Understanding Customer Behavior using Machine Learning

A Project Work
submitted in partial fulfillment of the
requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering with specialization in Data Analytics

Submitted By

Rohan Bali
(1613112039)

Under the supervision of
Dr. Satyajee Srivastava
Professor

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA – 201306
MAY 2020

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this projectreport **“UNDERSTANDING CUSTOMER BEHAVIOUR USING MACHINE LEARNING** is the bonafide work of **“ROHAN BALI (1613112039)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD
Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science &
Engineering

SIGNATURE OF SUPERVISOR
Dr. SATYAJEE SRIVASTAVA, Ph.D.,
Professor
School of Computing Science &
Engineering

ABSTRACT

This project deals with understanding the customer behavior using different machine learning techniques.

The study of customer behavior both in online and offline purchases plays a very important role for the seller. The aim of this study is to identify customers on various parameters and thus re-define policies based on the behavior of customers. This paper works on churn analytics for retaining customers, a market-based analysis for identifying the support and confidence among products and a recommendation system built on the IBCF approach. Churn Analytics helps the seller to answer about whether the customers are leaving there products or services. The goal of every seller is to maintain a low churn rate and thus have large margins and bigger profits. Further, performing a market-based analysis can be very fruitful for a supermart. This approach helps in organizing the items in a store in an efficient and scientific manner. This paper conducts the above analysis using the 'Apriori' algorithm. To conclude, a recommendation system is used to suggest customers products based on the history of their purchase or the similarities of that product with other products or other consumers. Thus, this study will help in understanding various aspects of customer behavior.

TABLE OF CONTENTS

1 **Introduction.. 1**

1.1 **Motivation and background.. 2**

1.2 **Theoretical framework and focus of the study......................... 3**

1.3 **Research questions and objectives.. 4**

1.4 **Methodology 4**

1.5 **Structure of the thesis.. 5**

2 **Machine learning.. 6**

2.1 **Data preprocessing and model optimization........................... 6**

2.1.1 Data cleaning, normalization, and transformation. 7

2.1.2 Missing data.. 8

2.1.3 Sampling.. 8

2.1.4 Feature and variable selection... 9

2.1.5 Hyperparameter optimization...10

2.2 **Methods..11**

2.2.1 Logistic regression 12

2.2.2 Decision trees..13

2.2.3 Ensemble methods 15

2.2.4 Naïve Bayesian...16

2.2.5 Support vector machine...16

2.2.6 K-nearest neighbor 17

2.2.7 Artificial neural networks..18

2.3 Model evaluation..19

2.3.1 Validation 19

2.3.2 Confusion matrix 20

2.3.3 Receiver operating characteristic curve 22

2.3.4 Top-Decile Lift..23

2.3.5 Mean squared error..23

3 Literature review on customer churn prediction 25

3.1 Methodology 23

3.2 Literature search process..26

3.3 Customer churn prediction 26

3.3.1 Review on the customer churn modeling field.........................29

3.3.2 Customer churn prediction in insurance.................................32

Summary...33

4 Developing machine learning models to predict future customers study...35

4.1 Tools and libraries...35

4.2 Data description and considerations......................................35

4.3 Data preprocessing and feature selection 36

4.3.1 Categorical values..37

4.3.2 Handling missing data 38

4.3.3 Data normalization..39

4.3.4 Feature Selection..39

4.3.5 Imbalanced data..41

4.4 Structure of processed data..43

4.5 Model selection..44

4.6 Model evaluation..46

5 Model development and results..47

5.1 Logistic regression..47

5.2 Support vector machines..48

5.3 Random forests..49

5.4 K-neighbors classifier..50

5.5 AdaBoost with Decision trees..52

5.6 Artificial neural network..54

5.7 Confusion matrixes..58

5.8 Receiver operating characteristic curves..60

5.9 Probability modeling..61

5.10 Summary and analysis of the results..62

6 Conclusions..66

6.1.1 Analysis of results..66

6.1.2 Limitations and future research..68

7 References..69

List of Abbreviations

AUC	Area Under the Curve
AB	Ada Boost
CCP	Customer churn prediction
CM	Confusion Matrix
CRM	Customer-Relations Management
CV	Cross-Validation
GAM	Generalized additive models
ER	Error rate
ET	Extra trees / Extremely randomized trees
FN	False Negative
FFNN	Feed-Forward Neural Network
FP	False Positive
GAM	Generalized additive model
GLM	General Linear Models
KNN	k-Nearest Neighbors
LR	Logistic Regression
LLM	Logit leaf model
ML	Machine Learning
MSE	Mean squared error

TDL	Top-Decile lift
T/E	Training/Evaluation
NN	(Artificial) Neural Network
PCA	Principal component analysis
PCC	Percentage correctly classified (accuracy)
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic curve
SVM	Support Vector Machines
SGB	Stochastic gradient boost
TN	True negative
TP	True positive

List of Figures

Figure 1 Research area of this thesis 3

Figure 2 Maximum likelihood (Wicklin, 2011) 13

Figure 3 Simple decision tree based on binary variable Y (Song and Lu, 2015)14

Figure 4 SVM 17

Figure 6 Example of ANN.. 18

Figure 6 Confusion matrix and performance metrics (Fawcett, 2006) 21

Figure 7 A ROC curve (Glen, 2016) 23

Figure 8 Model building process 35

Figure 9 Distribution of the duration of customer relationship 37

Figure 10 Language distribution... 38

Figure 11 Gender distribution 38

Figure 12 Distribution of quality of business relationship 39

Figure 13 Importance of features selected by the algorithm........................... 40

Figure 14 Distribution of churners in the data using the first selected data 41

Figure 15 Distribution of churners in the data using the last available data…….41

Figure 16 Distribution of churners and non-churners by customer relationship..43

Figure 17 Distribution of churners in new and old customer datasets............. 44

List of Tables

Table 1 5-fold cross-validation.. 20

Table 2 approaches to literature review (Webster and Watson, 2002) 25

Table 3 Authors & year, model, algorithms and data................................... 27

Table 4 Concept matrix of different algorithms and evaluation...................... 48

Table 5 LR metrics.. 48

Table 6 SVM metrics... 48

Table 7 RF metrics before a grid search 48

Table 8 RF metrics after a grid search (change)... 49

Table 9 KNN metrics before a grid search 51

Table 10 KNN metrics after a grid search 51

Table 11 AB metrics before a grid search 53

Table 12 AB metrics after a grid search.. 53

Table 13 ANN metrics before enhancements 54

Table 14 ANN accuracy by epoch before grid search 55

Table 15 ANN metrics after a grid search 57

Table 16 ANN Accuracy by epoch after a grid search 58

Table 17 Confusion matrixes 59

Table 18 ROC-curves 61

Table 19 MSE of different models on datasets... 62

Table 20 Summary of average metrics 64

Table 21 Average prediction scores by dataset 65

1 Introduction

Customer churn prediction (CCP) is a form of customer relationship management (CRM) in which a company tries to create a model that predicts if a customer is planning on leaving or reducing its purchases from a company. CCP is studied very commonly across different industries such as telecommunications, retail markets, subscription management, financial services, and electronic commerce (Chen, Fan and Sun, 2012). Companies use machine learning (ML) based methods for customer churn prediction. ML is a field that intersects between computer science and statistics (Jordan and Mitchell, 2015).

The motivation for CCP comes from the point made in CRM, which is that companies hold valuable information about their customers in their databases (Herman, 1965; Jones, Mothersbaugh and Beatty, 2000; Thomas, 2001). The data can be used to assess whether a customer could be leaving and what could be the reasons for that. Since it is more profitable to keep existing customers compared to attracting new ones (Reinartz and Kumar, 2003), it makes sense for companies to try to predict leaving customers and try to prevent them from leaving or decreasing purchases. CCP has, therefore, become a field with much research with different methods, which are very well introduced in the seminar works of Verbeke, Martens, Muse, & Baesens, before 2011 and between 2011 and 2017 by De Caigny, Coussement, & De Bock, 2018. As an overview, many of the models before 2011 were using logistic regression (LR), decision trees (DT) but some were already using more modern methods, for example, artificial neural networks (ANN), random forests (RF), and support vector machines (SVM). In 2015 Mahajan, Misra, and Mahajan researched the telecommunication industry and found that DTs, LR, and ANN were still on top of most used models.

The general idea, with most ML-applications, is that the dataset is split into a test and training data. Then the training data is fed to an ML-model which learns from the data. Then the model is fed the yet unseen test data from which it predicts the results, which are then compared to real values. From the differences between the predictions and real values, metrics of how good the model is, are calculated (Louridas and Ebert, 2016). The method of using machine learning methods to make predictions has become increasingly popular as the volumes of data are continually increasing (Louzada, Ara and Fernandes, 2016). Predictions for the future can be valuable since they allow companies to adjust better to the possible future (Roos and Gustafsson, 2007).

This work has two parts: first, to explain concepts and review the literature on predicting customer churn with machine learning. Second, to create a model that predicts customer churn for the next period (one year) with machine learning (ML) methods and compare the performance of these methods to methods currently in use.

