshold values of these quantities that draw the line between churners and non-churners as mentioned above: 35 text messengers and 30 out-net phone calls are both under the averages. It is curious that even if a customer makes fewer out-net phone calls per month than the population's average, he is still in churn risk. The mean of 136 text messengers per month is rather high. A closer look at the data reveals that there is a certain number of customers who receive remarkably many text messengers per month. This group lifts the population's mean upp.

Table 5-2. J48tree classifier trained on prepaid full set with $min = 64$ and $cf = 0,001$

```
MAX_FROM_NEIBOR_VOL <= 73: censoring (2446.0/389.0)
MAX_FROM_NEIBOR_VOL > 73
|  AVG_SMSIN <= 16.333333
|  |  CUST_AGE <= 56
|  |  |  INNET_TCHARGE_RAT1 <= 0.711228
|  |  |  |  ISPAYER = 1: churn (704.23/256.24)
|  |  |  |  ISPAYER = 10
|  |  |  |  |  OUTNET_FREQ <= 2: censoring (230.75/94.88)
|  |  |  |  |  OUTNET_FREQ > 2: churn (226.24/81.66)
|  |  |  INNET_TCHARGE_RAT1 > 0.711228
|  |  |  |  IMAX_SMS_OUTIN_RATIO <= 2: censoring (303.27/111.79)
|  |  |  |  IMAX_SMS_OUTIN_RATIO > 2: churn (89.71/38.73)
|  |  CUST_AGE > 56
|  |  |  AVG_TOTAL_DC_CHURN_NEIBOR <= 0.000057: censoring (698.33/248.26)
|  |  |  AVG_TOTAL_DC_CHURN_NEIBOR > 0.000057: churn (69.47/28.94)
|  AVG_SMSIN > 16.333333
|  |  MAX_ABROAD_TCHARGE_RAT <= 0.984246: churn (3674.0/974.0)
|  |  MAX_ABROAD_TCHARGE_RAT > 0.984246: censoring (26.0/6.0)
```

Table 5-2 shows a J48tree classifier trained on prepaid full set with $min = 64$ and $cf = 0,005$. Standing out in this tree is a node of 2446 non-churners with the probability of 84% who receive less than 73 seconds of phone calls in a month for the latest three months. The value of 73 seconds per month is really low, indicates that the customers use their phone very little. The largest node of churners contains 3674 churners with the probability of 73%. The characteristics of a customer who belongs to this group are: he receives more than 73 seconds of phone calls and on average more than 16 text messengers in a month for the latest three months and spends no more than 98% of his total expense on international calls per month.

The value 16 text messengers is below the population's mean of 47 text messengers. This value is even lower in prepaid than in postpaid which verifies the known fact that prepaid customers are more sensitive than postpaid customers regarding churn. By receiving more than 35 text messengers, a postpaid customer becomes more likely to churn while for a prepaid customer, the required number of messenger is only 16. Note that for the sake of simplicity, nominal features RATEPLAN and MARITALSTATUS have been excluded from tables 5-1 and 5-2.

5.2 Alternating Decision Tree

In the same manner as for J48tree classifiers, the performances of ADtree classifiers are tested by varying its parameter, the number of boosting iterations nb from 5 to 25. Figures 5-5 and 5-6 display the results based on OA and AUC for post- and prepaid data. In postpaid, ADtree classifiers trained on CFS and LVF sets are in overall better than the others. The highest OA 67,03% at $nb = 7$ and AUC 73,04% at $nb = 20$ are both achieved using CFS set. In prepaid, the classifier trained on LVF set provides the highest OA 73,35% at $nb = 25$ and the one trained on the full set gives the highest AUC 81,19% at $nb = 25$. ADtree classifier trained on InfoGain set performs the worst in both post- and prepaid.

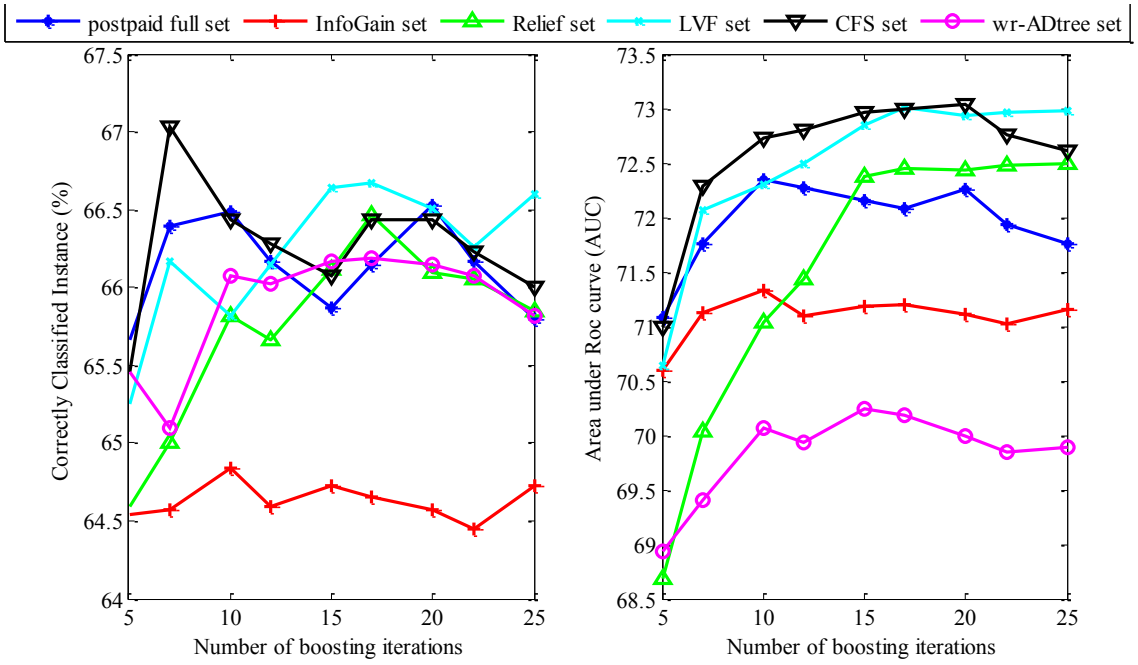

Figure 5-5. Cross-validation accuracy and AUC of ADtree on postpaid data as the number of boosting iterations is varied

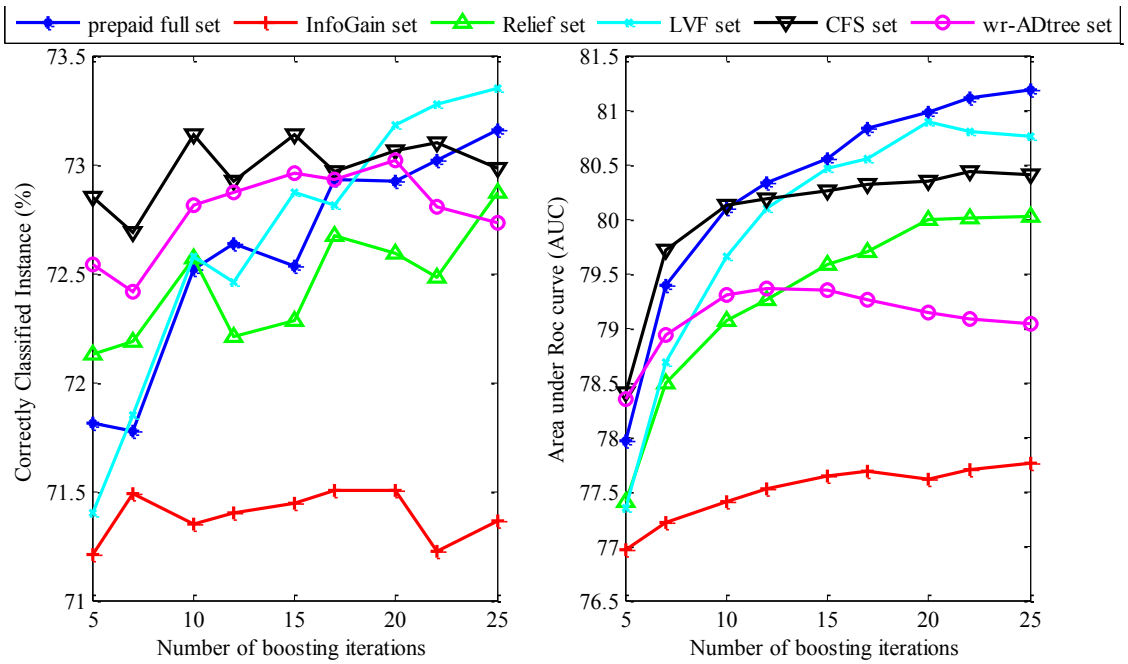

Figure 5-6. Cross-validation accuracy and AUC of ADtree on prepaid data as the number of boosting iterations is varied

Regarding the trade-off between training time and performance, more than 15 boosting iterations seems to be unnecessary. The performances of classifiers increase up to $nb = 15$. After that, they maintain quite stable. More boosting doesn't seem to improve the performances significantly compare to the training time which becomes quite longer. The stability underlines an advantage of ADtrees that they aren't as sensitive to over-fitting as C4.5 tree. More boosting generates a bigger tree which explains the data even better. The reason is that an ADtree doesn't divide and distribute instances downward to nodes as C4.5 tree does. When a sub-tree is added, it takes into account the weights of all available instances in the data set.

Based on the same reasoning for J48tree, the two best ADtree classifiers are chosen. In postpaid, ADtree classifier trained on postpaid CFS set performs the best with 67,03% OA and 72,29% AUC at $nb = 7$. In prepaid, ADtree classifier trained on prepaid full set performs the best with 73,16% OA and 81,19% AUC at $nb = 25$.

Yet another advantage of AD trees is that they are less complex and smaller compared to approximately equally accurate C4.5 trees. Figure 5-7 shows a ADtree classifier trained on postpaid CFS set with $nb = 7$. In an ADtree, each base rule that consists of a decision node and two prediction nodes can be interpreted individually. The higher positive value of a prediction node indicates higher churn risk and the lower negative value indicates higher possibility that customers stay. Figure 5-7 reveals that the churn likelihood of a customer is high in rate plan G4 but is low in rate plan G34. Other indicators of churn are such as customer's age younger than 51,5 and monthly average out-net calls frequency higher than 42,83. Let's recall that this variable AVG_OUTNET_FREQ also appears in the C4.5 tree built for postpaid data in the previous chapter. Its population's mean is 37. Hence a churner according to the ADtree is a customer who makes more out-net phone calls per month than all customers do on average.

Take an example of a customer with the following characteristics: he is younger than 51,5, uses G4 and makes more than 42,83 out-net calls per month on average in the latest three month. His churn score according to figure 5-7 will be $0{,}142 + 0{,}817 + 0{,}192 + 0{,}165 = 1{,}316$. A positive value denotes that a customer is classified as churner. Substitute this value into equation (3.3-2) gives that the total weight of churners is 14 times higher than of non-churners associate with this path. In other words, the chances are 93% that this customer is a churner.