1.1 Motivation and background

Insurance, in general, is based on pricing individual risk profile and adding some premium on top of the value that is calculated for that risk (David, 2015). This has led to an industry where analytics is paramount for business success. Since overpricing means, fewer customers, and too low prices mean potential losses for the company. For this reason, the insurance industry is commonly known to have gathered detailed information about their customers, to correctly price the customer-specific risks. For the data-intensive insurance industry, the ML-based applications provide a fruitful avenue of research (Jordan and Mitchell, 2015). Some of the existing studies in the field include fraud detection using machine learning (Kirlidog and Asuk, 2012; Bayerstadler, Van Dijk and Winter, 2016), which have helped insurance companies to speed up the processing times and remove fraudulent from compensation requests. CCP is very fitting for the insurance business since firstly, acquiring new customers can be 12 times the cost of retaining one (Torkzadeh, Chang and Hansen, 2006) secondly, the insurance is regarded as “mostly a necessary evil” (Gidhagen and Persson, 2011) which makes customers harder to find, and thirdly, customers and insurance companies are in contact very infrequently (Mau, Pletikosa and Wagner, 2018) which makes it harder to have early indicators on customer churn. All previous points amplify the need for some customer retention management or CCP.

The purpose of this study is to provide insurance companies with an effective method to help predict whether the customer relationship will be renewed after the first period or not. There is already some prediction research on insurance customer profitability (Fang, Jiang and Song, 2016), but it does not try to model how the customer relationship will continue after the first period. The model proposed in this study should predict the future churn of the customer after the first period, regardless of whether the customer is a new or an existing or has another insurance product. The future churn of the customer is of interest because the insurance company can target and attract customers that offer better longevity with loss leaders that would turn into profit later.

1.2 Theoretical framework and focus of the study

This study focuses mainly on the literature on CCP with machine learning to find a model that is best suitable for predicting the churn of an insurance customer from a dataset containing the information of private customers. After that, the focus is on empirical research and developing to make such a model with the provided data. Then the study compares the reliability and accuracy of the suggested machine learning model to the previous logistic regression model used in the insurance company providing the data.

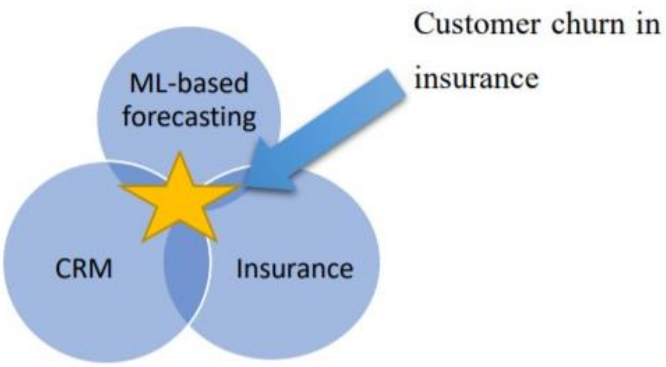


Figure 1 Research area of this thesis

1.3 Research questions and objectives

The main goal for this thesis is to predict the future churn or customer status (stays/churns) for an insurance customer for the next period (one year) when he or she is acquiring new private insurance such as a car, life or property insurance. The model should be able to predict the churn for both new and old customers. To create a good model, a solid overview of the machine learning field regarding our prediction of customer relationships and possible applications to the insurance industry is needed. In addition, this study compares proposed methods to logistic regression, since it is the current statistical method used in the Finnish insurance company considered in this study. Based on the objectives and the specific data type that we have; the primary and sub-research questions are formulated below:

1. What is the current state of customer churn prediction in the literature?
   a. What algorithms are used in customer churn prediction, and how are they evaluated?
   b. What is the current state of customer churn prediction literature on the insurance field?
2. What is the most suitable machine learning model to be used to predict future customer churn for the given dataset on customer feature data?
   a. How different methods compare to one another?

1.4 Methodology

This thesis was conducted in three parts. Research questions were formed based on wanted outcomes and the literature. The first part was to make a sufficient compilation of the literature more widely and then narrow it down to get a good overview. The term CCP was taken as a focal point for the literature review since the goal of this study is to try to predict the churn/retention of a customer after the first period. A review of the CCP is conducted, which serves as the basis to answer the first research question. Second, based on the literature review, the most suitable ML-methods were selected for further studies, and their predictive performances are compared. The results are then analyzed and reported, and the suggestions for the most suitable method are given.

The data for this study was obtained from a Finnish insurance company and consisted of real customer data from the year 2016 to 2018 since the data is precious for the provider; it cannot be made publicly available alongside this thesis.

1.5 Structure of the paper

The structure of this thesis is as follows. First, in chapter 2, critical methodologies and concepts are explained at a high-level, which are required to understand this thesis. It includes the introduction of the ML field, different models, and what is the process of building an ML model is. In chapter 3, an overview of the past and current literature, issues, and development on the churn prediction field are reviewed, and the classifiers for this study are chosen. Next, in chapter 4, the experimental process of developing the ML model, decisions, and considerations for this study are explained. In chapter 5, the results are explained and analyzed, and the answers for the second subset of research questions are answered. Chapter 6 discusses the results and limitations and represents the conclusions along with proposals for future research on the topic.

2 Machine learning

Recently, the interest in ML has increased since the amount of computing power, and the amount of data gathered has increased tremendously (Louridas and Ebert, 2016). The term machine learning can be defined as "computational methods using the experience to improve performance or to make accurate predictions." Experience, in this case, means information about the past, which is often electronic data, which size and quality have tremendous importance to the success of the predictions that the algorithms will be making.

Standard machine learning tasks include *classification, regression, ranking, clustering,* and *dimensionality reduction,* or *manifold learning. Classification* is a problem of finding the correct category for inputs. These problems can be, for example, image classifications, text classification, or finding a proper customer segment for a customer. *Regression* is a problem where a value needs to be determined for an input. For example, future stock value or duration of the customer relationship. In *Ranking,* the problem is to order items with some criteria, for example, web searches. *Clustering* means to try to partition the data to homogenous groups that are not yet known. For example, a company might wish to find new customer segments or in social networks to find communities. *Dimensionality reduction* or *manifold learning* means to reduce the representation of data to lower-dimensional representation. The question of this study is whether a customer is going to be churned or not, which is a typical classification problem between 1 and 0. That is why the methods presented in this chapter are used in classification problems. (Mohri, Rostamizadeh and Talwalkar, 2018, 1-3)

Machine learning methods can be divided into *supervised learning, unsupervised learning,* where the main difference is that with supervised learning, the data is labeled, and in unsupervised learning, it is not. An everyday use case for unsupervised learning is clustering or dimension reduction and for example, email spam filter for supervised learning. (Mohri, Rostamizadeh and Talwalkar, 2018, 6-7)

2.1 Data preprocessing and model optimization

Data preprocessing is an essential part of creating a machine learning model. It has an impact on the generalization performance of the model and on improving the understandability of the model. Data preprocessing includes such things as data cleaning, normalization, transformation, feature extraction, or selection, amongst others. (Kotsiantis, Kanellopoulos and Pintelas, 2006) Data preprocessing or preparation can be separated into value *transformation* (cleaning, normalization, transformation, handling missing values, etc.) and value *representation* (variable selection and evaluation) (Coussement, Lessmann and Verstraeten, 2017).

2.1.1 Data cleaning, normalization, and transformation.

Data cleaning is the process of checking the quality of the data, and there are two approaches: filtering and wrapping. *Filtering* is concerned just with the removal of data with predefined rules, i.e., removing outliers, misspelled words, duplicates, or impossible data, such as over 120-year-old customers. *Wrapping* which focuses more on the quality of data by detecting and removing mislabeled data. (Kotsiantis, Kanellopoulos and Pintelas, 2006)

Normalization means to "scale down" the features by leveling the absolute values to the same scale. It is crucial for many algorithms such as ANNs and KNN, to prevent bias towards values that are on different scales. Normalization can be done using multiple methods, for example, the *min-max* method, which uses the maximum value of the feature as one and minimum as 0 and scales values between them. (Aksoy and Haralick, 2001)

Transformation or *feature construction* is a method to discover missing information about the relationships between features and constructing new features from the feature set that would provide more accurate and concise classifiers, in addition to providing more comprehensibility. These features could be combinations of present and future values such as a_n+2. (Kotsiantis, Kanellopoulos and Pintelas, 2006; Rizoiu, Velcin and Lallich, 2013)

2.1.2 Missing data

Often data used to create an ML model includes missing values. Especially after setting the requirements for cleaning the data, one should decide what to do with the missing data points. A straightforward method is to delete the instance that has the missing data, which often leads to data loss, or the empty values can be filled with some estimated value. These values can be derived from similar cases, using mean values or statistical or machine-learning methods. (Zhu *et al.*, 2012)

2.1.3 Sampling

Often, especially in CCP cases, there exists a phenomenon called *class imbalance*. For example, in the framework of CCP, it means that in a dataset, a churning customer is a rare object. However, when building a model with this kind of imbalanced data it leads to problems such as improper evaluation metrics, lack of data (absolute rarity), relative lack of data (relative rarity), data fragmentation, inappropriate inductive bias and noise (Burez and Van den Poel, 2009), in addition to poor generalizability (Galar *et al.*, 2012).

To solve these problems, researches commonly use sampling, where the basic idea is to minimize the rarity by adjusting the distribution of the training set. Basic methods are called over-sampling and under-sampling. Over-sampling in a simple way means to duplicate the rare incidences while under-sampling eliminates the overrepresented classes. Both methods are suitable and decrease the imbalance, but they both are with drawbacks. Under-sampling removes the information and degrades classifier performance and over-sampling, in turn, can increase the time required to train the model as well as may lead to overfitting (Chawla *et al.*, 2002; Drummond and Holte, 2003).

2.1.4 Feature and variable selection

Feature and variable selection are the means of extracting as much information from multiple different variables as possible. As the number of variables and data has increased due to more advanced data gathering, it is essential to include only the most critical and useful variables for the model one is building. There are three main objectives in selection: achieving better predictive performance, getting faster and more efficient predictions, and getting a better and more precise understanding of the predictive process. Adding unnecessary variables to the model adds complexity or can introduce the model to overfitting, but missing essential variables leads to more reduced predictive performance (Guyon and Elisseeff, 2003). Feature selection has different categories that split up to *filter, wrapper,* and *embedded* methods (Chandrashekar and Sahin, 2014).

Filtering works by using decided feature relevance criteria. It could, for example, be the variance of the feature. By computing variance of each feature and defining a threshold variance with a more significant variance than the threshold is taken into the model. One other standard method is using a ranking method, which is based on the idea that essential features are relevant if they can be independent of the input data but are not independent of the class labels. “The feature that does not influence the class labels can be discarded.” Filter methods are simple but sometimes do not take into account the interdependence of the features, or with ranking methods, there is a possibility of getting a redundant subset. (Chandrashekar and Sahin, 2014)

The *wrapper* method uses algorithms to go through possible feature subsets and try to maximize the classification performance. Large feature sets can become computationally very heavy because the problem grows exponentially as features add up, which is also called an NP-hard problem. NP-hard means that it belongs to other class of commonly known computer science problems NP (nondeterministic polynomial (time)) problems, where given a solution, it can efficiently be verified to be correct, but it is unknown whether there is efficient algorithm to find the solution. The other problems in class P can be efficiently solved with an algorithm. There are optimized algorithms such as Genetic algorithms or particle swarm optimization, which are more complicated, but simpler ones are called sequential selection algorithms. The methods above iterate through the features and by adding the best classifier into the subset. (Chandrashekar and Sahin, 2014)

In *Embedded* methods, the main goal is to try to reduce the computational time taken by reclassifying different subsets and incorporating the feature selection into the training process. The simplest way to understand this method is to add a penalty variable to the model when it is adding more bias, i.e., more variables. (Chandrashekar and Sahin, 2014)

One more common method is to use principal component analysis (PCA), which is a linear extraction method that transforms the data into a low-dimensional subspace. The idea is to retain most of the information but reduce the features into a smaller vector. (Li, Wang and Chen, 2016)

2.1.5 Hyperparameter optimization

Many of the machine learning models have parameters that can be chosen before the training is initiated, such as the kernel function in support vector machines (SVM). These parameters are called hyperparameters, and they can be tweaked to achieve higher performance of a model with a chosen criterion such as accuracy or recall rate. Hyperparameter search can be done manually, following rules of thumb, or it can be automatized. Searching automatically has multiple benefits such as reproducibility and speed, in addition to outperforming the manual search. (Claesen and De Moor, 2015)

There are multiple ways of doing the hyperparameter optimization automatically such as grid search, random search, Bayesian optimization, gradient-based optimization, and others. Grid search is a well-known and straight forward method of doing the optimization. A systematic grid search goes through all parameters that have been inputted to it by changing only one at a time (Beyramysoltan, Rajkó and Abdollahi, 2013). Then the models are evaluated against a chosen criterion, and the best parameters are returned. Since grid search goes through all the possibilities, it can be computationally hard (Bergstra and Bengio, 2012), but often there are only a few parameters to go through (Claesen and De Moor, 2015). One other common way of doing the optimization is by using random search, which moves away from going through all the combinations of parameters and instead selects them randomly. Random search can outperform grid search, especially if there are only a few hyperparameters that affect the final performance (Bergstra and Bengio, 2012). However, since random search searches best variables only randomly it might not find the real best values.

2.2 Methods

As mentioned, supervised learning requires labels on the data that it is using to learn the features of the data and then uses the training to predict values for an unseen datapoint. The most used supervised classification methods for predicting customer churn are (Sahar F. Sabbeh, 2018):

- Logistic regression (LR)
- Decision tree (DT)
- Naïve Bayesian (NB)
- Support vector machine (SVM)
- K-nearest neighbor (KNN)
- Ensemble learning: Ada Boost (AB), Stochastic gradient boost (SGB), Random forest (RF)
- Artificial neural network (ANN)

Sahar includes linear discriminant analysis, but the source literature is more focused on using the methods above.

2.2.1 Logistic regression

LR belongs to a group of regression analysis techniques, which are primarily used to investigate and estimate relationships among features in the dataset. When the dependent variable, i.e., the variable tried to be forecasted, is binary, LR is appropriate (Sahar F. Sabbeh, 2018).In LR models, the relationship between the dependent variable and the given feature set, and it can be used with discrete, continuous, or categorical explanatory variables. The model is favored by many since it's straightforward to implement and interpret, in addition to being robust (Buckinx and Van Den Poel, 2005; Hanssens *et al.*, 2006; Neslin *et al.*, 2006).

What regression methods are trying to do is to fit a curve between data points in sets. A similar *linear regression* uses the least-squares method to measure the error or the distance between the data point and the line. In logistic regression, maximum likelihood is used. Maximum likelihood (Figure 2) works by trying to maximize the probability of obtaining the observed set of data by using likelihood function. The maximum likelihood estimators are chosen to be those that maximize the likelihood function and agree most with the data. Logistic regression can be represented, as shown in equation 1.

$$logit(\pi(x)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \quad \Leftrightarrow \quad \pi(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \quad (1)$$

Where π(x) is the probability of predicted event and β_i regression coefficients for each explanatory variable x_i . Solving π(x) from the equation gives the probability of belonging to the predicted class. (Hosmer, W. and Lemeshow, 2000, 7-12).

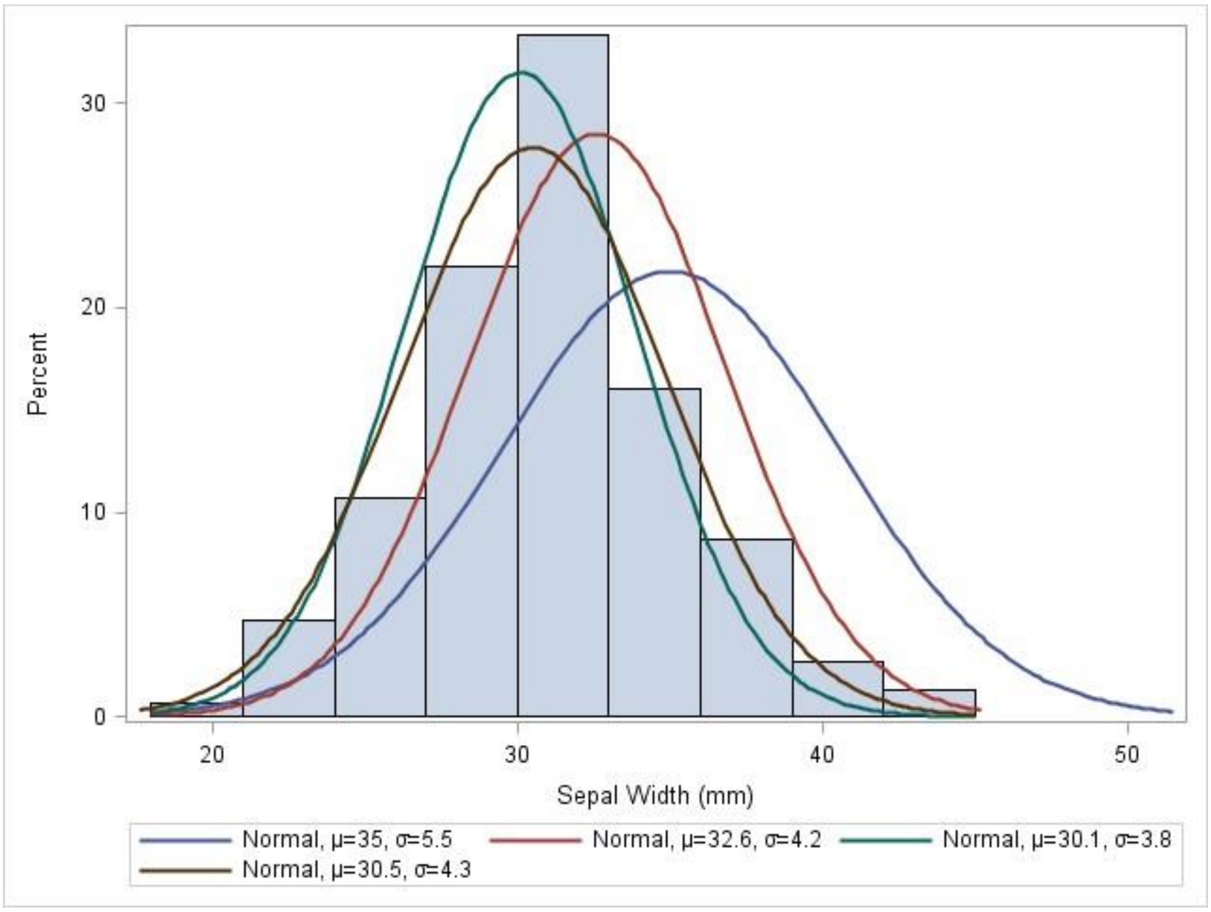


Figure 2 Maximum likelihood (Wicklin, 2011)

2.2.2 Decision trees

DTs are simple, very popular (Sahar F. Sabbeh, 2018), fast to train, and easy to interpret models that use comparison or if-then-else method of learning features from the data. They can be applied to both categorical and continuous data, and they are reasonably competent in their predictions but are prone to overfitting. Their efficiency can be enhanced with boosting (Mohri, Rostamizadeh and Talwalkar, 2018). In Figure 3, we can see a simple *binary decision tree*. DTs are divided into classification and regression trees, depending on the outcome.