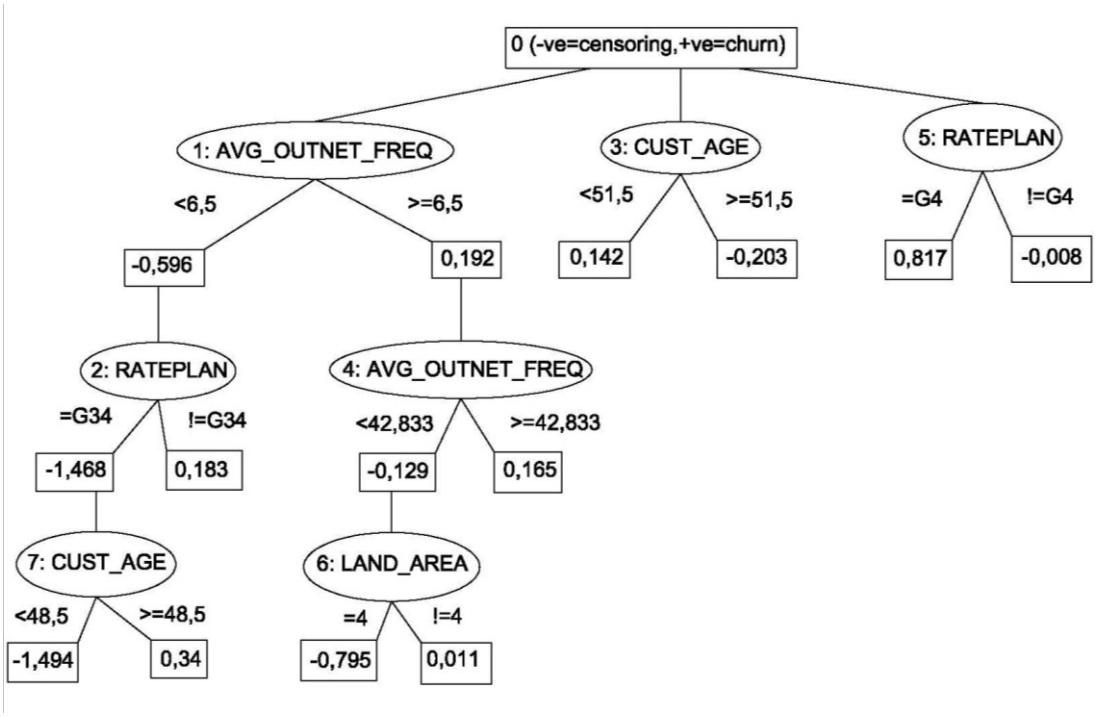

Figure 5-7. ADtree classifier trained on postpaid CFS set with $nb = 7$

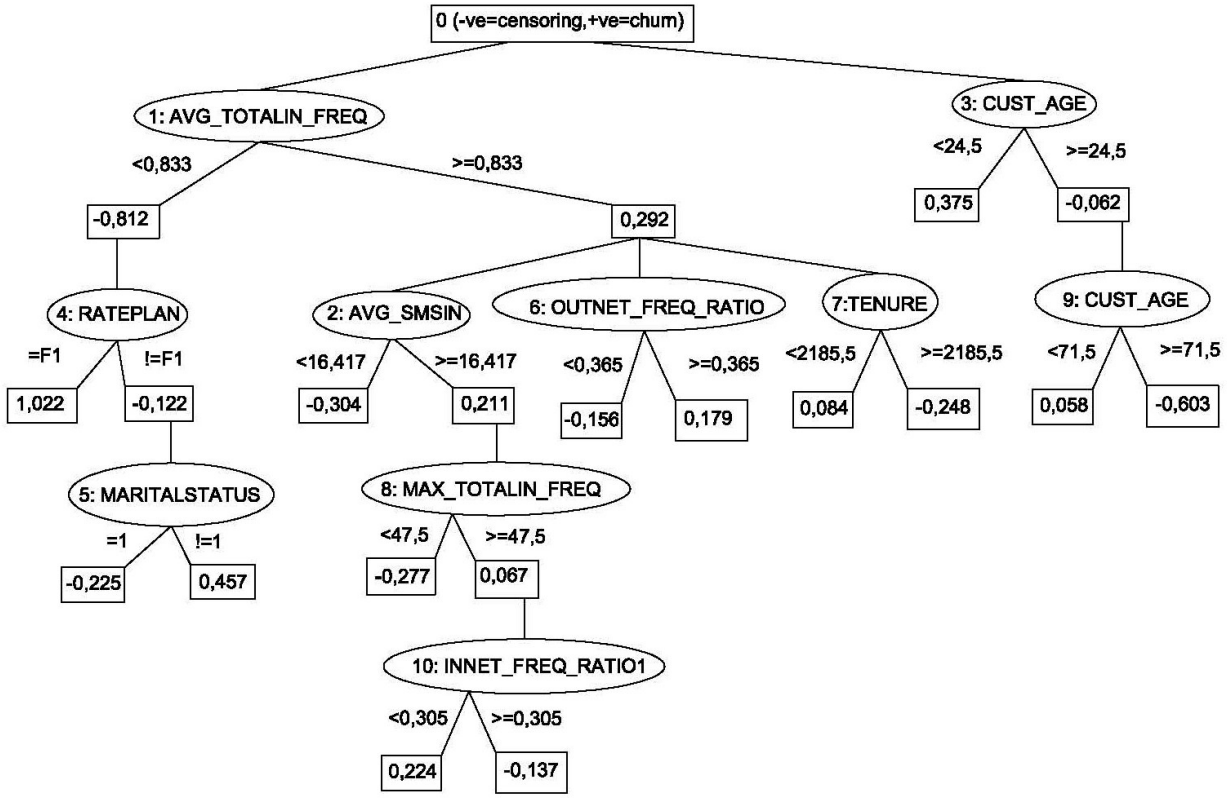

Figure 5-8. ADtree classifier trained on prepaid full set with $nb = 10$

Figure 5-8 displays an ADtree classifier trained on prepaid full set with $nb = 10$. If only customer's age is considered, nodes 3 and 9 provide relationships between different age groups and churn. Customers younger than 24,5 have prediction value 0,375 so the churn likelihood according to equation (3.3-2) is 68%. Those who are older than 24,5 and younger than 71,5 score $-0{,}062 + 0{,}058 = -0{,}004$ corresponds to 50% churn likelihood. Customers older than 71,5 score -0,603 which means churn probability of 23%. Summing up, high age customers are less likely to churn while young customers are more likely to churn, especially those who are under 24,5.

Interaction between features can be read from an AD tree. Take a look closer at nodes 1 and 4. Customers of rate plan F1 have quite high churn probability. However, given that a customer receives on average less than about one phone call per month in three month reduce the churn score about $1{,}022 - 0{,}812 = 0{,}21$ or 30%. The final prediction score is the sum of the values of prediction nodes. Therefore nodes that have high positive values or high negative values contribute the most to the final churn score.

The index in front of each decision node in figure 5-8 indicates in which boosting iteration the node is added. Lower indices correspond to more influential nodes that are added earlier in the boosting process (Freund & Mason, 1999). Note that the two features AVG_OUTNET_FREQ and CUST_AGE are quite relevant since they are the first and third nodes that are added in both post- and prepaid ADtree.

5.3 Naïve Bayes

Figure 5-9 illustrates the performances of Naïve Bayes classifiers trained on reduced data sets after different feature selection methods. The feature subset selection methods LVF, CFS and wrapper produce slightly better classifiers than the individual feature selection InfoGain and Relief. Naïve assumption presumes that all features are independent. Therefore, redundant features will affect the performance of Naïve Bayes. The poorer performance of InfoGain and Relief can be explained by the fact that they don't take into account interdependence among features. Naive Bayes classifiers trained on wr-Bayes sets have the highest OA (68,08%;73,36%) and AUC (74,44%;79,05%) in post- and prepaid respectively.

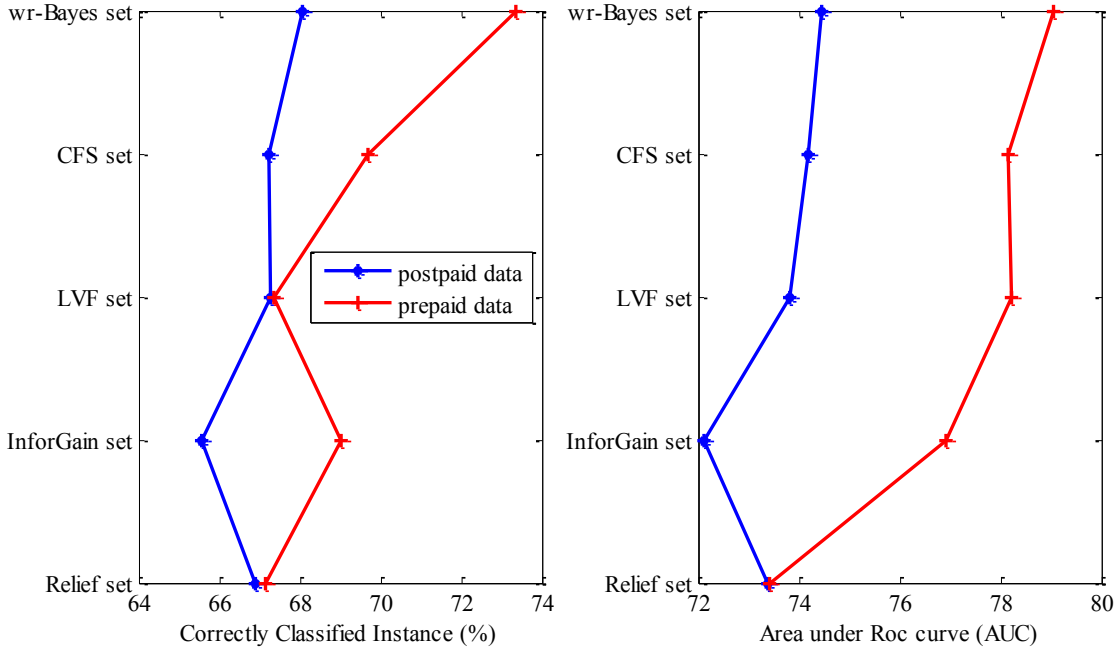

Figure 5-9. Classification accuracy and AUC of Naïve Bayes on reduced post- and prepaid data sets after feature selection

Lists of features that are chosen of wr-Bayes for post- and prepaid can be found in Appendix B. The corresponding classifiers are showed in tables 5-3 and 5-4. One of the features that appear in both tables is CUST_AGE. The average age of churners in postpaid is 45,6 while in prepaid is 32. This is understandable since young people are the majority of all prepaid customers. For comparison, the average age of postpaid customers in the training data is 49 while of prepaid customers is 36. Hence the churners in both post- and prepaid are under the population' average age.

In postpaid, land area 1 is the only one which has more churners than non-churners. In prepaid, the churners are more than non-churners in every land areas but the most in area 1 with 65% churn proportion. The features: MAX_OUTNET_FREQ_RATIO in postpaid and MAX_OUTNET_TCHARGE_RAT in prepaid are correlated. They imply that postpaid churners make almost half of their phone calls out-net and prepaid churners spend more than half of their expense on out-net calls on the top month. Network features: MAX_NUM_NEIBOR, SUM_TOTAL_DC_NEIBOR and PAGERANK indicate that churners are well-connected to the social network. They either have many neighbors, well-connected neighbors or are well-connected themselves.

Table 5-3. Naïve Bayes classifier trained on postpaid wr-Bayes training set

Naive Bayes Classifier

Attribute	Class censoring (0.5)	churn (0.5)
CUST_AGE		
mean	52.8437	45.5743
std. dev.	15.3051	13.3264
LAND_AREA		
0	38.0	25.0
1	1177.0	1341.0
2	193.0	174.0
3	311.0	283.0
4	136.0	43.0
5	341.0	330.0
TENURE		
mean	2323.4092	2030.0355
std. dev.	1766.7629	1725.2368
NUM_SERVICE		
mean	3.2113	3.4502
std. dev.	3.093	2.7364
MAX_OUTNET_FREQ_RATIO		
mean	0.3415	0.4745
std. dev.	0.2626	0.2242
IMAX_OUTNET_TCHARGE_RAT		
mean	1.7521	1.9548
std. dev.	0.8238	0.8205
MAX_NUM_NEIBOR		
mean	28.658	44.9667
std. dev.	31.606	38.3593
IMAX_TOTAL_DC_NEIBOR		
mean	1.6836	1.9333
std. dev.	1.0402	0.8205
SUM_TOTAL_DC_NEIBOR		
mean	0.3311	0.4921
std. dev.	0.2638	0.2113

Table 5-4. Naïve Bayes classifier trained on prepaid wr-Bayes training set

Naive Bayes Classifier

Attribute	Class censoring (0.5)	churn (0.5)
CUST_AGE		
mean	42.5699	32.084
std. dev.	19.4864	15.6107
GENDER		
1	959.0	1516.0
10	858.0	1410.0
LAND_AREA		
0	132.0	62.0
1	822.0	1513.0
2	166.0	257.0
3	315.0	477.0
4	97.0	118.0
5	291.0	503.0
MARITALSTATUS		
1	2417.0	1311.0
10	1.0	1.0
100	38.0	61.0
1000	1.0	1.0
10000	15.0	23.0
100000	149.0	193.0
1000000	24.0	38.0
10000000	89.0	30.0
100000000	711.0	779.0
1000000000	793.0	1805.0

10000000000	4.0	3.0
PAYER_AGE		
mean	48.8934	43.3169
std. dev.	16.8634	13.5223
MAX_OUTNET_TCHARGE_RAT		
mean	0.2774	0.581
std. dev.	0.3515	0.3236
PAGERANK		
mean	0.7974	1.8595
std. dev.	1.1123	1.3576
IMAX_TO_NEIBOR_VOL		
mean	1.1115	1.8106
std. dev.	1.1667	0.9404
SUM_TOTAL_DC_NEIBOR		
mean	0.2392	0.5178
std. dev.	0.2886	0.2604

5.4 Logistic Regression

Figure 5-10 illustrates the performances of Logistic classifiers trained on reduced data sets after different feature selection methods. The best performance in postpaid is obtained by logistic classifier trained on wr-logistic set with 68,08% OA and 74,44% AUC. In prepaid, logistic classifier trained on wr-logistic set gives the highest OA of 73,82% and the one trained on LVF set gives the highest AUC of 80,63%.

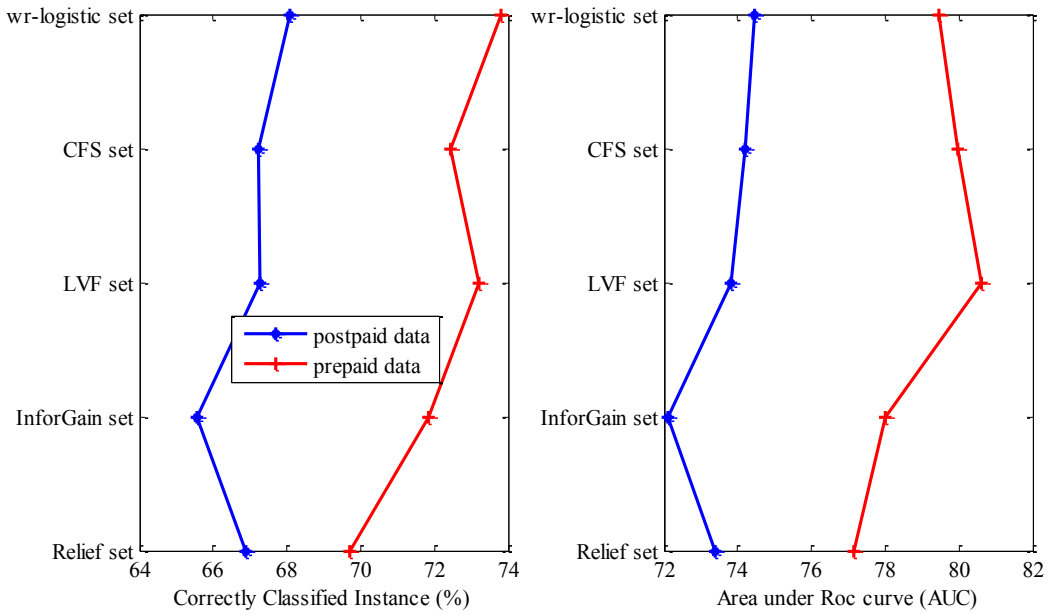

Figure 5-10. Classification accuracy and AUC of Logistic on reduced post- and prepaid data sets after feature selection

Tables 5-5 and 5-6 display the logistic classifiers trained on wr-logistic postpaid set and LVF prepaid set respectively. The value associate with each variable is the weight of that variable. A positive weight in these tables indicates a negative relation between a variable and churn. In other words, the increase in value of that variable will reduce the churn likelihood. In contrast, a negative weight indicates a positive relation between a variable and churn. Hence, the increase in value of that variable will increase the churn likelihood. The higher the absolute value of a weight is, the more relevant the corresponding variable is. Logistic classifiers underline the relevant of CUST_AGE as the other classifiers did. They imply that older

customers are less likely to churn than younger customers. RATEPLAN variables have quite higher weights than the other variables therefore a further discussion about rate plans will be provided in the following text.