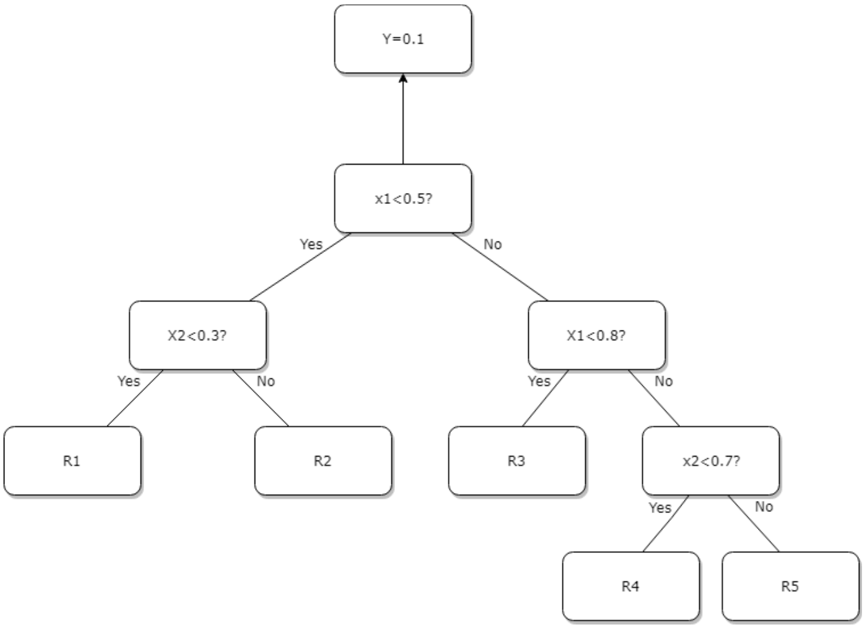


Figure 3 Simple decision tree based on binary variable Y (Song and Lu, 2015)

Nodes are split into the *root*, *chance*, and *leaf* nodes. The *root* node is a choice that will split records into two or more nodes. *Chance* nodes represent the choices available at that point in the tree, and *leaf* nodes are the results. Branches are between the nodes and represent classification rules that can be described with if-then. Splitting means to split parent nodes into purer child nodes, which continues until stopping criteria is met. Defining stopping criteria is vital since the too complicated model would be overfitted and would not predict the future that well nor be generalizable. Stopping criteria could be a minimum number of records in a leaf or records in a node before splitting, and the depth of the tree. Pruning means building large trees and removing less informational nodes. (Song and Lu, 2015)

There are different models of DTs, which are called CART, C4.5, CHAID, QUEST, and more (Song and Lu, 2015) of which CART (Classification and regression trees) is mostly used in studies that were considered in this study.

2.2.3 Ensemble methods

Ensemble methods are algorithms that create a set of classifiers which they use to classify new data points by using weighted voting (Dietterich, 2000). Using multiple base-classifiers results in better performance compared to using a single one (Verbeke *et al.*, 2012). The ones in the interest of this study are random forests (RF), bootstrap aggregating (bagging), and boosting. They are both methods that create multiple classifiers, or weak learners, from the instance. Bagging takes random values from the original dataset, even the same ones, and creates multiple learners. Boosting takes this idea further and creates weights for data points according to the error rates, in order that the wrong predicted values are more presented in the next weak learner (Quinlan, 2006).

RF is a tree-based method but belongs to the ensemble learning category. RFs work by generating collections of DTs, which get their subset of observations, and each split in trees is based on a most discriminative threshold on the random variable subset. Forests generate predictions by an average of predictions from individual trees. (Fang, Jiang and Song, 2016) RFs often use CART as a base learner (Verbeke *et al.*, 2012) and are a form of bagging (Rodríguez, Kuncheva and Alonso, 2006).

Extremely randomized trees or extra trees (ET) are similar to RF in a sense that it also takes a random subset of candidates, but instead of picking the next split by looking for a discriminative threshold, the thresholds are randomly drawn. This allows for lower variance but can increase bias. In addition, ETs are computationally faster to create. (Geurts, Ernst and Wehenkel, 2006)

Both bagging and boosting can use decision trees, for example, as base learners. They both work by creating *weak learners*, for example, decision stumps, a one-level decision trees, to an ensemble structure from which the structures will vote for the end prediction. Bagging works by repeatedly choosing samples (bags) from a data set according to a uniform probability distribution and trains the base classifiers on the resulting data samples. This means that there can be more than one instance of the same data point. Boosting continues with the same logic, but the classifier is trained on data, which has been hard for the previous classifier. This means that the base classifier will focus more on harder to classify problems, and weights are added for classifiers according to the difficulty of the training set. Voting for the results is done by using majority voting. (Rodríguez, Kuncheva and Alonso, 2006; Verbeke *et al.*, 2012) Hence we can see that RF is a bagging based method of ensemble learning (Sahar F. Sabbeh, 2018).

2.2.4 Naïve Bayesian

Naïve Bayesian (NB), based on Bayes' theorem, is a supervised classification method that belongs to the Bayesian category of machine learning. Bayesian algorithms estimate the probability for a future event based on previous events and follow the idea of variable independence. This means that the presence or absence of other features is unrelated to the presence or absence of another feature and that variables independently contribute classification of an instance. Instead of just classifying outcomes, NB predicts the probability of the prediction to belong to specific categories. (Sahar F. Sabbeh, 2018)

2.2.5 Support vector machine

Support vector machines (SVM) are a very effective supervised classification technique (Verbeke *et al.*, 2011; Mohri, Rostamizadeh and Talwalkar, 2018; Sahar F. Sabbeh, 2018) which tries to model patterns in the data, even non-linearities. SVM was first introduced by Cortes and Vapnik (1995). SVM works by representing observations in a high dimensional space by constructing an N-dimensional hyperplane that isolates data points into two categories. The goal is to find a hyperplane that optimally divides the data points in a way that one category is on the one side of the hyperplane and the other on the other side. (Kumar and

Ravi, 2008) The boundary between classes is mapped via a kernel function, which is applied to each data instance that is then mapped into higher dimensional feature space, as we can see from Figure 4 (Coussement and Van den Poel, 2008). A kernel is essentially a way to compute the dot product of vectors x and y. Since the kernel has a great impact on the generalization performance of SVM, multiple kernel SVM's with better predicting performance have been suggested (Chen, Fan, and Sun, 2012).

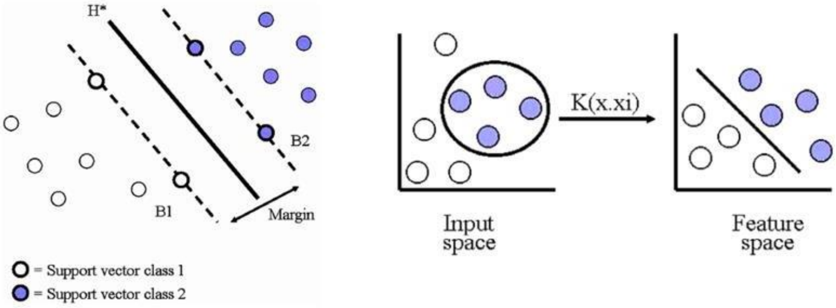


Figure 4 SVM in binary linearly(left) and non-linearly(right)

2.2.6 K-nearest neighbor

KNN belongs to a category called instance-based learning or memory-based learning, where new instances get labeled based on previous instances, stored in memory. KNN is most widely used in this category of methods. (Sahar F. Sabbeh, 2018) KNN is also non-parametric, which means that it does not make assumptions over data and is hence more applicable for real-world problems. It is also called a lazy algorithm, which means all of the data points are used in the test phase (Keramati *et al.*, 2014).

KNN works by using the distances between data points to classify records. Distance is measured using by using multidimensional vectors in feature space. Euclidean distance, meaning the length of a straight line between two points (Tripathy, Ghosh and Panda, 2012), is often used for measuring in KNN. Besides, other distance measures, such as Manhattan, Murkowski, and hamming, distances are used. When classifying objects, its feature vector is compared to the training data, and the class closest to it is its class. The “K” comes from the number of training instances that are closest to the new point. (Keramati *et al.*, 2014; Sahar F. Sabbeh, 2018)

2.2.7 Artificial neural networks

Inspired by biological nervous systems, ANN uses interconnected neurons to solve problems. ANN is comprised of layered nodes and weighted connections between them. It takes multiple input values and makes a single output. Both the weights and the arrangement of nodes have an impact on the result. The training phase is used to adjust the weights of the connections to achieve wanted predictions. ANNs can be used for complex problems and have tremendous predictive performance. There are different variations of ANNs, which are called Feed-Forward (FFNN) and recurrent neural networks (RNN). FFNN is similar to what is seen in Figure 5, which means input, hidden, and an output layer with unidirectional arrows. The difference is that RNNs have backward connections. (Mohammadi, Tavakkoli-Moghaddam and Mohammadi, 2013; Keramati *et al.*, 2014; Sahar F. Sabbeh, 2018)

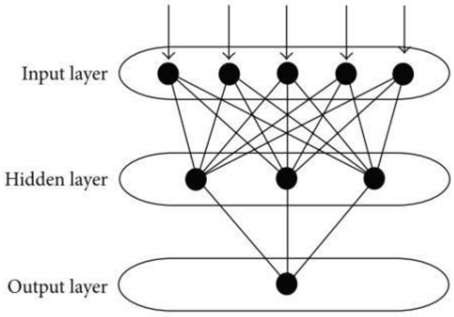


Figure 5 Example of an ANN (Mohammadi, Tavakkoli-Moghaddam and Mohammadi, 2013)

2.3 Model evaluation

Evaluation of the model is essential since that is the way to compare models. Models need to be accurate and generalizable, which means models are not overfitted to a specific dataset. This section of the study will consider metrics that are used to measure the accuracy of the models.