Table 5-5. Logistic regression classifier trained on postpaid wr-logistic training set

```
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                        Class
Variable                              censoring
================================================
CUST_AGE                                0.0314
LAND_AREA=0                             0.1207
LAND_AREA=1                            -0.0818
LAND_AREA=2                            -0.0503
LAND_AREA=3                              0.001
LAND_AREA=4                             1.1317
LAND_AREA=5                            -0.1687
RATEPLAN=G17                           -0.3427
RATEPLAN=G35                           21.6656
RATEPLAN=G1                           -71.6332
RATEPLAN=G20                                 0
RATEPLAN=G36                           30.4571
RATEPLAN=G15                           -0.5246
RATEPLAN=G21                                 0
RATEPLAN=G22                                 0
RATEPLAN=G12                           -0.5934
RATEPLAN=G9                            -0.7772
RATEPLAN=G34                            3.2112
RATEPLAN=G33                            2.5105
RATEPLAN=G23                                 0
RATEPLAN=G11                           -0.7302
RATEPLAN=G5                            -1.4984
RATEPLAN=G38                           81.4132
RATEPLAN=G8                            -0.9658
RATEPLAN=G31                            0.5722
RATEPLAN=G16                           -0.3654
RATEPLAN=G37                           77.7422
RATEPLAN=G24                                 0
RATEPLAN=G13                           -0.5882
RATEPLAN=G25                                 0
RATEPLAN=G26                                 0
RATEPLAN=G6                            -1.2845
RATEPLAN=G30                            0.0687
RATEPLAN=G29                            0.0513
RATEPLAN=G27                                 0
RATEPLAN=G2                           -13.5335
RATEPLAN=G3                           -66.8587
RATEPLAN=G4                            -2.3614
RATEPLAN=G19                           -0.1596
RATEPLAN=G7                            -0.9973
RATEPLAN=G18                           -0.2831
RATEPLAN=G32                            0.9634
RATEPLAN=G28                                 0
RATEPLAN=G10                           -0.7718
RATEPLAN=G14                           -0.5665
OUTNET_FREQ_RATIO1                     -0.7423
MAX_SMS_OUTIN_RATIO                     0.1639
MAX_OUTNET_FREQ                        -0.0037
MAX_OUTNET_TCHARGE_RAT                 -0.2333
IMAX_ABROAD_CHARGE                      0.0752
IMAX_S_OUTNET_CHARGE                    -0.018
NUM_NEIBOR2                            -0.0013
SUM_TOTAL_PR_NEIBOR                          0
Intercept                              -0.3957
```

Table 5-6. Logistic regression classifier trained on prepaid LVF training set

```
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                          Class
Variable                                censoring
================================================
CUST_AGE                                   0.0244
GENDER                                    -0.0596
LAND_AREA=0                                0.2706
LAND_AREA=1                               -0.1108
LAND_AREA=2                                0.0797
LAND_AREA=3                                0.1704
LAND_AREA=4                                0.2994
LAND_AREA=5                               -0.1058
MARITALSTATUS=1                            0.3365
MARITALSTATUS=10                                0
MARITALSTATUS=100                         -0.0667
MARITALSTATUS=1000                              0
MARITALSTATUS=10000                        -0.885
MARITALSTATUS=100000                      -0.1226
MARITALSTATUS=1000000                     -0.4234
MARITALSTATUS=10000000                     0.3427
MARITALSTATUS=100000000                   -0.2148
MARITALSTATUS=1000000000                  -0.2048
MARITALSTATUS=10000000000                 -1.2314
RATEPLAN=F3                               -0.1388
RATEPLAN=F6                                0.5438
RATEPLAN=F1                               -0.6597
RATEPLAN=F7                                2.8608
RATEPLAN=F4                                     0
RATEPLAN=F2                               -0.4971
RATEPLAN=F5                                0.1694
RATEPLAN=F8                              420.4701
PAYER_AGE                                 -0.0101
TENURE                                     0.0002
REFILL_FREQ                                0.0369
REFILL_AMOUNT1                             0.0001
MAX_REFILL_FREQ                           -0.0711
TOTALOUT_FREQ                              0.0014
OUTNET_FREQ2                               0.0018
SMSIN2                                    -0.0004
IMAX_TOTALIN_FREQ                         -0.0254
AVG_TOTALIN_FREQ                          -0.0042
AVG_SMSOUT                                 0.0018
SUM_SMSIN                                 -0.0004
MAX_INNET_FREQ_RATIO                      -0.0484
MAX_OUTNET_FREQ_RATIO                     -1.6017
INNET_TCHARGE_RAT1                         0.1784
MAX_NUM_NEIBOR                            -0.0051
SUM_DEGREE_CENTRALITY                   -211.2721
SUM_TOTAL_PR_NEIBOR                             0
Intercept                                  0.8279
```

Figures 5-11 and 5-12 display the odds of churn for different rate plans given by the post- and prepaid logistic classifiers trained on wr-logistic postpaid set and LVF prepaid set respectively. The odds of an event are defined as the ratio of the probability that an event occurs to the probability that it fails to occur. The odds lie between 0 and $+\infty$. Odds equal to 1 means that both outcomes are equally likely.

$$\text{odds} = \frac{P(y=+1)}{1-P(y=+1)} \qquad \textbf{(5.4-1)}$$

Note that WEKA outputs the odds of censoring. The odds of churn are one divide by the odds of censoring. The odds are sorted in descending order in figures 5-11 and 5-12. Note that the first three values in figure 5-11 for rate plans G1, G2 and G3 approach positive infinity due to the fact that the corresponding odds of censoring are approximately zeros. The first 19 rate plans farthest to the left are those having odds of churn above 1. It indicates that a customer uses one of those rate plans is more likely to be a churner than a non-churner. Take an example of rate plan G11 with the odds of churn approximately 2. Customers in this rate plan are two times more likely to churn than to stay. Similar risky rate plans in prepaid are rate plans F1, F2 and F3 according to figure 5-12. Customers of rate plan F4 are neutral with respect to churn and the remaining rate plans have churn likelihood below 50%.

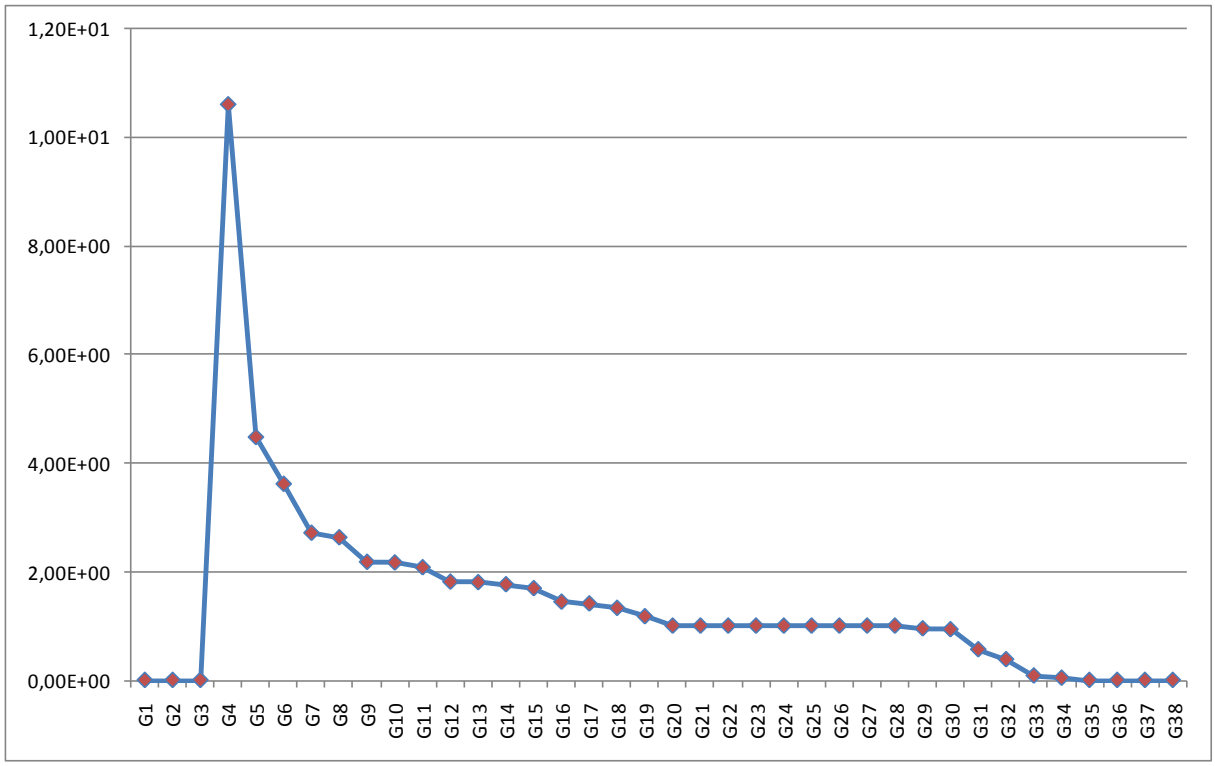

Figure 5-11. The odds of churn for postpaid rate plans given by logistic classifier trained on wr-logistic training set

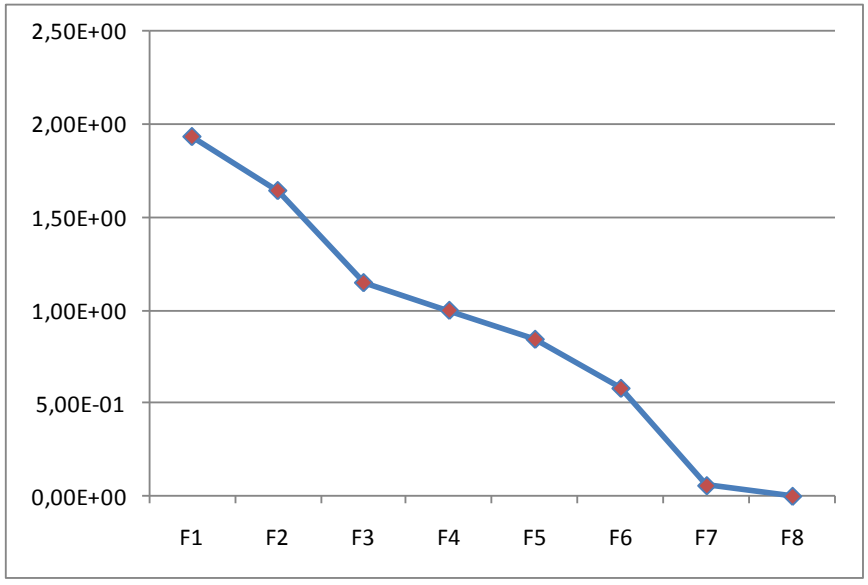

Figure 5-12. The odds of churn for prepaid rate plans given by logistic classifier trained on LVF training set

5.5 Models Comparison

In chapters from 5.1 to 5.4, classifiers trained on different reduced data sets are compared based on their performances in cross-validation. As the results, the best reduced data set and parameters are chosen for each learning algorithm. In this chapter, the best representative of each learning algorithm is tested on the prepared testing set in order to find out which one is the best of all. As showed in figure 5-13, the ROC curves of the postpaid classifiers are almost identically. In figure 5-14, the ROC curves of prepaid classifiers are also almost overlapped. The areas under ROC curves of classifiers in figure 5-14 are apparently larger than of those in figure 5-13 which indicate that prepaid classifiers perform better than postpaid classifiers.

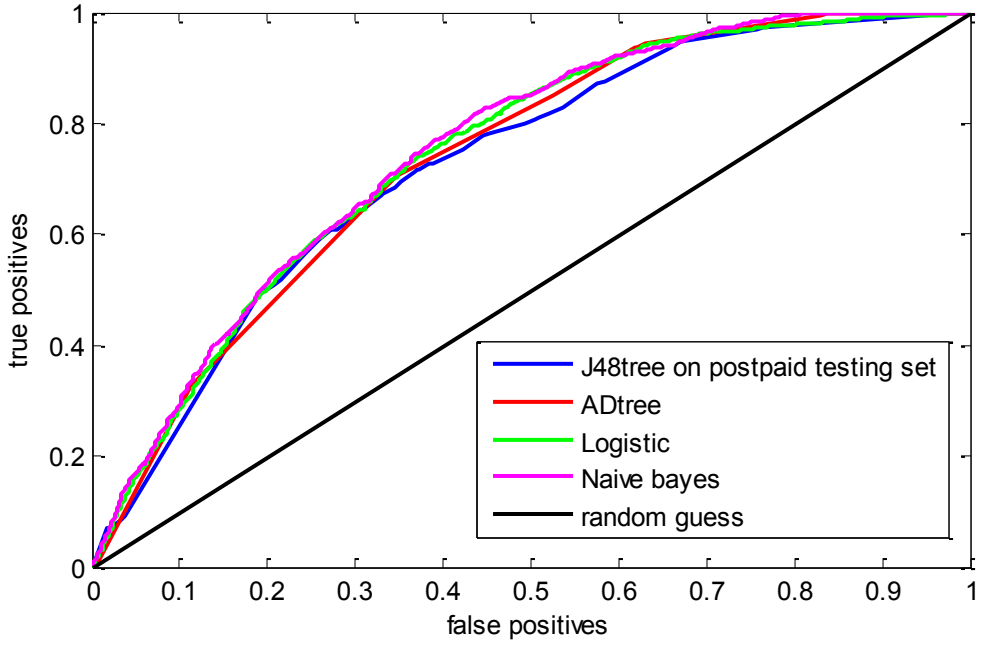

Figure 5-13. ROC curves of all classifiers on postpaid testing set

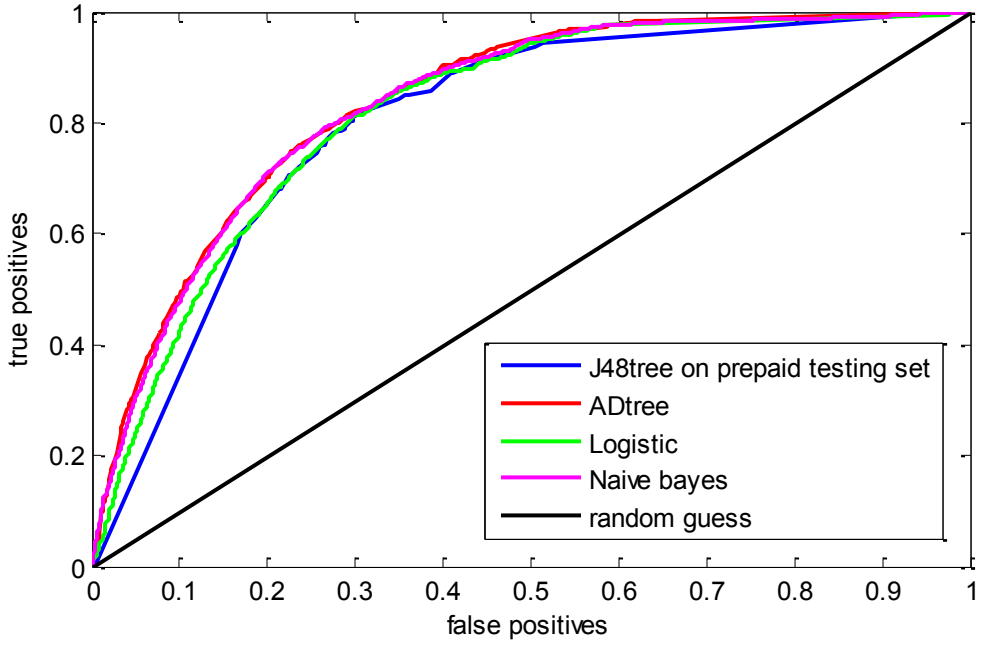

Figure 5-14. ROC curves of all classifiers on prepaid testing set

It is hard to pick the best classifier based on ROC curves in figures 5-13 and 5-14. As introduced in chapter 3.6, the confidence level measures how confident a classifier is with its prediction. Focusing on churners only, figures 5-15 and 5-16 display the results of postpaid classifiers while figures 5-17 and 5-18 of prepaid classifiers. The blue columns denote the proportion of correctly classified churners while the red columns denote the proportion of misclassified churners at each confidence level. Note that the total height of all blue columns is equal to the true positive (TP) rate. The desirable classifier is the one that captures large proportion of churners with high confidence level. Hence its performance is characterized by high blue columns which are located as far to the right hand side of the graph as possible.