2.3.1 Validation

The process of quantitively verifying that the results between input variables and results are acceptable descriptions of the data is called validation. One way of error estimation is an evaluation of residuals, which means to measure the error between predicted and actual value called training error. However, this does not consider the possibility of over- or underfitting. To measure the generalizability, we can use cross-validation, which includes such techniques as the holdout method and the k-fold cross-validation.

The holdout method or 2-fold cross-validation means to split the data into training and test sets with often a ratio of 2/3 for training. As is suggested by their names, training data is used to train the model, and test data is used to test the model's predictions after training. The variable to adjust is the relation between training and testing data. More substantial testing data usually means more bias towards the training data, but too small testing data size can lead to more significant confidence intervals for testing accuracy. (Kohavi, 1995)

K-fold cross-validation, or rotation estimation, means that the dataset is randomly split into k- mutually exclusive subsets (folds) that are of equal size. Then the method is trained and tested k times with different sets. The accuracy estimate is the number of correct classifications divided by the number of instances in the dataset. (Kohavi, 1995; Mohri, Rostamizadeh and Talwalkar, 2018) An example of 5-fold cross-validation can be seen from Table 1.

Table 1 5-fold cross-validation

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

Test data	Training data

2.3.2 Confusion matrix

A confusion matrix (CF) is a popular evaluation metric in terms of classification problems. It can be used to test the reliability of the classification method. To illustrate the idea, we can think of the classification problem as a binary problem where the instance either is classified correctly or is not. Hence, there are four possibilities for the instance to end up:

- True Positives (TP): predicted positive, true value positive
- False Positives (FP): predicted positive, true value negative
- False Negatives (FN): predicted negative, true value positive
- True Negatives (TN): predicted negative, true value negative

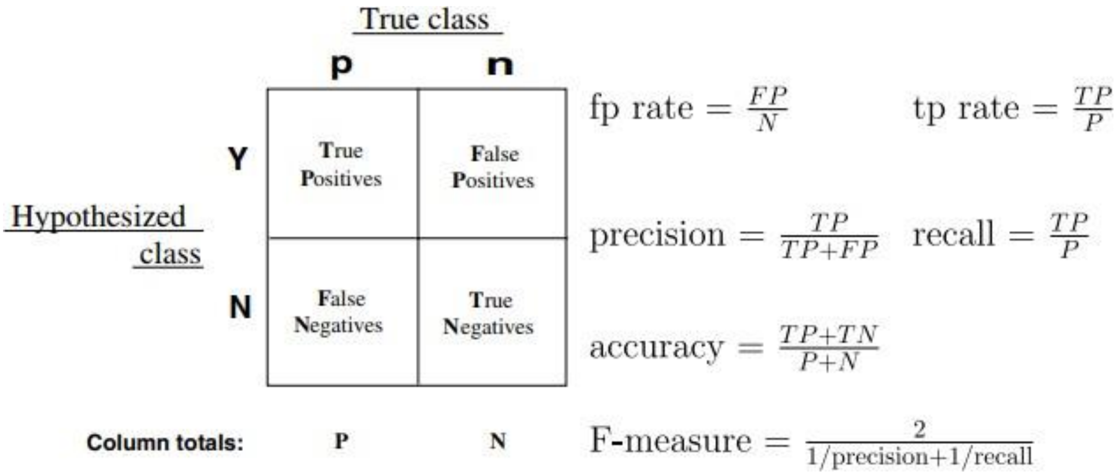


Figure 6 Confusion matrix and performance metrics (Fawcett, 2006)

Figure 6 shows an example of a confusion matrix and performance metric that can be calculated from it. *True positive rate, hit rate, or recall* (*sensitivity*) can be calculated by dividing the number of positives correctly classified by the total amount of positives. Similarly, the *false positive rate* (FP rate) or *false alarm rate* is calculated by dividing the number of false positives with the total amount of negative values. Additionally, there are terms such as *precision* and *specificity* from which the first measures the accuracy of correct positive values (true positive of total positives) and the latter the same but for negative values (true negatives of total negatives). (Fawcett, 2006) *Accuracy* is often used as a useful base metric for models since it describes the total amount of correctly classified predictions. However, previous scores do not necessarily mean satisfactory performance if, for example, data is severely imbalanced good accuracy can be achieved just by predicting the bigger proportioned class. Furthermore, good scores in precision or recall do not necessarily mean that the classifier is right on the other metric. Hence *F-measure* is introduced, which is an excellent single metric that combines precision and recall in a harmonious way. Values closer to one imply excellent performancein both precision and recall. (Vafeiadis *et al.*, 2015) F-measure can also be tweaked in favor of precision or recall by introducing a β variable (Equation 2). The harmonic version can be though as F_1 and $F_{0.5}$ favours precision more than recall and F_2 recall more than precision.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \qquad (2)$$

Measurements in the confusion matrix (CF) can be used to calculate the misclassification cost, which is wanted to be minimized. It can be calculated as follows:

$$\text{Cost} = \text{FP x CFP} + \text{FN x CFN} \qquad (3)$$

CFP is the cost of a false positive, and CFN is the cost of a false negative. The cost functions can be calculated case by case, but in general, it is some general cost associated with the model predicting wrong results. Minimizing Cost as a measurement makes more sense, compared to just minimizing the probability of error, since it can be adjusted which one, FP or FN, is more detrimental. However, often, the costs are not known. (Bradley, 1997)

2.3.3 Receiver operating characteristic curve

The receiver operating characteristic curve (ROC) builds on top of the confusion matrix and plots the TP rate on Y and FP rate on X-axis as discrete points. The ROC shows the relationship between TP and FP or in other words, benefits, and costs. An example can be seen in Figure 7. The curve starts from 0,0, where there are no correct classifications, but there are no false positives either or end in 1,1, where the model always predicts a positive classification result. Coordinates 0,1 represent a perfect model. (Fawcett, 2006) However, just having discrete points does not show the performance when decision thresholds are varied, and only graphical representation can be seen. A better metric is called area under the ROC curve (AUC), which comprises the area under the curve into a single number, which is easier to interpret and make comparisons. AUC is more sensitive (Bradley, 1997) and better measurement (Huang and Ling, 2005) than *accuracy*.

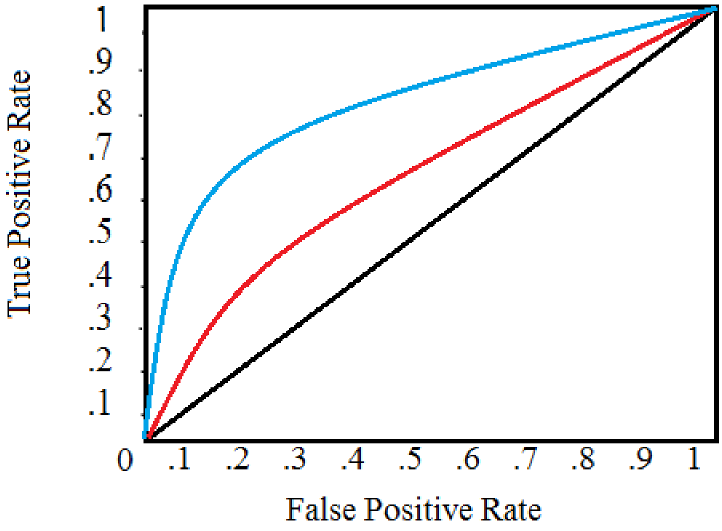


Figure 7 A ROC curve (Glen, 2016)

2.3.4 Top-Decile Lift

Top-Decile lift (TDL) focuses on the most certainly classified data points. For example, in the case of this study, the proportion of people that are most likely to be churned divided by the proportion of churners in the whole dataset. The higher the TDL is, the better the classifier is, since the higher TDL means that there are more actual churners in the segment of churners. (Lemmens and Croux, 2006) TDL is an excellent assessment criterion because it focuses on managerial value by focusing on customers that are most likely to leave the company. It is also prevalent in CCP (Coussement, Lessmann and Verstraeten, 2017) as also the literature review in this thesis shows.

2.3.5 Mean squared error

All previous evaluation methods would work with classification, i.e., discrete numbers. However, when probabilities or continuous values are used, other methods are required. Mean squared error (MSE) provides a way to evaluate the predictive performance of a model. The MSE is calculated as:

$$\mathrm{MSE}(\mathrm{y}, \hat{\mathrm{y}}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2 \quad (4)$$

3 Literature review on customer churn prediction

This chapter reviews the present literature of CCP. The first two parts clarify the methodology used in this study, and how equivalent research can be conducted. Subsequent chapters present the literature on different models, data preprocessing, and model evaluation.

3.1 Methodology

The gathering of studies that are relevant for the purposes of this study was carried out according to suggestions of Webster & Watson, 2002.

1. Search leading journals but also look outside the primary discipline.
2. *Go backward,* reviewing citations of the articles in step 1 to find prior contributors.
3. *Go forward,* by using the Web of Science (the electronic version of the Social Sciences Citation Index) to identify the critical articles. Then determine which of these should be included in the review.

As for the structure, Webster & Watson, 2002 suggest using a concept-centric approach compared to author-centric since it allows better synthetization of the literature. The difference between these two is captured well in Table 2. This study first uses an author-centric approach to show overview from different studies and, after that, gathers concepts to a concept matrix.

Concept-centric	Author-centric
Concept X … [Author A, Author B, …] Concept Y… [Author A, author C, …]	Author A … concept X, concept Y, … Author B … concept Y, concept W, …

Table 2 approaches to literature review (Webster and Watson, 2002)

3.2 Literature search process

The searches were done using Finna-service that searches many different databases or portals in conjunction with Ex Libris, arranges results with relevancy, and has an option to search only peer-reviewed studies. The articles, from leading journals, were found mostly from "Scopus (Elsevier)," "ABI/INFORM Global," "Science Citation Index Expanded (Web Of Science)," "ScienceDirect Journals (Elsevier)," "Computer And Information Systems Abstracts" an "SpringerLink "portals. A search to the databases with the string "predicting" AND "Customer churn" AND "insurance" provided 240 peer-reviewed studies that were good enough precision to start scanning the literature on the surface.

The exclusion of studies was done with different metrics. These metrics were a different type of data (behavioral, transactional, etc.), studies of business results or marketing, studies that were not available, or studies that were more concerned with the data mining/ data gathering aspect. Also, some studies that were of similar subjects, according to their abstracts, were discarded. This resulted in 20 studies that were *backward* tracked to articles that were common between different articles, which provided 30 articles in total. After further research, concepts were starting to look familiar, which concluded the search. These articles provided a clear review of the literature now, and the overview can be seen below. For article management, a program called Mendeley was used, which allowed sorting and automating the addition of articles.

3.3 Customer churn prediction

There is much research made in the field of CCP, much of it quite recently, and focusing on the telecommunication industry and some on the insurance industry. The research can, in the context of CCP, be split from the data structure to customer informational data and customer behavioral data. Behavioral data is data collected from the behaviors of the customer, for example, where customers drive daily and how much has the customer just unsubscribed from the newsletter.

Customer informational data is geographical such as gender, income, place of residence. Because the data in this study is in the form of stationary customer data, studies done on behavioral data are not considered within the scope of this review. Behavioral data makes sense in many CCP scenarios where customer behavior is actively followed, and much data is available on it, as well as in cases where customer can end the contract very fast. Hence it makes sense for the company to be involved as fast as possible. This is the case often in the telecommunication industry. Table 3 shows different studies, that are considered in this review, to get a representative overview.

Table 3 Authors & year, model, algorithms and data

ARTICLE & YEAR	PURPOSE	DATA
Coussement & Van Den Poel, 2008	Classification of churners and comparison	Newspaper marketing dataset
Sharma & Kumar Panigrahi, 2011	Classification	Telecom operator customer data with voice calls
Cerbeke, Martens, Mues, & Baesens, 2011	Classification	Telecom operator customer data.
de Bock & Van Den Poel, 2012	Classification of churners and comparison	Multiple different datasets from different industries
Ballings & Van Den Poel, 2012	Effect of data time period	Newspaper customer data
Mohammadi, Tavakkoli-Moghaddam, & Mohammadi, 2013	Classification	Telecom operator customer dataset
Günther, Tvete, Aas, Sandnes, & Borgan, 2014	Predicting the risk of leaving	Insurance company customer data

Farquad, Ravi and Raju, 2014	Predicting the risk of leaving	Chinese credit card company customer dataset
Keramati *et al.*, 2014	Comparing data mining techniques in CCP	
Vafeiadis, Diamantaras,2015	Comparison of techniques used in churn Prediction	Telecom operator customer data, Monte Carlo simulation
li, Wang, & Chen, 2016	Feature extraction	Telecom operator customer data
Tamaddoni, stakhovych and ewing, 2016	Comparison of techniques used in churn Prediction	Transactional records of two firms
Ahmed and Linen, 2017	Review of CPP methods	Telecommunication operat or customer data
Coussement, Lessmann and Verstraeten, 2017	Data preparation	European telecommunication provider customer dataset
Faris, 2018	Classification using optimization technique for Inputs	Telecom operator customer data
de Caigny, Coussement, & De Bock, 2018	Classification	Financial services, Retail, Telecom, Newspaper, Energy, DIY
Sivasankar and Vijaya, 2018	Classification	Three datasets from Tera data center, Duke University

Mau, Pletikosa and Wagner, 2018	Likelihood of future customer and churn Probability	Insurance company’s customer data

CCP, from a machine learning perspective, is a classifying problem. Hence, we try to predict "0" if the customer is not churning and "1" if the customers are churning. Therefore, literature is focused on models that are used for classification such as SVM, LR, DT, and RF. Prediction accuracy is the most researched point of evaluation when it comes to CCP. According to (Ahmed and Linen, 2017), the prediction accuracy can be enhanced in the literature by enhancing the methods or through better pre-processing and feature selection. In addition, one shouldn’t just focus on predicting churning accuracy (Verbeke *et al.*, 2011; De Bock and Van Den Poel, 2012; De Caigny, Coussement and De Bock, 2018) but the model should also be comprehensible, meaning that it should also provide reasons for the churning so that experts can validate its results and check that it predicts intuitively correctly. Comprehensible models would allow the company to know what is driving the churn and how they can improve customer satisfaction to increase retention (Buckinx and Van Den Poel, 2005). The next chapter will introduce studies from the CCP field, their models, methods, and results.

3.3.1 Review on the customer churn modeling field

The research on CCP has started by implementing single classifier models and trying to improve predictive performance and having the interpretability as a secondary objective. When it comes to a single model’s predictive classification performance, SVMs seem to have high predictive performance as they can model non-linear relationships. Verbeke *et al.* (2011) used Ant-Miner+ and ALBA methods to not only achieve better accuracy but also to achieve better comprehensibility. Ant-Miner is based on ant colony optimization, and ALBA is based on non- linear SVM. The results show that both ALBA and Ant-Miner achieved better performance compared to traditional models, in addition to achieving comprehensibility. However, Coussement and Van den Poel (2008) compared SVM's with two different parameter-selection techniques, based on grid search and cross-validation, and compared them to LR and RF. They found out that SVM's outperformed LR only if parameter selection was successful, but RF was always found to be more accurate. Another modern single classifier consideration is ANN- based models. Sharma, Panigrahi and Kumar (2011) suggested the ANN-based approach and were able to achieve high accuracy. Sahar F. Sabbeh (2018) did a review from current ML methods used in the field and ranked them according to their accuracy. She used behavioral data for her predictions and found out that RFs had the best accuracy, followed by AdaBoost, SGB, and SVM. NB and LR were found at the bottom of the models.

Another essential factor to consider regarding model picking is the data preparation phase and boosting. A study comparing data preparation algorithms and their effects on LR's performance against more state-of-the-art techniques such as Bayesian network, DT, ANN, NB, RF, and others found out that when data preparation was done well, LR was able to perform on-par with the advanced techniques. The authors also implied that implementing LR is less cumbersome, and data preparation is nevertheless required to be done for more advanced classifiers (Coussement, Lessmann and Verstraeten, 2017). Regarding boosting, a study comparing the classification performance of SVM, LR, and DT models, found that with adaptive boosting, DTs had the best predictive performance among them (Tamaddoni, Stakhovych and Ewing, 2016). However, the differences between the precision scores of the methods above were not very significant. Vafeiadis *et al.* (2015) compared SVM, LR, ANN, DT, and NB with and without boosting. LR and NB could not be boosted since they lack free parameters to be boosted. Without boosting ANN with back propagation was found to be the most accurate and NB and LR the least accurate. However, with boosting SVM was the most accurate according to accuracy and F-measure.

A recent trend has, however, been that not only a single classifier is used but multiple, to enhance the accuracy or interpretability. A review on multiple CCP studies, including models such as LR, SVM, ANN, DT, and RF, found that recent studies are often able to reach high accuracy with single method models. However, the best accuracy is obtained by using hybrid models (Ahmed and Linen, 2017). Sivasankar and Vijaya (2018) implemented a hybrid method that clustered the data first and then used ANN to make predictions on the data. They were able to achieve high accuracy. Mohammadi, Tavakkoli-Moghaddam and Mohammadi (2013) suggested the use of hybrid ANN models called hierarchical models. They are comprised of clustering, classification and survival analysis to make more accurate predictions whilst getting outputs of the reasons behind the predictions. They found out that a combination of Alpha-Cut Fuzzy C-Means Clustering, ANN, and Cox was the best combination for their dataset, and they were able to achieve very high accuracy. Keramati *et al.* (2014) also suggested the use of hybrid methods and compared its performance against DT, ANN, KNN, and SVM. The hybrid model they used was to get predictions on all other models and make predictions by calculating the average score and making the prediction accordingly. They found out that from the fore mentioned models, ANN performed the best in terms of prediction accuracy, but the hybrid model achieved the best results. De Caigny and Coussement and De Bock (2018) benchmarked the logit leaf model (LLM) against DT, LR, RF, and Logistic model tree (LMT). LLM uses a hybrid approach that creates decision trees to classify segments in the first step and applies logistic regression to each segment. This means that LLM has a built-in feature selection and can select the most important variables for each group separately. The study found that by combining the LR and DT, it was able to achieve better prediction accuracy compared to other methods.