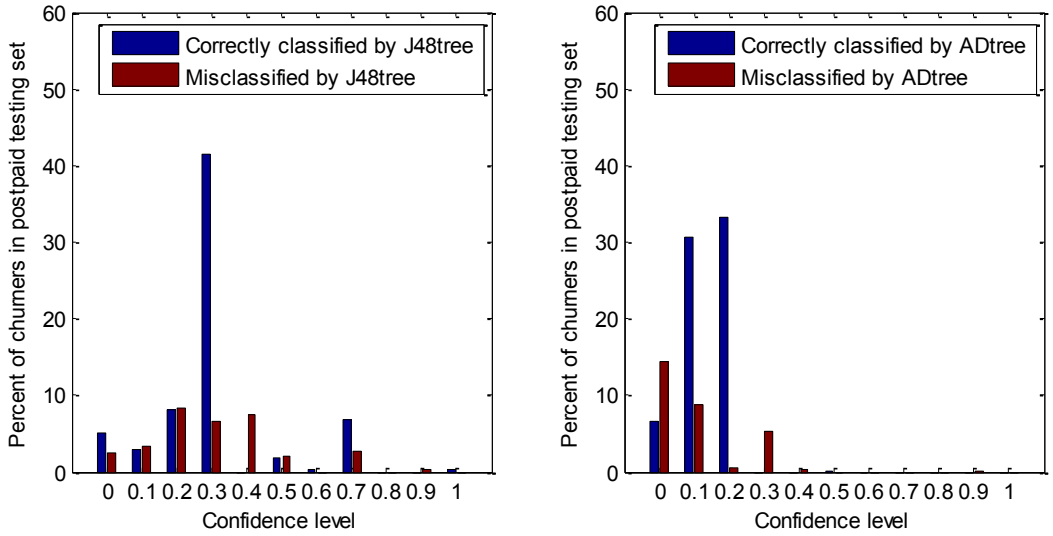

Figure 5-15. Prediction confidence level of J48tree and ADtree on postpaid testing set

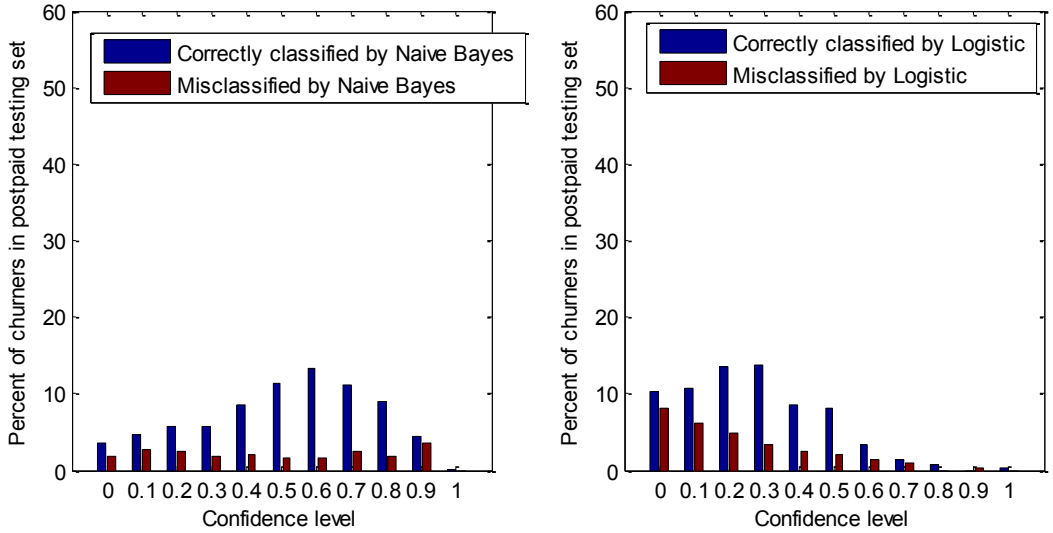

Figure 5-16. Prediction confidence level of Naïve Bayes and Logistic on postpaid testing set

In postpaid, between J48tree and ADtree, J48tree is more preferable. ADtree has in general low confidence on its predictions with the level which is equal or lower than 0,2. This confidence level corresponds to the probability of 60%. J48tree performs the best at a medium confidence level of 0,3 which is equivalent 65% certain. At this level, it captures about 42% of churners. J48tree shows also high confidence at a level of 0,7 on about 5% of its correct classification. As showed in figure 5-16, Naïve Bayes shows a desirable property of a

classifier which is high confidence level on major portion of its predictions. The peak is reached at a confidence level equals to 0,6 where about 13% of churners are captured. Confidence level is associated with the prediction quality. As can be seen in figure 5-16 the misclassification rate of the logistic classifier decreases as the confidence level rises.

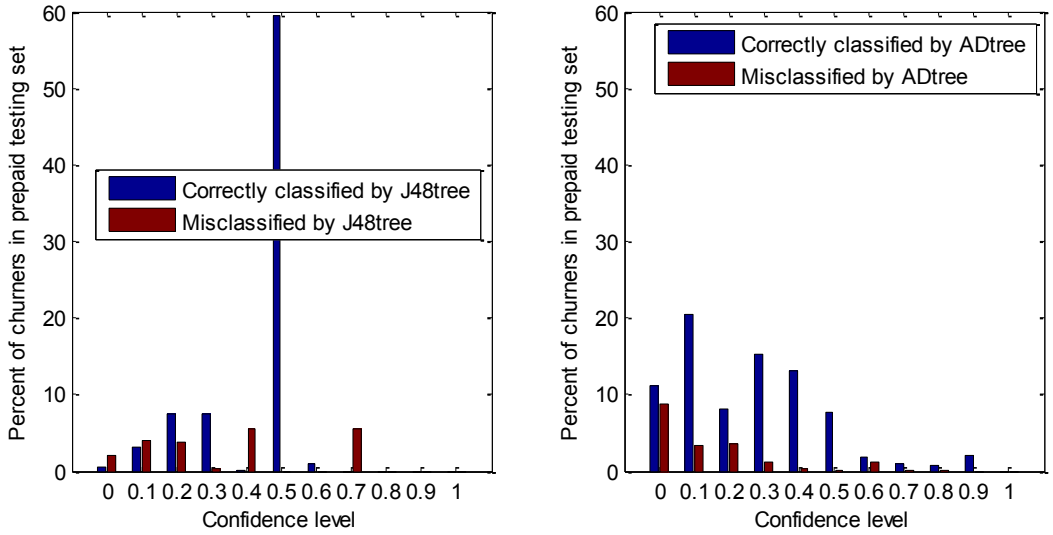

Figure 5-17. Prediction confidence level of J48tree and ADtree on prepaid testing set

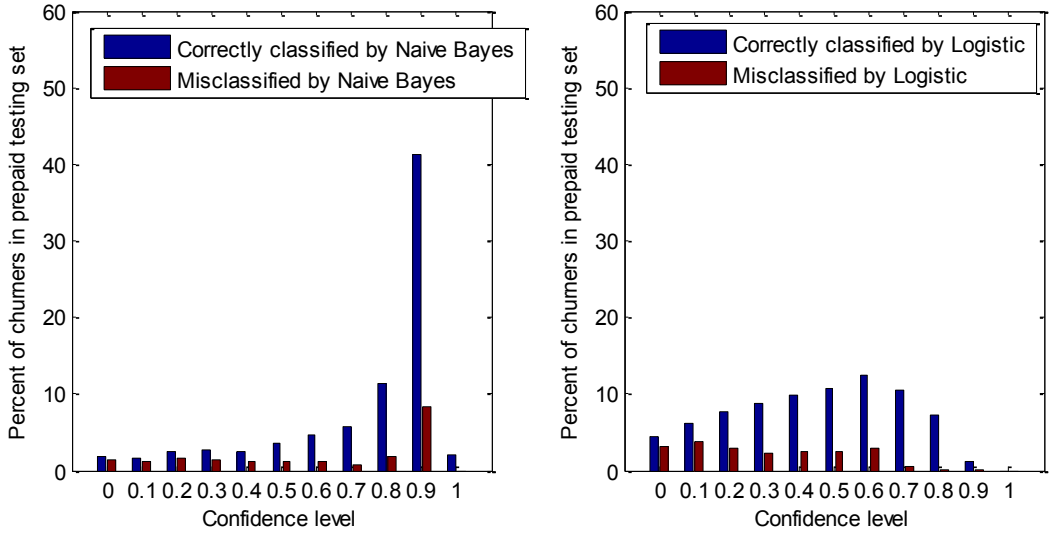

Figure 5-18. Prediction confidence level of Naïve Bayes and Logistic on prepaid testing set

According to figure 5-17, J48tree classifier performs even better in prepaid than in postpaid, mainly due to the fact that the prepaid training data is two times more than postpaid training data. The highest blue column indicates that it can identify nearly 60% of the churners with a confidence level of 0,5 or 75% certain. ADtree can identify similar number of churners but with lower level of certainty. The Naïve Bayes classifier makes more and more correct classifications as the confidence level rises. At its peak, it correctly predicts about 42% of the churners with the probability of 0,95% (the confidence level is 0,9). Logistic classifier underlines again that the misclassification rate decreases, more and more instances are correctly classified as the confidence level goes up.

Tables 5-7 and 5-8 sum up the performances of the eights best classifiers in postpaid and prepaid. In postpaid, J48tree has the highest OA of 67,2%, Naïve Bayes has the highest AUC of 74,4% and also the highest overall prediction confidence level (OCL) of 39,7%. In prepaid, logistic has the highest OA of 73,8%. ADtree has the highest AUC of 83,9%. Naïve Bayes again scores the highest on OCL with 59,4%. The performances are quite closed. It is interesting that being the most simple model, Naïve Bayes still works well although it needs only 10 features in both post and prepaid.

By observing the values of two metrics OA and AUC in both tables, it can be seen that all prepaid classifiers perform in overall better than all postpaid classifiers. A test is carried out to examine if this difference is formed due to the fact that the amount of training data in prepaid is two times more than in postpaid. In this test, the logistic classifier is trained on the prepaid LVF training set. The classifier is trained on 20% to 90% of the whole training set, and then tested on the full testing set. The effect of training data amount is not significant. The performance level of the classifier remains stable even when the training set has been reduced to only 20%.

Table 5-7. Overall performance comparison of classifiers for postpaid

algorithm	features set	parameter	training time	num features	OA	AUC	OCL
J48tree	CFS	min=40	0,44	28	67,2	72,7	20,5
Adtree	CFS	nb=7	1,18	28	65,4	73,5	9,8
Bayes	wr-Bayes		0,03	10	59,5	74,4	39,7
Logistic	wr-logistic		1,06	12	66,6	71,5	19,0

Table 5-8. Overall performance comparison of classifiers for prepaid

algorithm	features set	parameter	training time	num features	OA	AUC	OCL
J48tree	full set	min=64	12,26	300	71,6	80,5	34,4
Adtree	full set	nb=25	208,65	300	71,4	83,9	21,4
Bayes	wr-Bayes		0,03	10	71,4	82,0	59,4
Logistic	LVF		1,9	24	73,8	83,6	35,7

Table 5-9. Results from prior researches on churn prediction based on the overall accuracy.

Research	Training set	Testing set	DT	SVMs	ANN	Logistic
(Hung, Yen, & Wang, 2006) – postpaid sector	146000/14000	50000/355	86,3			
(Hadden, Tiwari, Roy, & Ruta, 2006)	101/101	490/210	82		72	81
(Archaux, Laanaya, Martin, & Khenchaf, 2004) – prepaid sector	3000/3000	3000/3000		69,8	69,6	
(Yi & Guo-en, 2010)	1433/1349	1588/396	57,4	61,3	58	59,3
(Guo-en & Wei-dong, 2008)	852/622	534/432	52,5	59,7	55,7	58,9
Results of this study for postpaid sector	2190/2190	27932/828	67,2			66,6
Results of this study for prepaid sector	4234/4234	36409/1355	71,6			73,8

Comparing to prior researches (see table 5-9) on churn prediction discussed in chapter 2.1, the performances of the proposed classifiers in this study are comparable. (Hung, Yen, & Wang, 2006) with a huge training set of 146000 non-churners and 14000 churners obtains a decision tree with 86,3% OA. (Hadden, Tiwari, Roy, & Ruta, 2006) employs decision tree, artificial neural network and logistic. They score up to 80% OA using a small training set contains 101 churners and 101 non-churners. (Archaux, Laanaya, Martin, & Khenchaf, 2004) has a similar large data set as in this study with 6000 cases. Number of churners and non-churners are equal. Training support vectors machine and artificial neural network on this data, they achieve about 70% OA. (Yi & Guo-en, 2010) and (Guo-en & Wei-dong, 2008) create decision tree, SVMs, ANN and logistic classifiers. Their OAs are about 60%.

5.6 Summary

This chapter gives and discusses detailed results of model building along with model comparison.

Both J48tree and ADtree are examined with different parameters and reduced data sets. J48tree classifiers undergo the effect of over-fitting as the confident factor is increased and the minimum number of instances per leaf is decreased. J48tree classifiers trained on the CFS set perform the best and on the full set perform the worst in postpaid. J48tree classifiers trained on the full set perform the best and on the lnfoGain set perform the worst in prepaid. The primary churn indicators according to J48tree are: customer age, his rate plan and marital status, in which land area he lives, number of text messengers he receives, his out-net calls amount. In addition, for a prepaid customer, the amount of phone calls he receives and his international calls expense also matter. Results from J48tree verifies the known fact that prepaid customers are more sensitive than postpaid customers regarding churn. By receiving more than 35 text messengers, a postpaid customer becomes more likely to churn while for a prepaid customer, the required number of messenger is only 16.

ADtrees aren't as sensitive to over-fitting when the tree grows bigger as J48trees thanks to its implementation in which all instances are considered at each node addition. ADtree classifiers trained on the CFS set perform the best in postpaid and on the full set perform the best in prepaid. Those trained on InfoGain sets perform worst overall in both sectors. ADtree classifiers point to similar churn indicators as J48tree classifiers do. The ADtree classifier for prepaid divides customers into three age groups of which churn likelihood ranges from low, medium to high. It is worth to notice that the two features AVG_OUTNET_FREQ and CUST_AGE are quite relevant since they are pronounced in both J48tree and ADtree for both post- and prepaid.

Naïve Bayes classifiers trained on the wr-Bayes post- and prepaid sets perform the best of all. They underline the churn indicators pointed out by J48tree and ADtree and imply that postpaid churners make almost half of their phone calls out-net and prepaid churners spend more than half of their expenses in out-net calls on the top month. In addition, they are well-connected to the social network by either having many neighbors, having well-connected neighbors or are well-connected themselves. Naïve Bayes classifiers also reveal that the churners in both post- and prepaid are under the population' average age.

Logistic classifier trained on wr-logistic sets performs the best overall, although the one trained on the LVF set achieves the highest AUC value. Comparing the odds of churn for

different rate plans, 19 rate plans in postpaid and 3 in prepaid having churn likelihood above 50% are revealed. Customers in these rate plans are more likely to churn than to stay.

The best classifiers representing each learning algorithm are compared based on their performance on the testing sets. They are almost undistinguishable regarding their ROC curves. When prediction confidence level is considered, both Naïve Bayes and Logistic are more preferable than J48tree and ADtree. They show desirable properties of a classifier. Naïve Bayes has high confidence level on a major portion of its prediction. Number of misclassified instances by logistic decreases as the confidence level rises. The performances of the classifiers are closed. In postpaid, J48tree has the highest OA of 67,2%, Naïve Bayes has the highest AUC of 74,4% and at the same time highest OCL of 39,7%. In prepaid, logistic has the highest OA of 73,8%. ADtree has the highest AUC of 83,9%. Naïve Bayes again scores the highest on OCL with 59,4%. The performances are comparable to the results achieved by prior researches on churn prediction.