Another way to leverage hybrid models is to use them to improve on the model’s interpretability. According to Farquad, Ravi and Raju (2014), SVM is a state of the art classification model, but its drawback is that it is the so-called “black box”-

model and does not reveal knowledge outside. Hence, it is not comprehensible by humans. In their research, they used a hybrid approach that first used SVM-recursive feature elimination to reduce features. Then the SVM model is created, and support vectors are extracted, and rules are generated using the Naïve Bayes tree. The researchers were able to outperform the SVM without feature selection and improved the comprehensibility of the model. De Bock and Van Den Poel (2012) were also interested in comprehensibility or interpretability and suggested an extension to generalized additive models (GAMs) called GAMensPlus, that combined training and prediction phases of GAMens with explanation phase. They compared classification performance against ordinary ensemble classifiers such as bagging and RF and LR and normal GAMs. GAMensPlus came on top in AUC, TDL, and lift.

Faris (2018) points out, an essential issue in the CCP, imbalanced data distribution, which means that non-churners are often much more common than churners in datasets, and the issue could lead to lousy generalizability of the model. The ways to tackle are divided into three categories: algorithm level approach, data level approach, an ensemble approach. An example of an algorithm level approach is to try modifying models to give more weight to the rare churn instances. The data level approach means to use oversampling or undersampling to modify the distribution of the data. Ensemble approach means to combine decisions from multiple classifiers to achieve higher accuracy examples of these methods include RF, Boosting, and bagging. The author (Faris, 2018) ended up solving the problem by processing the data first with an oversampling algorithm, then running a loop between optimization algorithm to optimize the weights and feeding the results to a random weight network.

3.3.2 Customer churn prediction in insurance

Two studies were found that were looking into customer churn in the insurance business. While Günther *et al.* (2014) focused only on customer churn from the point of an insurance company, Mau, Pletikosa and Wagner (2018) had extended their research to predict customer retention and cross-selling opportunities as well.

Günther *et al.*, 2014 suggested that logit-models, logistic regression models, seem to be the most popular among churn prediction studies since they are simple, show excellent performance, and are interpretable. However, a linear relationship between explanatory variables and the logit is assumed, which leads to loss of information when the relationship is not linear. Therefore, they present a logit model that can capture non-linear relationships. They achieve this by using GAMs.

Mau, Pletikosa and Wagner (2018) take a more bottom-up approach and start the model development from enriching the data. They imply that companies are struggling to select relevant data. To resolve the issue, they suggest using, in addition to traditional personal data, data about customer participation such as inquiries from the company's website to improve CCP performance. With the enriched data and using RF as the classifier, authors were able to improve significantly on their model accuracy.

3.4 Summary

In this review, answers for the first research question, "*What is the current state of customer churn prediction in the literature?* "and its sub-questions were found. For the first sub-question, "*What algorithms are used in customer churn prediction, and how are they evaluated?*" The results of the literature review have been summarized in Table 4. From the table, we can see that the most used models are: support vector machines (SVM), Logistic regression (LR), Artificial neural networks (ANN), decision trees (DT), and random forests (RF). There are variations of these models, but they are categorically within these models. In addition, most used validation methods seem to be the area under the curve (AUC), receiver operating characteristics curve (ROC), percentage correctly classified/ accuracy (PCC), and top decile lift (TDL). However, the confusion matrix (CM) and training and evaluation (T/E) are also widely used.

The second sub-question "*What is the current state of customer churn prediction literature on the insurance field?*" is answered under title 3.3.2. There was not much directly relevant research done under this question, but two different studies from the insurance field arenreviewed, and their models explained. Both studies agreed that CCP has its place and rationale in the insurance business since retaining a customer is cheaper and more profitable, contact occurs infrequently, and insurance is seen as a necessary evil. From these studies, the other one was using GAM and the other one RF. Both models are used in other industries as well.

4 Developing machine learning model to predict future churning customers – A case study

This chapter presents the steps of how the application of machine learning methodologies was conducted in this study. Assumptions and data cleaning procedures are also explained in this chapter. Figure 8 represents an overview of the process.

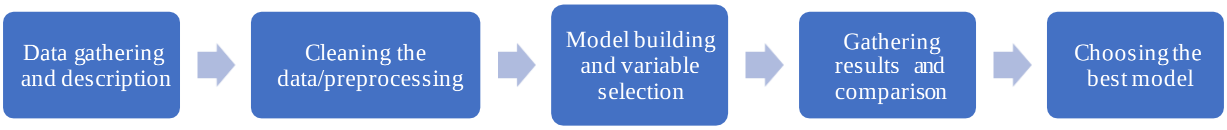

Data gathering and description
Cleaning the data/preprocessing
Model building and variable selection
Gathering results and comparison
Choosing the best model

Figure 8 Model building process

4.1 Tools and libraries

The empirical part of this study is done using the Python programming language and libraries that have been developed for it. Python is a high-level open-source language (Python Software Foundation, 2019), which has become one of the most used (Pedregosa *et al.*, 2011) if not the most used (Elliott, 2019) languages in machine learning and scientific computing. It has multiple open source developed libraries from which a few are used in this study. The libraries are called *numpy*, *scipy*, *pandas*, *matplotlib*, *sci-kit-learn*, *TensorFlow*, *Keras*, and *seaborn*. These libraries provide tools for preprocessing, algorithms for ML, and ways to plot the data.

4.2 Data description and considerations

The data was provided by a Finnish insurance company that wanted to predict whether a customer is going to stay or leave after the current period or not. Leaving or churning is defined as “1” and not leaving is defined as “0” in this dataset. In the dataset, there are almost 350 000 individual customer data points from which there are a maximum of three points in time from periods 2016-2018 if the customer has stayed during that period. Because the data gathering is already completed, this study is not concerned about the data gathering aspect.

The predicted variable, churning of the customer, has four different types in this dataset. Three different variables describe the churning of the customer from specific insurance. The different insurances are traffic insurance, full comprehensive traffic insurance, and personal insurance. Also, there is one variable that describes whether a customer has left altogether. The model proposed in this study tries to predict the churn of the customer altogether.

The dataset provides multiple predictor variables, e.g., features that can be used to make the prediction. The features can be divided into four groups: traffic insurance, full comprehensive traffic insurance, and personal insurance-related and general customer data such as age, area of residence, and gender.

As mentioned in the introduction, because the data has three points in time for the same customer, it is a longitudinal type of data. This would make it possible to take the time aspect into account in predictions by introducing lagging variables, for example. For simplicity and request from the insurance company, this study only considers the information from a single year, which means that every row of data is handled as an individual data point. Nevertheless, adding all the 806 000 datapoints would lead to having the same customer appear at most three times and by doing so, skewing the model. Hence it was decided to add all the churned customers and only the latest data point from the customers that have stayed. The result is then 350 000 individual datapoints.

4.3 Data preprocessing and feature selection

The data was collected initially from SAS enterprise guide software and was then converted to a CSV file. Some processing and data cleaning has been done on the data before giving it out in the insurance company, which means that the data is of high quality. However, for the data to be used for ML and statistical models, the data must be fitted to a specific format. The data included categorical string values, missing data points, and the two classes were imbalanced. Data imbalance is caused by the fact that more customers are staying than churning.

4.3.1 Categorical values

As said, the data includes nine categorical values that were both string values and date values. *The sales channel, language, region, gender, quality of the business relationship, time since the last move, time since last purchase, days from last accident,* and *duration of the customer relationship*. Date values were the starting date of customer relationship and last date of incident. From these dates calculated columns were made that were "days as a customer" and "days from the incident," which were calculated from the timestamp when the row had been recorded. Both date groups were labeled with years from "new," "1y" to "10y+". The distribution can be seen from Figure 9, where it can be seen that the long-time customers are overrepresented in the dataset but in turn, can add valuable information.

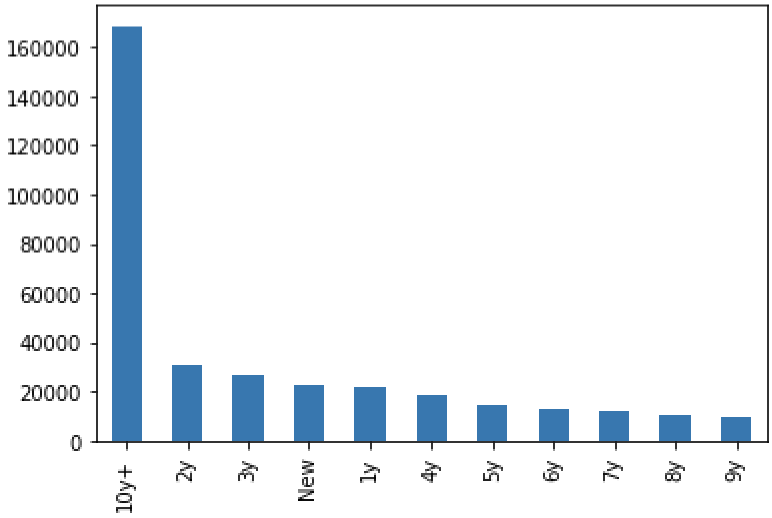


Figure 9 Distribution of the duration of customer relationship

After the creation of values from the date variables, there were multiple options to be done to change categorical values to numeric. One option would be labeling, e.g., changing categories to numbers so that one region gets a corresponding number. However, this could result in models ranking regions with higher numbers as better. Another simple method would be to add dummy variables for every corresponding category, but that would add additional features. One more way would be to binarize the values which would not increase the number of columns that much but would be harder to interpret. After testing, it was decided to create dummy variables because of simplicity, better interpretation, and because the number of different columns was not that high.