6 Conclusions and Future Work

Customer churn is a heavy concern in almost every industry that offers customers products or services. During the last years, mobile operators experience increasing customer movements from one service provider to another. Loss of customers corresponds to loss of revenue, not to be mentioned the domino effect that it may cause when one customer churn influences others to churn as well. Regarding the Icelandic mobile market, Síminn and Vodafone remained the two largest mobile operators for many years. However, the whole picture is changing apparently since 2007. The entries of new mobile operators have increased the competition level. The market shares are gradually divided more evenly between operators, especially in the prepaid sector. In this market environment, customer retention becomes more important and it is worth to build a churn prediction model.

Utilizing the data warehouse of the mobile operator, a database is established from which features belong to six different categories can be extracted. They are the following: demographics, billing data, refill history, calling pattern, CDR billed and network features. Monthly aggregated features in the target window of three months are extracted for all active customers at a particular time point. For the purpose of training a classifier, an observation period of five months following the target window is used for labeling customers as churn or censoring depends on their status during the period. The proposed data acquisition method makes it convenient to update the classifiers on a regular basis so that they reflect potential changes in the market environment. The training period includes the target window and the observation period which are both adjustable.

The modeling part starts by selecting the most significant features. Two individual feature selection methods Relief and InfoGain, two filter approaches of feature subset selection methods and four wrappers are used. Among more than 300 original features, each feature selection method selects less than 30 most relevant features. Some features are chosen by more than one feature selection method which underlines their relevancy. Feature selection as a pre-processing step not only makes the later built classifiers less complex but also improves their performance. Classifiers using the reduced data perform overall better than those using the full data set.

The best classifiers are chosen for post- and prepaid separately by making comparisons between different combinations of feature selection methods and learning algorithms. By taking into account the interaction between features, wrappers and filters perform overall better than individual feature selectors such as InfoGain and Relief. Comparing between wrappers and filters, the differences in their performances and number of chosen features are not significant. However regarding execution time, while wrappers take many hours to run, filters take only a few minutes.

The findings of this study confirm that churn can be predicted successfully with certain level of accuracy and confidence using available resources and techniques. All postpaid classifiers achieve above 60% overall accuracy on a testing set of 30 thousand instances. All prepaid classifiers achieve above 70% overall accuracy on a testing set of 40 thousand instances. These performances are comparable to the results achieved by prior researches on churn prediction.

The following primary churn indicators are underlined repeatedly by all classifiers: customer age, his rate plan and marital status, in which land area he lives, amount of calls and text messengers he receives, his out-net calls amount and expense. In addition, churners appear to be those who are well-connected to the social network by either having many neighbors, having well-connected neighbors or are well-connected themselves.

Churn cannot be predicted with 100% accuracy due to many uncontrollable factors both environmental and personal unless they can somehow be included in the models such as strategies of the competitors, education and income of customers, etc. There is, however, still room for improvement based on the results of this study. Ideas about future work are the following.

- It is clear that features are those that matter the most. It is worth to generate more relevant features. Even irrelevant features, if any can still be eliminated by feature selection. Many researches use information such as handset features, customer service cases and calls to the service centre. The results of feature selection of this study can be utilized by exploiting deeper into the most significant feature categories and generating more features from these sources.
- All prepaid classifiers achieve about 10% more accuracy than postpaid classifiers. A test has been carried out on the logistic classifier to examine if this difference is formed due to the fact that the amount of training data in prepaid is two times more than in postpaid. The result shows that the performance of the classifier remains stable while the amount of training data is gradually reduced. The question needs to be answered why the prepaid classifiers perform better than postpaid classifiers.
- From the theoretical perspective, different experiments can be conducted such as using other search methods in feature selection than BestFirst or apply a wrapper to search a reduced feature space provided by a filter. Other more sophisticated learning algorithms such as artificial neural network, support vectors machine and Bayesian networks can be examined. Besides, customers can be segmented into groups according to specific criteria and different classifiers are created for each group.
- The remaining work that is crucial is to adopt the proposed framework of churn prediction. That is to integrate churn prediction tools into the current business process. It is desirable to build a comprehensive system for churn prediction from data extraction to model building and scoring which can be run automatically and carried out on a regular basis. For a short term, the procedure code can be optimized so that the data set can be generated from the data warehouse in as a short time as possible.
- Nowadays, churn prediction cooperated with customer life time value research forms a powerful tool. How churn prediction and customer life time value benefit from each other is an interesting research direction.

References

Alpaydin, E. (2010). *Introduction to Machine Learning.* London, England: The MIT Press.

Archaux, C., Laanaya, H., Martin, A., & Khenchaf, A. (2004). An SVM based Churn Detector in Prepaid Mobile Telephony. *IEEE* .

Berson, A., Smith, S., & Thearling, K. (2000). *Building Data Mining Applications for CRM.* New York, NY: McGraw-Hill.

Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th international conference on World Wide Web*, (pp. 107-117). Brisbane, Australia.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Elsevier* , 4626-4636.

Cetnik, B. (1990). Estimating Probabilities: A crucial task in machine learning. *Ninth European Conference on Artificial Intelligence*, (pp. 147-149). London.

CRISP-DM - Process Model. (n.d.). Retrieved April 29, 2011, from CRISP-DM - Home: www.crisp-dm.org

Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukheejea, S., & Nanavati, A. A. (2008). Social ties and their relevance to churn in mobile telecom networks. *EDBT '08 Proceedings of the 11th international conference on Extending database technology: Advances in database technology* (pp. 668-677). Nantes, France: ACM.

Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Inrelligent Data Analysis 1* , 131-156.

Easley, D., & Kleinberg, J. (2010). *Networks, Crowds and Markets: Reasoning about a Highly Connected World.* Cambridge University Press.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters* , 861-874.

Freund, y., & Mason, L. (1999). The alternating decision tree learning algorithm. *Proceedings of the Sixteenth International Conference on Machine Learning*, (pp. 124-133). Bled, Slovenia.

George, H. J., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. *Proceeding of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338-345). San Mateo: Morgan Kaufmann .

Gislason, H. V. (2005). *Samkeppni á grunnneti Landssímans.* Reykjavik: Icelandic Post- and Telecom Administration.

Guo-en, X., & Wei-dong, J. (2008). Model of Customer Churn Prediction on Support Vector Machine. *ScienceDirect* , 71-77.

Gutkin, M. (2008). *Feature selection methods for classification of gene expression profiles.* Tel-Aviv University.

Hadden, J., Tiwari, A., Roy, R., & Ruta, D. (2006). Churn Prediction: Does Technology Matter? *International Journal of Intelligent Systems and Technologies , 1*.

Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning.* Hamilton, NewZealand: The University of Waikato.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemmann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations , 11* (1).

He, Z., Xu, X., Huang, J. Z., & Deng, S. (2004). Mining class outliers: Concepts, algorithms and applications in CRM. *Expert Systems with Applications , 27*, 681-697.

Hung, S. Y., Yen, D. C., & Wang, H. Y. (2006). Applying data mining to telecom churn management. *Expert Systems with Applications , 31*, 515-524.

Hunt, E. B., Marin, J., & Stone, P. J. (1966). Experiments in induction. *Academic Press* .

Icelandic Post- and Telecom Administration. (2009). *Statistics on the Icelandic electronic communications market 2008.* Reykjavik: Icelandic Post- and Telecom Administration.

Icelandic Post- and Telecom Administration. (2011). *Statistics on the Icelandic electronic communications market 2010.* Reykjavik: Icelandic Post- and Telecom Administration.

Icelandic Post- and Telecom Admistration. (2007). *Statistics on the Icelandic electronic communications market 2006.* Reykjavik: Icelandic Post- and Telecom Admistration.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *Proceedings of Eleventh International Conference on Machine Learning* (pp. 121-129). San Francisco, CA: Morgan Kaufmann.

Kentrias, S. (2001). *Customer relationship management: The SAS Perspective*. Retrieved March 24, 2011, from CRM Today: www.crm2day.com

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Tradition methods and a new algorithm. *Tenth National Conference on Artificial Intelligence* (pp. 129-134). MIT Press.

Kohavi, H., & John, G. H. (1997). Wrapper for Feature Subset Selection. *AIJ special issue on relevance* .

Liu, H., & Setiono, R. (1996). A Probabilistic Approach to Feature Selection- A Filter Solution. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 319-327). Morgan Kaufmann.

Liu, H., Motoda, H., & Yu, L. (2004). A selective sampling approach to active feature selection. *Artificial Intelligence* , 49-74.

MATLAB. (2009). *Version R2009a* . Natick, Massachusetts:: The MathWorks Inc.

Mattersion, R. (2001). *Telecom churn management.* Fuquay- Varina, NC: APDG Publishing.

Mitchell, T. M. (2005). Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression. In *Machine Learning.* Unpublished manuscript.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Science/Engineering/Math.

Ng, A. Y., & Jordan, M. (2002). On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes. *Neural Information Processing Systems: NIPS 14.*

Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature reveiw and classification. *Expert Systems with Applications , 36*, 2592-2602.

Nordic National Regulatory Authorities. (2005). *Competition and regulation in the Nordic mobile markets.* Nordic National Regulatory Authorities.

Pfahringer, B., Holmes, G., & Kirkby, R. (2001). Optimizing the induction of alternating decision trees. *Proceedings of the Fith Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining* (pp. 477-487). Berlin: Springer.

Press, W. H., Flannery, B. P., Teukolski, S. A., & Vetterling, W. T. (1988). *Numerical Recipes in C.* Cambridge University Press.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Francisco, CA: Morgan Kaufmann Publishers Inc.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning , 1*, 81-106.

Richter, Y., Yom-Tov, E., & Slonim, N. (2010). Predicting customer churn in mobile networks through analysis of social groups. *SDM* (pp. 732-741). SIAM.

SAP. (2011). *Churn Analysis*. Retrieved 2011, from SAP Library - Customer Relationship Management: http://help.sap.com/saphelp_nw04//helpdata/en/a1/e816b98e0f3a4c8681154ffdd91f42/content.htm

Shearer, C. (2000). The CRISP_DM model: The New Blueprint for Data Mining. *Journal of Data Warehousing* , 13-22.

Siminn. (2011). *Siminn*. Retrieved July 1, 2011, from History - Siminn - About us: http://www.siminn.co.uk/about-us/siminn/history/

Thearling, K. (1999). An introduction of data mining. *Direct Marketing Magazine* .

Tong, W., Xie, Q., Hong, H., Shi, L., Fang, H., & Perkins, R. (2004). Assessment of prediction confidence and domain extrapolation of two structure-activity relationship models for predicting estrogen receptor binding activity. *Environ Health Perspect* , 1249-1254.

Wei, C. P., & Chiu, I. T. (2002). Turning telecommunications call detail to churn prediction: A data mining approach. *Expert Systems with Applications , 23*, 103-112.

Witten, I. H., & Frank, E. (2005). *Data Mining, Practical Machine Learning Tools and Techniques.* San Francisco, CA: Morgan Kaufmann Publishers.

Yan, L., Wolniewicz, R. H., & Dodier, R. (2004). Predicting Customer Behavior in Telecommunications. *IEEE Intelligent Systems* , 1094-7167.

Yi, L., & Guo-en, X. (2010). The Explanation of Support Vector Machine in Customer Churn Prediction. *IEEE* .

Appendix A - Description of the Data Acquisition Process

Instruction - Steps to build a data set of customer signatures

A- BUILDING CUST_DEMOGRAPHIC TABLE
A01- create new table phonenr_list, note that 2 date values need to be modified
A02- create new table cust_history
A03- create INDEX for table cust_history
A04- create new table customers
A05- compile and execute build_customers
A06- delete rows from customers so that it contains only one snapshot per one customer, remember the date value
A07- delete duplicated rows from customers
A08- create or replace view vw_porting
A09- create new table churn_list, remember to specify the churn observation period
A10- create new table cust_demographic
A11- compile and execute build_cust_demographic
A12- delete duplicated rows from cust_demographic
A13- create primary key for table cust_demographic

B- BUILDING BILLING_DATA TABLE
B01- create new table dim_service
B02- create new table fact_revenue, check if it contains the necessary billing_date
B03- create primary key for table fact_revenue
B04- create or replace view vw_billing_data_agr
B05- create new table billing_data
B06- compile and execute build_billing_data

C- BUILDING REFILL_HISTORY TABLE
C01- create or replace view vw_refill_history
C02- create new table refill_history
C03- compile and execute build_refill_history

D- BUILDING CALLING_PATTERN TABLE
D01- create and insert into table cdr_calling_pattern, check if it contains the necessary partition
D02- create INDEX for table cdr_calling_pattern
D03- create or replace view vw_billing_data_agr
D04- create new table calling_pattern
D05- compile and execute build_calling_pattern

E- BUILDING CDR_BILLED TABLE
E01- create or replace view vw_cdr_billed_postpaid
E02- create new table cdr_billed_prepaid
E03- create INDEX for table cdr_billed_prepaid

E04- create new table cdr_billed
E05- compile and execute build_cdr_billed_postpaid
E06- compile and execute build_cdr_billed_prepaid

F- BUILDING SNA_NETWORK TABLE
F01- create new table network_data
F02- import data into table network_data from CDR data
F03- create new table pagerank
F04- import data into table pagerank by running centrality program on the CDR data in F02
F05- create INDEX for table pagerank
F06- create new table decoder
F07- create new table porting_neibor
F08- create new table network_data_with_churn
F09- create new table sna_network
F10- compile and execute build_sna_network

G- BUILDING CUST_SIGNATURE TABLE
G01- create new table cust_signature
G02- drop columns from table cust_signature
G03- create new table prepaid and extract data
G04- create new table postpaid and extract data

H- EXTRACT DATA
H01- export query results from Oracle to *.csv file
H02- open in Textpad and convert , to . and ; to ,
H03- save as *.arff file in Weka
H04- open the *.arff file in Textpad and modify the header if necessary

Appendix B - Results of Feature Selection

I. Postpaid feature selection

feature	CFS	Relief	wr-Bayes	LVF	InfoGain	wr-logistic	wr-J48tree	wr-Adtree	Total
LAND_AREA	1	1	1	1		1	1	1	7
CUST_AGE	1	1	1			1	1	1	6
RATEPLAN	1	1		1	1	1			5
SUM_SMSIN	1				1		1	1	4
PAYER_AGE	1	1		1					3
IMAX_OUTNET_TCHARGE_RAT		1	1						2
MARITALSTATUS		1		1					2
SUM_TOTAL_PR_NEIBOR					1	1			2
IMAX_S_OUTNET_CHARGE		1				1			2
MAX_OUTNET_FREQ					1	1			2
NUM_NEIBOR2				1		1			2
OUTNET_FREQ				1	1				2
GENDER		1		1					2
AVG_TCHARGE	1				1				2
IMAX_SMSIN		1		1					2
SUM_TOTAL_DC_NEIBOR			1		1				2
INNET_TCHARGE_RAT2				1			1		2
AVG_OUTNET_FREQ	1				1				2
TENURE		1	1						2
MAX_OUTNET_CHARGE	1				1				2
IMAX_FROM_NEIBOR_VOL		1							1
SMS_INNET_CHARGE2							1		1
OUTNET_FREQ_RATIO	1								1
IMAX_INNET_VOL_RATIO		1							1
SMSIN1							1		1
IMAX_NUM_NEIBOR	1								1
NUM_PRODUCT2		1							1
IMAX_OUTNET_FREQ_RATIO		1							1
ABROAD_VOL1								1	1
AVG_OUTNET_VOL					1				1
SMS_OUTNET_TCHARGE_RAT	1								1
IMAX_OUTNET_VOL_RATIO		1							1
SUM_OUTNET_CHARGE					1				1
IMAX_S_INNET_CHARGE		1							1
TOTALIN_VOL1				1					1
IMAX_S_INNET_TCHARGE_RAT				1					1

OUTNET_CHARGE				1				1
AVG_S_INNET_CHARGE							1	1
OUTNET_FREQ2	1							1
IMAX_SMS_OUTIN_RATIO		1						1
RATIO_DISCOUNT1			1					1
AVG_SMSIN				1				1
SMS_OUTNET_CHARGE			1					1
IMAX_SMSOUT		1						1
SMS_OUTNET_TCHARGE_RAT2	1							1
IMAX_TOTAL_DC_NEIBOR			1					1
SMSOUT							1	1
IMAX_TOTALIN_VOL		1						1
SUM_OUTNET_VOL				1				1
IMAX_VOICE_OUTIN_VOL_RATIO			1					1
IMAX_INNET_TCHARGE_RAT							1	1
AMOUNT_DISCOUNT1	1							1
VOICE_OUTIN_VOL_RATIO1							1	1
INNET_VOL	1							1
NUM_SERVICE			1					1
INNET_VOL2							1	1
FAMILYSIZE			1					1
AVG_TOTAL_DC_NEIBOR				1				1
OUTNET_FREQ_RATIO1					1			1
AVG_TOTAL_PR_NEIBOR				1				1
OUTNET_VOL	1							1
MAX_CHURN_NEIBOR_RAT		1						1
IMAX_ABROAD_CHARGE					1			1
MAX_INNET_FREQ_RATIO	1							1
RATIO_GSM			1					1
MAX_INNET_TCHARGE_RAT	1							1
SMS_OUTIN_RATIO							1	1
MAX_NUM_NEIBOR			1					1
SMS_OUTNET_CHARGE2							1	1
ABROAD_VOL_RATIO1	1							1
SMS_OUTNET_TCHARGE_RAT1	1							1
CHURN_NEIBOR_RAT	1							1
SMSIN			1					1
MAX_OUTNET_FREQ_RATIO			1					1
SMSIN2	1							1
MAX_OUTNET_TCHARGE_RAT					1			1
SUM_NUM_NEIBOR				1				1
MAX_OUTNET_VOL				1				1
SUM_OUTNET_FREQ				1				1
MAX_OUTNET_VOL_RATIO	1							1
IMAX_DEGREE_CENTRALITY	1							1
MAX_S_INNET_CHARGE							1	1

IMAX_INNET_FREQ_RATIO				1					1
MAX_S_OUTNET_TCHARGE_RAT	1								1
TO_CHURN_NEIBOR_VOL_RAT								1	1
MAX_SMS_OUTIN_RATIO						1			1
VOICE_OUTIN_VOL_RATIO	1								1
MYSUM1	1								1
IMAX_INNET_VOL				1					1
AVG_OUTNET_CHARGE					1				1
Total	**27**	**20**	**9**	**19**	**20**	**11**	**14**	**5**	**125**

II. Prepaid feature selection

feature	CFS	Relief	wr-Bayes	wr-logistic	wr-J48tree	wr-ADtree	LVF	InfoGain	Total
CUST_AGE	1	1	1	1	1	1	1		7
AVG_TOTALIN_FREQ					1	1	1	1	4
MARITALSTATUS	1	1	1				1		4
RATEPLAN	1	1			1		1		4
PAYER_AGE			1	1			1		3
LAND_AREA		1	1				1		3
TENURE		1		1			1		3
GENDER		1	1				1		3
MAX_OUTNET_TCHARGE_RAT	1		1	1					3
SUM_TOTALIN_FREQ					1			1	2
REFILL_FREQ1	1					1			2
PAGERANK_RAT2				1	1				2
AVG_TOTAL_DC_NEIBOR				1				1	2
SUM_SMSIN							1	1	2
IMAX_OUTNET_VOL_RATIO		1		1					2
OUTNET_FREQ_RATIO	1			1					2
IMAX_TOTALIN_FREQ		1					1		2
REFILL_FREQ	1						1		2
INNET_TCHARGE_RAT1		1					1		2
SUM_DEGREE_CENTRALITY							1	1	2
MAX_NUM_CHURN_NEIBOR				1		1			2
SUM_TOTAL_DC_NEIBOR			1					1	2
SUM_TOTAL_PR_NEIBOR							1	1	2
MAX_TOTALIN_FREQ								1	1
SMS_OUTIN_RATIO	1								1
PAGERANK			1						1
AVG_TCHARGE						1			1
SUM_OUTNET_FREQ	1								1
IMAX_SMSIN		1							1
FROM_NEIBOR_VOL2						1			1
IMAX_TCHARGE		1							1
REFILL_AMOUNT1							1		1

IMAX_TO_NEIBOR_VOL			1					1
SMSIN2						1		1
IMAX_TOTAL_DC_NEIBOR	1							1
IMAX_OUTNET_TCHARGE_RAT		1						1
IMAX_TOTAL_PR_NEIBOR	1							1
NUM_NEIBOR1				1				1
AVG_TOTAL_PR_CHURN_NEIBOR	1							1
OUTNET_TCHARGE_RAT1	1							1
IMAX_TOTALIN_VOL		1						1
AVG_OUTNET_FREQ							1	1
IMAX_TOTALOUT_FREQ		1						1
IMAX_NUM_NEIBOR	1							1
IMAX_TOTALOUT_VOL		1						1
SMS_OUTIN_RATIO2	1							1
IMAX_VOICE_OUTIN_VOL_RATIO		1						1
IMAX_OUTNET_CHARGE		1						1
AVG_TOTAL_PR_NEIBOR							1	1
SUM_REFILL_AMOUNT				1				1
INNET_VOL_RATIO				1				1
MAX_TOTAL_DC_NEIBOR							1	1
ISPAYER		1						1
MAX_VOICE_OUTIN_VOL_RATIO	1							1
AVG_NUM_CHURN_NEIBOR	1							1
OUTNET_CHARGE				1				1
AVG_NUM_NEIBOR							1	1
OUTNET_FREQ2						1		1
MAX_DEGREE_CENTRALITY							1	1
OUTNET_VOL_RATIO				1				1
MAX_INNET_FREQ_RATIO						1		1
AVG_DEGREE_CENTRALITY							1	1
MAX_INNET_TCHARGE_RAT		1						1
AVG_PAGERANK							1	1
ABROAD_TCHARGE_RAT1					1			1
IMAX_INNET_VOL_RATIO		1						1
MAX_NUM_NEIBOR						1		1
REFILL_FREQ2					1			1
ABROAD_VOL_RATIO					1			1
SMS_OUTIN_RATIO1	1							1
DEGREE_RAT2					1			1
SMSIN1				1				1
AVG_SMSIN							1	1
SMSOUT				1				1
TO_CHURN_NEIBOR_VOL2					1			1
SUM_NUM_NEIBOR							1	1
TOTALIN_FREQ	1							1
SUM_PAGERANK							1	1

TOTALIN_VOL2	1								1
IMAX_OUTNET_FREQ_RATIO					1				1
AVG_SMSOUT							1		1
MAX_SMSOUT	1								1
MAX_OUTNET_FREQ_RATIO							1		1
SUM_TOTALOUT_FREQ						1			1
VOICE_OUTIN_VOL_RATIO1	1								1
TO_CHURN_NEIBOR_VOL					1				1
ABROAD_CHARGE1					1				1
TOTAL_PR_CHURN_NEIBOR				1					1
MAX_PAGERANK								1	1
TOTALIN_VOL					1				1
MAX_REFILL_FREQ							1		1
TOTALOUT_FREQ							1		1
MAX_SMS_OUTIN_RATIO	1								1
MAX_SMSIN								1	1
MAX_OUTNET_VOL				1					1
Total	**23**	**20**	**9**	**18**	**16**	**5**	**23**	**20**	**134**

Appendix C - Descriptions of Features

Name	Type	Description	Category
phonenr	nominal	--customer's phone number	demographics
custid	nominal	--customer's social security id	demographics
cust_age	numeric	--customer's age	demographics
familysize	numeric	--family size	demographics
gender	nominal	--gender	demographics
land_area	nominal	--land area	demographics
maritalstatus	nominal	--marital status	demographics
rateplan	nominal	--rate plan	demographics
stype	nominal	--subscription type postpaid or prepaid	demographics
payer_age	numeric	--payer's age	demographics
Ispayer	nominal	--customer is the payer for his own service account or not	demographics
sl_fromdate	numeric	--effective date of the service account	demographics
sl_todate	numeric	--expiry date of the service account	demographics
status	nominal (class variable)	--status can be churn or censoring	demographics
tenure	numeric	--tenure(how long customer has been in this status)	demographics
num_return	numeric	--how many times has the customer churned and turned back	demographics
cust_skey	nominal	--customer's surrogate key	demographics
num_service	numeric	--number of billed services one month before the month of the data extraction	billing data
num_service1	numeric	--number of billed services two months before the month of the data extraction	billing data
num_service2	numeric	--number of billed services three months before the month of the data extraction	billing data
num_product	numeric	--number of billed products one month before the month of the data extraction	billing data
num_product1	numeric	--number of billed products two months before the month of the data extraction	billing data
num_product2	numeric	--number of billed products three months before the month of the data extraction	billing data
amount_gsm	numeric	--billed amount due to gsm usage one month before the month of the data extraction	billing data
amount_gsm1	numeric	--billed amount due to gsm usage two months before the month of the data extraction	billing data
amount_gsm2	numeric	--billed amount due to gsm usage three months before the month of the data extraction	billing data
amount_discount	numeric	--discount amount one month before the month of the data extraction	billing data
amount_discount1	numeric	--discount amount two months before the month of the data extraction	billing data
amount_discount2	numeric	--discount amount three months before the month of the data extraction	billing data

ratio_gsm	numeric	--ratio of gsm usage to total billed amount one month before the month of the data extraction	billing data
ratio_gsm1	numeric	--ratio of gsm usage to total billed amount two months before the month of the data extraction	billing data
ratio_gsm2	numeric	--ratio of gsm usage to total billed amount three months before the month of the data extraction	billing data
ratio_discount	numeric	--ratio of discount to total billed amount one month before the month of the data extraction	billing data
ratio_discount1	numeric	--ratio of discount to total billed amount two months before the month of the data extraction	billing data
ratio_discount2	numeric	--ratio of discount to total billed amount three months before the month of the data extraction	billing data
mysum	numeric	--total billed amount one month before the month of the data extraction	billing data
mysum1	numeric	--total billed amount two months before the month of the data extraction	billing data
mysum2	numeric	--total billed amount three months before the month of the data extraction	billing data
refill_freq	numeric	--refill frequency one month before the month of the data extraction	refill history
refill_freq1	numeric	--refill frequency two months before the month of the data extraction	refill history
refill_freq2	numeric	--refill frequency three months before the month of the data extraction	refill history
refill_amount	numeric	--refill amount one month before the month of the data extraction	refill history
refill_amount1	numeric	--refill amount two months before the month of the data extraction	refill history
refill_amount2	numeric	--refill amount three months before the month of the data extraction	refill history
max_refill_freq	numeric	--the maximum refill frequency in a month	refill history
max_refill_amount	numeric	--the maximum refill amount in a month	refill history
imax_refill_freq	numeric	--the month when the maximum refill frequency occurs	refill history
imax_refill_amount	numeric	--the month when the maximum refill amount occurs	refill history
sum_refill_freq	numeric	--total refill frequency over three months precede the month the data extraction	refill history
sum_refill_amount	numeric	--total refill amount over three months precede the month the data extraction	refill history
avg_refill_freq	numeric	--average monthly refill frequency	refill history
avg_refill_amount	numeric	--average monthly refill amount	refill history
innet_vol	numeric	--inside network call volume one month before the month of the data extraction	calling pattern
innet_freq	numeric	--inside network call frequency one month before the month of the data extraction	calling pattern
outnet_vol	numeric	--outside network call volume one month before the month of the data extraction	calling pattern
outnet_freq	numeric	--outside network call frequency one month before the month of the data extraction	calling pattern
abroad_vol	numeric	--abroad call volume one month before the month of the data extraction	calling pattern
abroad_freq	numeric	--abroad call frequency one month before the month of the data extraction	calling pattern

totalout_vol	numeric	--total orginating call volume one month before the month of the data extraction	calling pattern
totalout_freq	numeric	--total orginating call frequency one month before the month of the data extraction	calling pattern
totalin_vol	numeric	--total terminating call volume one month before the month of the data extraction	calling pattern
totalin_freq	numeric	--total terminating call frequency one month before the month of the data extraction	calling pattern
smsout	numeric	--total sending sms frequency one month before the month of the data extraction	calling pattern
smsin	numeric	--total receiving sms frequency one month before the month of the data extraction	calling pattern
innet_vol_ratio	numeric	--ratio of inside network to total orginating call volume one month before the month of the data extraction	calling pattern
innet_freq_ratio	numeric	--ratio of inside network to total orginating call frequency one month before the month of the data extraction	calling pattern
outnet_vol_ratio	numeric	--ratio of outside network to total orginating call volume one month before the month of the data extraction	calling pattern
outnet_freq_ratio	numeric	--ratio of outside network to total orginating call frequency one month before the month of the data extraction	calling pattern
abroad_vol_ratio	numeric	--ratio of abroad to total orginating call volume one month before the month of the data extraction	calling pattern
abroad_freq_ratio	numeric	--ratio of abroad to total orginating call frequency one month before the month of the data extraction	calling pattern
innet_vol1	numeric	--inside network call volume two months before the month of the data extraction	calling pattern
innet_freq1	numeric	--inside network call frequency two months before the month of the data extraction	calling pattern
outnet_vol1	numeric	--outside network call volume two months before the month of the data extraction	calling pattern
outnet_freq1	numeric	--outside network call frequency two months before the month of the data extraction	calling pattern
abroad_vol1	numeric	--abroad call volume two months before the month of the data extraction	calling pattern
abroad_freq1	numeric	--abroad call frequency two months before the month of the data extraction	calling pattern
totalout_vol1	numeric	--total orginating call volume two months before the month of the data extraction	calling pattern
totalout_freq1	numeric	--total orginating call frequency two months before the month of the data extraction	calling pattern
totalin_vol1	numeric	--total terminating call volume two months before the month of the data extraction	calling pattern
totalin_freq1	numeric	--total terminating call frequency two months before the month of the data extraction	calling pattern
smsout1	numeric	--total sending sms frequency two months before the month of the data extraction	calling pattern
smsin1	numeric	--total receiving sms frequency two months before the month of the data extraction	calling pattern

innet_vol_ratio1	numeric	--ratio of inside network to total orginating call volume two months before the month of the data extraction	calling pattern
innet_freq_ratio1	numeric	--ratio of inside network to total orginating call frequency two months before the month of the data extraction	calling pattern
outnet_vol_ratio1	numeric	--ratio of outside network to total orginating call volume two months before the month of the data extraction	calling pattern
outnet_freq_ratio1	numeric	--ratio of outside network to total orginating call frequency two months before the month of the data extraction	calling pattern
abroad_vol_ratio1	numeric	--ratio of abroad to total orginating call volume two months before the month of the data extraction	calling pattern
abroad_freq_ratio1	numeric	--ratio of abroad to total orginating call frequency two months before the month of the data extraction	calling pattern
innet_vol2	numeric	--inside network call volume three months before the month of the data extraction	calling pattern
innet_freq2	numeric	--inside network call frequency three months before the month of the data extraction	calling pattern
outnet_vol2	numeric	--outside network call volume three months before the month of the data extraction	calling pattern
outnet_freq2	numeric	--outside network call frequency three months before the month of the data extraction	calling pattern
abroad_vol2	numeric	--abroad call volume three months before the month of the data extraction	calling pattern
abroad_freq2	numeric	--abroad call frequency three months before the month of the data extraction	calling pattern
totalout_vol2	numeric	--total orginating call volume three months before the month of the data extraction	calling pattern
totalout_freq2	numeric	--total orginating call frequency three months before the month of the data extraction	calling pattern
totalin_vol2	numeric	--total terminating call volume three months before the month of the data extraction	calling pattern
totalin_freq2	numeric	--total terminating call frequency three months before the month of the data extraction	calling pattern
smsout2	numeric	--total sending sms frequency three months before the month of the data extraction	calling pattern
smsin2	numeric	--total receiving sms frequency three months before the month of the data extraction	calling pattern
innet_vol_ratio2	numeric	--ratio of inside network to total orginating call volume three months before the month of the data extraction	calling pattern
innet_freq_ratio2	numeric	--ratio of inside network to total orginating call frequency three months before the month of the data extraction	calling pattern
outnet_vol_ratio2	numeric	--ratio of outside network to total orginating call volume three months before the month of the data extraction	calling pattern
outnet_freq_ratio2	numeric	--ratio of outside network to total orginating call frequency three months before the month of the data extraction	calling pattern

abroad_vol_ratio2	numeric	--ratio of abroad to total orginating call volume three months before the month of the data extraction	calling pattern
abroad_freq_ratio2	numeric	--ratio of abroad to total orginating call frequency three months before the month of the data extraction	calling pattern
voice_outin_vol_ratio	numeric	--ratio of orginating to terminating call volume one month before the month of the data extraction	calling pattern
voice_outin_vol_ratio1	numeric	--ratio of orginating to terminating call volume two months before the month of the data extraction	calling pattern
voice_outin_vol_ratio2	numeric	--ratio of orginating to terminating call volume three months before the month of the data extraction	calling pattern
sms_outin_ratio	numeric	--ratio of sending to receiving sms frequency one month before the month of the data extraction	calling pattern
sms_outin_ratio1	numeric	--ratio of sending to receiving sms frequency two months before the month of the data extraction	calling pattern
sms_outin_ratio2	numeric	--ratio of sending to receiving sms frequency three months before the month of the data extraction	calling pattern
max_voice_outin_vol_ratio	numeric	--the maximum orginating to terminating call volume ratio in a month	calling pattern
imax_voice_outin_vol_ratio	numeric	--the month when the maximum orginating to terminating call volume ratio occurs	calling pattern
max_sms_outin_ratio	numeric	--the maximum sending to receiving sms frequency ratio in a month	calling pattern
imax_sms_outin_ratio	numeric	--the month when the maximum sending to receiving sms frequency ratio occurs	calling pattern
max_innet_vol	numeric	--the maximum inside network call volume in a month	calling pattern
imax_innet_vol	numeric	--the month when the maximum inside network call volume occurs	calling pattern
sum_innet_vol	numeric	--total inside network call volume over three months precede the month the data extraction	calling pattern
avg_innet_vol	numeric	--average monthly inside network call volume	calling pattern
max_innet_freq	numeric	--the maximum inside network call frequency in a month	calling pattern
imax_innet_freq	numeric	--the month when the maximum inside network call frequency occurs	calling pattern
sum_innet_freq	numeric	--total inside network call frequency over three months precede the month the data extraction	calling pattern
avg_innet_freq	numeric	--average monthly inside network call frequency	calling pattern
max_outnet_vol	numeric	--the maximum outside network call volume in a month	calling pattern
imax_outnet_vol	numeric	--the month when the maximum outside network call volume occurs	calling pattern
sum_outnet_vol	numeric	--total outside network call volume over three months precede the month the data extraction	calling pattern
avg_outnet_vol	numeric	--average monthly outside network call volume	calling pattern
max_outnet_freq	numeric	--the maximum outside network call frequency in a month	calling pattern
imax_outnet_freq	numeric	--the month when the maximum outside network call frequency occurs	calling pattern

sum_outnet_freq	numeric	--total outside network call frequency over three months precede the month the data extraction	calling pattern
avg_outnet_freq	numeric	--average monthly outside network call frequency	calling pattern
max_abroad_vol	numeric	--the maximum abroad call volume in a month	calling pattern
imax_abroad_vol	numeric	--the month when the maximum abroad call volume occurs	calling pattern
sum_abroad_vol	numeric	--total abroad call volume over three months precede the month the data extraction	calling pattern
avg_abroad_vol	numeric	--average monthly abroad call volume	calling pattern
max_abroad_freq	numeric	--the maximum abroad call frequency in a month	calling pattern
imax_abroad_freq	numeric	--the month when the maximum abroad call frequency occurs	calling pattern
sum_abroad_freq	numeric	--total abroad call frequency over three months precede the month the data extraction	calling pattern
avg_abroad_freq	numeric	--average monthly abroad call frequency	calling pattern
max_totalout_vol	numeric	--the maximum total orginating call volume in a month	calling pattern
imax_totalout_vol	numeric	--the month when the maximum total orginating call volume occurs	calling pattern
sum_totalout_vol	numeric	--sum of total orginating call volume over three months precede the month the data extraction	calling pattern
avg_totalout_vol	numeric	--average monthly total orginating call volume	calling pattern
max_totalout_freq	numeric	--the maximum total orginating call frequency in a month	calling pattern
imax_totalout_freq	numeric	--the month when the maximum total orginating call frequency occurs	calling pattern
sum_totalout_freq	numeric	--sum of total orginating call frequency over three months precede the month the data extraction	calling pattern
avg_totalout_freq	numeric	--average monthly total inside network call frequency	calling pattern
max_totalin_vol	numeric	--the maximum total terminating call volume in a month	calling pattern
imax_totalin_vol	numeric	--the month when the maximum total terminating call volume occurs	calling pattern
sum_totalin_vol	numeric	--sum of total terminating call volume over three months precede the month the data extraction	calling pattern
avg_totalin_vol	numeric	--average monthly total terminating call volume	calling pattern
max_totalin_freq	numeric	--the maximum total terminating call frequency in a month	calling pattern
imax_totalin_freq	numeric	--the month when the maximum total terminating call frequency occurs	calling pattern
sum_totalin_freq	numeric	--sum of total terminating call frequency over three months precede the month the data extraction	calling pattern
avg_totalin_freq	numeric	--average monthly total terminating call frequency	calling pattern
max_smsout	numeric	--the maximum sending sms frequency in a month	calling pattern
imax_smsout	numeric	--the month when the maximum sending sms frequency occurs	calling pattern
sum_smsout	numeric	--total sending sms frequency over three months precede the month the data extraction	calling pattern
avg_smsout	numeric	--average monthly sending sms frequency	calling pattern

max_smsin	numeric	--the maximum receiving sms frequency in a month	calling pattern
imax_smsin	numeric	--the month when the maximum receiving sms frequency occurs	calling pattern
sum_smsin	numeric	--total receiving sms frequency over three months precede the month the data extraction	calling pattern
avg_smsin	numeric	--average monthly receiving sms frequency	calling pattern
max_innet_vol_ratio	numeric	--the maximum inside network to total call volume ratio in a month	calling pattern
imax_innet_vol_ratio	numeric	--the month when the maximum inside network to total call volume ratio occurs	calling pattern
max_innet_freq_ratio	numeric	--the maximum inside network to total call frequency ratio in a month	calling pattern
imax_innet_freq_ratio	numeric	--the month when the maximum inside network to total call frequency ratio occurs	calling pattern
max_outnet_vol_ratio	numeric	--the maximum outside network to total call volume ratio in a month	calling pattern
imax_outnet_vol_ratio	numeric	--the month when the maximum outside network to total call volume ratio occurs	calling pattern
max_outnet_freq_ratio	numeric	--the maximum outside network to total call frequency ratio in a month	calling pattern
imax_outnet_freq_ratio	numeric	--the month when the maximum outside network to total call frequency ratio occurs	calling pattern
max_abroad_vol_ratio	numeric	--the maximum abroad to total call volume ratio in a month	calling pattern
imax_abroad_vol_ratio	numeric	--the month when the maximum abroad to total call volume ratio occurs	calling pattern
max_abroad_freq_ratio	numeric	--the maximum abroad to total call frequency ratio in a month	calling pattern
imax_abroad_freq_ratio	numeric	--the month when the maximum abroad to total call frequency ratio occurs	calling pattern
innet_charge	numeric	--charged amount due to inside network call one month before the month of the data extraction	cdr billed
outnet_charge	numeric	--charged amount due to outside network call one month before the month of the data extraction	cdr billed
abroad_charge	numeric	--charged amount due to abroad call one month before the month of the data extraction	cdr billed
innet_tcharge_rat	numeric	--ratio of inside network call to total charged amount one month before the month of the data extraction	cdr billed
outnet_tcharge_rat	numeric	--ratio of outside network call to total charge amount one month before the month of the data extraction	cdr billed
abroad_tcharge_rat	numeric	--ratio of abroad call to total charge amount one month before the month of the data extraction	cdr billed
sms_innet_charge	numeric	--charged amount due to sending sms inside network one month before the month of the data extraction	cdr billed
sms_outnet_charge	numeric	--charged amount due to sending sms outside network one month before the month of the data extraction	cdr billed
sms_abroad_charge	numeric	--charged amount due to sendind sms abroad one month before the month of the data extraction	cdr billed

sms_innet_tcharge_rat	numeric	--ratio of inside network sms sending to total charged amount one month before the month of the data extraction	cdr billed
sms_outnet_tcharge_rat	numeric	--ratio of outside network sms sending to total charged amount one month before the month of the data extraction	cdr billed
sms_abroad_tcharge_rat	numeric	--ratio of abroad sms sending to total charged amount one month before the month of the data extraction	cdr billed
tcharge	numeric	--total charged amount one month before the month of the data extraction	cdr billed
innet_charge1	numeric	--charged amount due to inside network call two months before the month of the data extraction	cdr billed
outnet_charge1	numeric	--charged amount due to outside network call two months before the month of the data extraction	cdr billed
abroad_charge1	numeric	--charged amount due to abroad call two months before the month of the data extraction	cdr billed
innet_tcharge_rat1	numeric	--ratio of inside network call to total charged amount two months before the month of the data extraction	cdr billed
outnet_tcharge_rat1	numeric	--ratio of outside network call to total charge amount two months before the month of the data extraction	cdr billed
abroad_tcharge_rat1	numeric	--ratio of abroad call to total charge amount two months before the month of the data extraction	cdr billed
sms_innet_charge1	numeric	--charged amount due to sending sms inside network two months before the month of the data extraction	cdr billed
sms_outnet_charge1	numeric	--charged amount due to sending sms outside network two months before the month of the data extraction	cdr billed
sms_abroad_charge1	numeric	--charged amount due to sendind sms abroad two months before the month of the data extraction	cdr billed
sms_innet_tcharge_rat1	numeric	--ratio of inside network sms sending to total charged amount two months before the month of the data extraction	cdr billed
sms_outnet_tcharge_rat1	numeric	--ratio of outside network sms sending to total charged amount two months before the month of the data extraction	cdr billed
sms_abroad_tcharge_rat1	numeric	--ratio of abroad sms sending to total charged amount two months before the month of the data extraction	cdr billed
tcharge1	numeric	--total charged amount two months before the month of the data extraction	cdr billed
innet_charge2	numeric	--charged amount due to inside network call three months before the month of the data extraction	cdr billed
outnet_charge2	numeric	--charged amount due to outside network call three months before the month of the data extraction	cdr billed
abroad_charge2	numeric	--charged amount due to abroad call three months before the month of the data extraction	cdr billed
innet_tcharge_rat2	numeric	--ratio of inside network call to total charged amount three months before the month of the data extraction	cdr billed

outnet_tcharge_rat2	numeric	--ratio of outside network call to total charge amount three months before the month of the data extraction	cdr billed
abroad_tcharge_rat2	numeric	--ratio of abroad call to total charge amount three months before the month of the data extraction	cdr billed
sms_innet_charge2	numeric	--charged amount due to sending sms inside network three months before the month of the data extraction	cdr billed
sms_outnet_charge2	numeric	--charged amount due to sending sms outside network three months before the month of the data extraction	cdr billed
sms_abroad_charge2	numeric	--charged amount due to sendind sms abroad three months before the month of the data extraction	cdr billed
sms_innet_tcharge_rat2	numeric	--ratio of inside network sms sending to total charged amount three months before the month of the data extraction	cdr billed
sms_outnet_tcharge_rat2	numeric	--ratio of outside network sms sending to total charged amount three months before the month of the data extraction	cdr billed
sms_abroad_tcharge_rat2	numeric	--ratio of abroad sms sending to total charged amount three months before the month of the data extraction	cdr billed
tcharge2	numeric	--total charged amount three months before the month of the data extraction	cdr billed
max_innet_charge	numeric	--the maximum charged amount due to inside network call in a month	cdr billed
max_outnet_charge	numeric	--the maximum charged amount due to outside network call in a month	cdr billed
max_abroad_charge	numeric	--the maximum charged amount due to abroad call in a month	cdr billed
max_innet_tcharge_rat	numeric	--the maximum inside network call to total charged amount ratio in a month	cdr billed
max_outnet_tcharge_rat	numeric	--the maximum outside network call to total charged amount ratio in a month	cdr billed
max_abroad_tcharge_rat	numeric	--the maximum abroad call to total charged amount ratio in a month	cdr billed
max_s_innet_charge	numeric	--the maximum charged amount due to sending sms inside network in a month	cdr billed
max_s_outnet_charge	numeric	--the maximum charged amount due to sending sms outside network in a month	cdr billed
max_s_abroad_charge	numeric	--the maximum charged amount due to sending sms abroad in a month	cdr billed
max_s_innet_tcharge_rat	numeric	--the maximum inside network sms sending to total charged amount ratio in a month	cdr billed
max_s_outnet_tcharge_rat	numeric	--the maximum outside network sms sending to total charged amount ratio in a month	cdr billed
max_s_abroad_tcharge_rat	numeric	--the maximum abroad sms sending to total charged amount ratio in a month	cdr billed
max_tcharge	numeric	--the maximum total charged amount in a month	cdr billed
imax_innet_charge	numeric	--the month when the maximum charged amount due to inside network call occurs	cdr billed
imax_outnet_charge	numeric	--the month when the maximum charged amount due to outside network call occurs	cdr billed

imax_abroad_charge	numeric	--the month when the maximum charged amount due to abroad call occurs	cdr billed
imax_innet_tcharge_rat	numeric	--the month when the maximum inside network call to total charged amount ratio occurs	cdr billed
imax_outnet_tcharge_rat	numeric	--the month when the maximum outside network call to total charged amount ratio occurs	cdr billed
imax_abroad_tcharge_rat	numeric	--the month when the maximum abroad call to total charged amount ratio occurs	cdr billed
imax_s_innet_charge	numeric	--the month when the maximum charged amount due to sending sms inside network occurs	cdr billed
imax_s_outnet_charge	numeric	--the month when the maximum charged amount due to sending sms outside network occurs	cdr billed
imax_s_abroad_charge	numeric	--the month when the maximum charged amount due to sending sms abroad occurs	cdr billed
imax_s_innet_tcharge_rat	numeric	--the month when the maximum inside network sms sending to total charged amount ratio occurs	cdr billed
imax_s_outnet_tcharge_rat	numeric	--the month when the maximum outside network sms sending to total charged amount ratio occurs	cdr billed
imax_s_abroad_tcharge_rat	numeric	--the month when the maximum abroad sms sending to total charged amount ratio occurs	cdr billed
imax_tcharge	numeric	--the month when the maximum total charged amount occurs	cdr billed
sum_innet_charge	numeric	--total charged amount due to inside network call over three months precede the month the data extraction	cdr billed
sum_outnet_charge	numeric	--total charged amount due to outside network call over three months precede the month the data extraction	cdr billed
sum_abroad_charge	numeric	--total charged amount due to abroad call over three months precede the month the data extraction	cdr billed
sum_s_innet_charge	numeric	--total charged amount due to sending sms inside network over three months precede the month the data extraction	cdr billed
sum_s_outnet_charge	numeric	--total charged amount due to sending sms outside network over three months precede the month the data extraction	cdr billed
sum_s_abroad_charge	numeric	--total charged amount due to sending sms abroad over three months precede the month the data extraction	cdr billed
sum_tcharge	numeric	--sum of total charged amount over three months precede the month the data extraction	cdr billed
avg_innet_charge	numeric	--average monthly charged amount due to inside network call	cdr billed
avg_outnet_charge	numeric	--average monthly charged amount due to outside network call	cdr billed
avg_abroad_charge	numeric	--average monthly charged amount due to abroad call	cdr billed
avg_s_innet_charge	numeric	--average monthly charged amount due to sending sms inside network	cdr billed
avg_s_outnet_charge	numeric	--average monthly charged amount due to sending sms outside network	cdr billed
avg_s_abroad_charge	numeric	--average monthly charged amount due to sending sms abroad	cdr billed
avg_tcharge	numeric	--average monthly total charged amount	cdr billed

num_neibor	numeric	--number of neighbours one month before the month of the data extraction	network
num_churn_neibor	numeric	--number of churn neighbours one month before the month of the data extraction	network
churn_neibor_rat	numeric	--Ratio of churn neighbours w.r.t. neighbours one month before the month of the data extraction	network
degree_centrality	numeric	--degree centrality of customer one month before the month of the data extraction	network
total_dc_neibor	numeric	--total degree centrality of neighbours one month before the month of the data extraction	network
total_dc_churn_neibor	numeric	--total degree centrality of churn neighbours one month before the month of the data extraction	network
degree_rat	numeric	--ratio of total degree centrality of churn neighbours w.r.t. neighbours one month before the month of the data extraction	network
pagerank	numeric	--page rank of customer one month before the month of the data extraction	network
total_pr_neibor	numeric	--total page rank of neighbours one month before the month of the data extraction	network
total_pr_churn_neibor	numeric	--total page rank of churn neighbours one month before the month of the data extraction	network
pagerank_rat	numeric	--ratio of total pagerank of churn neighbours w.r.t. neighbours one month before the month of the data extraction	network
to_churn_neibor_vol	numeric	--total call volume to churn neighbours one month before the month of the data extraction	network
from_churn_neibor_vol	numeric	--total call volume from churn neighbours one month before the month of the data extraction	network
to_neibor_vol	numeric	--total orginating call volume to neighbours one month before the month of the data extraction	network
from_neibor_vol	numeric	--total terminating call volume from neighbours one month before the month of the data extraction	network
to_churn_neibor_vol_rat	numeric	--Ratio of call volume to churn neighbours w.r.t. total orginating call volume one month before the month of the data extraction	network
from_churn_neibor_vol_rat	numeric	--Ratio of call volume from churn neighbours w.r.t. total terminating call volume one month before the month of the data extraction	network
num_neibor1	numeric	--number of neighbours two months before the month of the data extraction	network
num_churn_neibor1	numeric	--number of churn neighbours two months before the month of the data extraction	network
churn_neibor_rat1	numeric	--Ratio of churn neighbours w.r.t. neighbours two months before the month of the data extraction	network
degree_centrality1	numeric	--degree centrality of customer two months before the month of the data extraction	network
total_dc_neibor1	numeric	--total degree centrality of neighbours two months before the month of the data extraction	network
total_dc_churn_neibor1	numeric	--total degree centrality of churn neighbours two months before the month of the data extraction	network
degree_rat1	numeric	--ratio of total degree centrality of churn neighbours w.r.t. neighbours two months before the month of the data extraction	network
pagerank1	numeric	--page rank of customer two months before the month of the data extraction	network

total_pr_neibor1	numeric	--total page rank of neighbours two months before the month of the data extraction	network
total_pr_churn_neibor1	numeric	--total page rank of churn neighbours two months before the month of the data extraction	network
pagerank_rat1	numeric	--ratio of total pagerank of churn neighbours w.r.t. neighbours two months before the month of the data extraction	network
to_churn_neibor_vol1	numeric	--total call volume to churn neighbours two months before the month of the data extraction	network
from_churn_neibor_vol1	numeric	--total call volume from churn neighbours two months before the month of the data extraction	network
to_neibor_vol1	numeric	--total orginating call volume to neighbours two months before the month of the data extraction	network
from_neibor_vol1	numeric	--total terminating call volume from neighbours two months before the month of the data extraction	network
to_churn_neibor_vol_rat1	numeric	--Ratio of call volume to churn neighbours w.r.t. total orginating call volume two months before the month of the data extraction	network
from_churn_neibor_vol_rat1	numeric	--Ratio of call volume from churn neighbours w.r.t. total terminating call volume two months before the month of the data extraction	network
num_neibor2	numeric	--number of neighbours three months before the month of the data extraction	network
num_churn_neibor2	numeric	--number of churn neighbours three months before the month of the data extraction	network
churn_neibor_rat2	numeric	--Ratio of churn neighbours w.r.t. neighbours three months before the month of the data extraction	network
degree_centrality2	numeric	--degree centrality of customer three months before the month of the data extraction	network
total_dc_neibor2	numeric	--total degree centrality of neighbours three months before the month of the data extraction	network
total_dc_churn_neibor2	numeric	--total degree centrality of churn neighbours three months before the month of the data extraction	network
degree_rat2	numeric	--ratio of total degree centrality of churn neighbours w.r.t. neighbours three months before the month of the data extraction	network
pagerank2	numeric	--page rank of customer three months before the month of the data extraction	network
total_pr_neibor2	numeric	--total page rank of neighbours three months before the month of the data extraction	network
total_pr_churn_neibor2	numeric	--total page rank of churn neighbours three months before the month of the data extraction	network
pagerank_rat2	numeric	--ratio of total pagerank of churn neighbours w.r.t. neighbours three months before the month of the data extraction	network
to_churn_neibor_vol2	numeric	--total call volume to churn neighbours three months before the month of the data extraction	network
from_churn_neibor_vol2	numeric	--total call volume from churn neighbours three months before the month of the data extraction	network
to_neibor_vol2	numeric	--total orginating call volume to neighbours three months before the month of the data extraction	network

from_neibor_vol2	numeric	--total terminating call volume from neighbours three months before the month of the data extraction	network
to_churn_neibor_vol_rat2	numeric	--Ratio of call volume to churn neighbours w.r.t. total orginating call volume three months before the month of the data extraction	network
from_churn_neibor_vol_rat2	numeric	--Ratio of call volume from churn neighbours w.r.t. total terminating call volume three months before the month of the data extraction	network
max_num_neibor	numeric	--the maximum number of neighbours in a month	network
max_num_churn_neibor	numeric	--the maximum number of churn neighbours in a month	network
max_churn_neibor_rat	numeric	--the maximum Ratio of churn neighbours w.r.t. neighbours in a month	network
max_degree_centrality	numeric	--the maximum degree centrality of customer in a month	network
max_total_dc_neibor	numeric	--the maximum total degree centrality of neighbours in a month	network
max_total_dc_churn_neibor	numeric	--the maximum total degree centrality of churn neighbours in a month	network
max_degree_rat	numeric	--the maximum ratio of total degree centrality of churn neighbours w.r.t. neighbours in a month	network
max_pagerank	numeric	--the maximum page rank of customer in a month	network
max_total_pr_neibor	numeric	--the maximum total page rank of neighbours in a month	network
max_total_pr_churn_neibor	numeric	--the maximum total page rank of churn neighbours in a month	network
max_pagerank_rat	numeric	--the maximum ratio of total pagerank of churn neighbours w.r.t. neighbours in a month	network
max_to_churn_neibor_vol	numeric	--the maximum total call volume to churn neighbours in a month	network
max_from_churn_neibor_vol	numeric	--the maximum total call volume from churn neighbours in a month	network
max_to_neibor_vol	numeric	--the maximum total orginating call volume to neighbours in a month	network
max_from_neibor_vol	numeric	--the maximum total terminating call volume from neighbours in a month	network
max_to_churn_neibor_vol_rat	numeric	--the maximum Ratio of call volume to churn neighbours w.r.t. total orginating call volume in a month	network
max_from_churn_neibor_vol_rat	numeric	--the maximum Ratio of call volume from churn neighbours w.r.t. total terminating call volume in a month	network
imax_num_neibor	numeric	--the month when the maximum number of neighbours occurs	network
imax_num_churn_neibor	numeric	--the month when the maximum number of churn neighbours occurs	network
imax_churn_neibor_rat	numeric	--the month when the maximum Ratio of churn neighbours w.r.t. neighbours occurs	network
imax_degree_centrality	numeric	--the month when the maximum degree centrality of customer occurs	network
imax_total_dc_neibor	numeric	--the month when the maximum total degree centrality of neighbours occurs	network
imax_total_dc_churn_neibor	numeric	--the month when the maximum total degree centrality of churn neighbours occurs	network

imax_degree_rat	numeric	--the month when the maximum ratio of total degree centrality of churn neighbours w.r.t. neighbours occurs	network
imax_pagerank	numeric	--the month when the maximum page rank of customer occurs	network
imax_total_pr_neibor	numeric	--the month when the maximum total page rank of neighbours occurs	network
imax_total_pr_churn_neibor	numeric	--the month when the maximum total page rank of churn neighbours occurs	network
imax_pagerank_rat	numeric	--the month when the maximum ratio of total pagerank of churn neighbours w.r.t. neighbours occurs	network
imax_to_churn_neibor_vol	numeric	--the month when the maximum total call volume to churn neighbours occurs	network
imax_from_churn_neibor_vol	numeric	--the month when the maximum total call volume from churn neighbours occurs	network
imax_to_neibor_vol	numeric	--the month when the maximum total orginating call volume to neighbours occurs	network
imax_from_neibor_vol	numeric	--the month when the maximum total terminating call volume from neighbours occurs	network
imax_to_churn_neibor_vol_rat	numeric	--the month when the maximum Ratio of call volume to churn neighbours w.r.t. total orginating call volume occurs	network
imax_from_churn_neibor_vol_rat	numeric	--the month when the maximum Ratio of call volume from churn neighbours w.r.t. total terminating call volume occurs	network
sum_num_neibor	numeric	--total number of neighbours	network
sum_num_churn_neibor	numeric	--total number of churn neighbours over three months precede the month the data extraction	network
sum_degree_centrality	numeric	--total degree centrality of customer over three months precede the month the data extraction	network
sum_total_dc_neibor	numeric	--sum of total degree centrality of neighbours over three months precede the month the data extraction	network
sum_total_dc_churn_neibor	numeric	--sum of total degree centrality of churn neighbours over three months precede the month the data extraction	network
sum_pagerank	numeric	--total page rank of customer over three months precede the month the data extraction	network
sum_total_pr_neibor	numeric	--sum of total page rank of neighbours over three months precede the month the data extraction	network
sum_total_pr_churn_neibor	numeric	--sum of total page rank of churn neighbours over three months precede the month the data extraction	network
sum_to_churn_neibor_vol	numeric	--sum of total call volume to churn neighbours over three months precede the month the data extraction	network
sum_from_churn_neibor_vol	numeric	--sum of total call volume from churn neighbours over three months precede the month the data extraction	network
sum_to_neibor_vol	numeric	--sum of total orginating call volume to neighbours over three months precede the month the data extraction	network
sum_from_neibor_vol	numeric	--sum of total terminating call volume from neighbours over three months precede the month the data extraction	network

avg_num_neibor	numeric	--average monthly number of neighbours	network
avg_num_churn_neibor	numeric	--average monthly number of churn neighbours	network
avg_degree_centrality	numeric	--average monthly degree centrality of customer	network
avg_total_dc_neibor	numeric	--average monthly total degree centrality of neighbours	network
avg_total_dc_churn_neibor	numeric	--average monthly total degree centrality of churn neighbours	network
avg_pagerank	numeric	--average monthly page rank of customer	network
avg_total_pr_neibor	numeric	--average monthly total page rank of neighbours	network
avg_total_pr_churn_neibor	numeric	--average monthly total page rank of churn neighbours	network
avg_to_churn_neibor_vol	numeric	--average monthly total call volume to churn neighbours	network
avg_from_churn_neibor_vol	numeric	--average monthly total call volume from churn neighbours	network
avg_to_neibor_vol	numeric	--average monthly total orginating call volume to neighbours	network
avg_from_neibor_vol	numeric	--average monthly total terminating call volume from neighbours	network