4.3.2 Handling missing data

The dataset had missing data in multiple fields such as region, sales channel, gender (Figure 11), language (Figure 10), and quality of the business relationship(Figure 12). After looking at the distributions, it was decided to replace the missing variables with extra variables that would indicate not having that information. The information about the missing variable could add information to the model. However, the number of missing variables was not that noticeable in many features. The only variable where the distribution was a significant factor was in the business relationship variable as the blue bar.

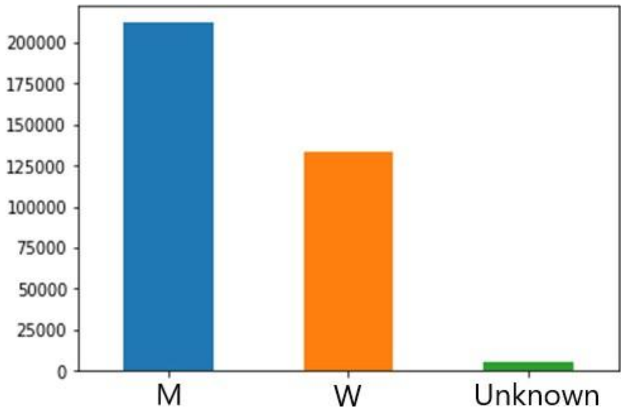


Figure 11 Gender distribution

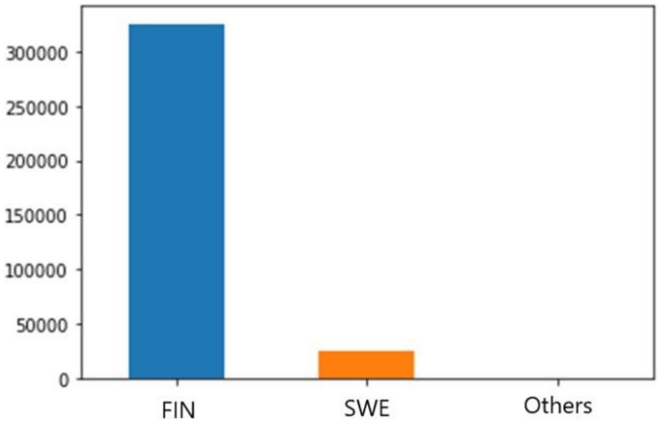


Figure 10 Language distribution

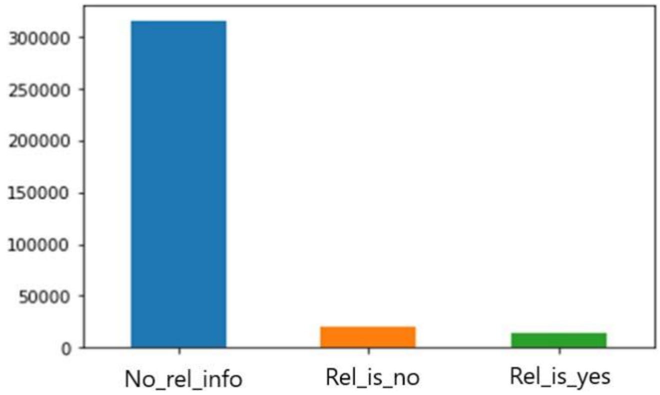


Figure 12 Distribution of quality of business relationship

4.3.3 Data normalization

To avoid the bias towards features in the dataset, the features needed to be normalized. Especially the calculated “days from” values since many data points had significant numbers in that column. All the other variables, except categorical values, were normalized. In normalization, a standard scaler that was provided by Scikit-learn was used.

$$z = (x - u) / s \qquad (5)$$

Standard scaler normalizes the features by removing the mean and scaling to unit variance (Equation 5). “u” is the mean and “s” is the standard deviation of the training set.

4.3.4 Feature Selection

Two different ways of feature selection were implemented. One was simply by using variables that were found significant by the insurance company that provided the data, and another was by using a feature selection algorithm. Feature selection with an algorithm was made by using ETs since their performance is very similar to RFs but is computationally much faster (Geurts, Ernst and Wehenkel, 2006). The number of features selected by the algorithm was 42, with the threshold for the selection being mean importance or above. The number of significant variables from the insurance company was 20.

The top five features selected by the algorithm can be seen in Figure 13. The features are ranked by the importance, which does not get very high with the given dataset. The features selected were very similar, but with an algorithm, the threshold to choose the variable could also be tweaked, which resulted either in having more or fewer variables compared to the insurance company. In the case of this study, the default threshold value, importance of higher or equal to the feature importance mean, was used.While more variables can be useful to have in terms of accuracy, it can also unnecessarily increase the complexity of the model or decrease the generalizability of the model by getting the model overfitted to the data.

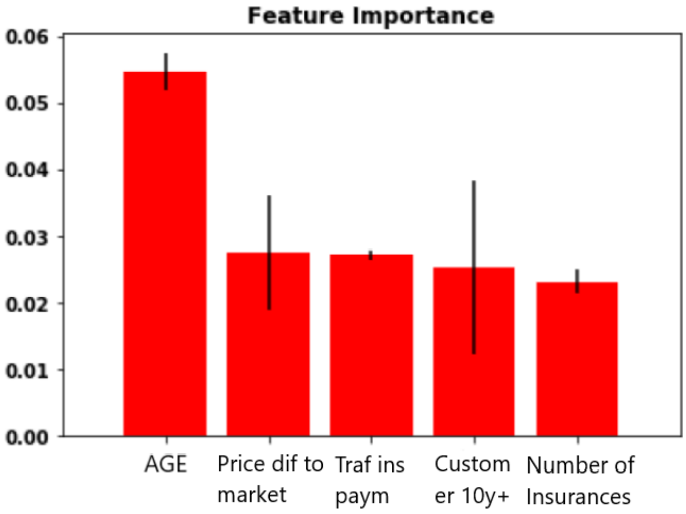


Figure 13 Importance of features selected by the algorithm

4.3.5 Imbalanced data

As with many ML problems and especially with CCP, the imbalance of the classes is an issue (Burez and Van den Poel, 2009; Farquad, Ravi and Raju, 2014; Faris, 2018; Amin *et al.*, 2019) as there is often more data about customers that have stayed compared to customers that have left. In the current dataset, if all 800 000 data points were to be used, the imbalance would be very severe (16% churners). Besides, one staying customer would have significantly more weight in comparison to churned ones, since in the worst case, the staying customers have three data points. Because of that, only the last available for each customer is used. Since the distribution between classes is better in the case where the last appearing datapoint from a customer is selected (Figure 14, Figure 15), it leads to having tolerable proportions 39% (1) churners, and 61% stayed.

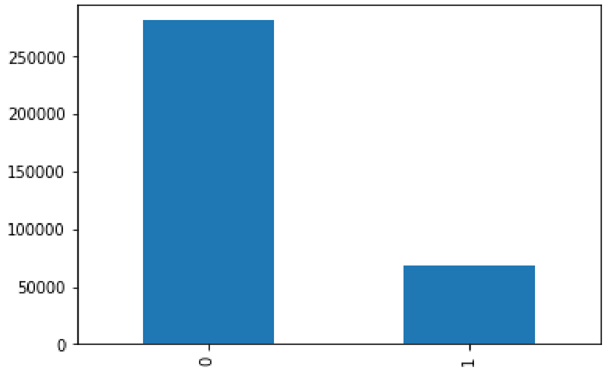


Figure 14 Distribution of churners in the data using the first selected data point

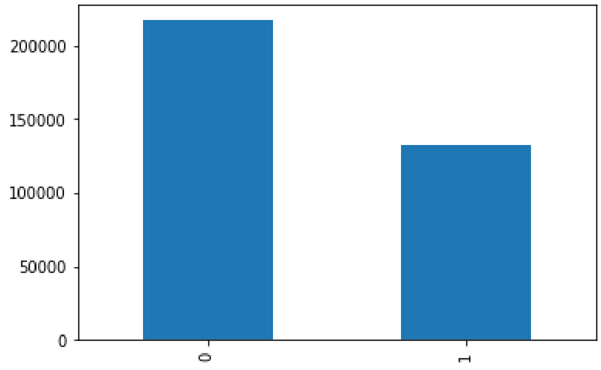


Figure 15 Distribution of churners in the data using the last available data point

As described in the theory part of this thesis, there are few ways of dealing with imbalanced data. The data can either be over or under-sampled. Wherewith over-sampling one would create values and with under-sampling, remove values. Since the imbalance in the current dataset is not that big, it could not be worth the risk of using sampling techniques that would remove information or cause overfitting. Hence the imbalance issue is to be dealt with by hyperparameter optimization in a later phase or by adding weights to classes in models where it is possible.

The second feature, which was found to be imbalanced, was the length of the customer relationship. As we can see from Figure 9, the number of customers in 10 years+ group is significantly more substantial compared to other groups. Since the differences in numbers between long stayed customers and newer customers is so considerable, it could make sense to create a different model for customers that belong to the 10 years+ group. Another factor that also suggests the making of another model is the fact that the distribution between churners and non-churners (Figure 16) in these groups is significantly different.

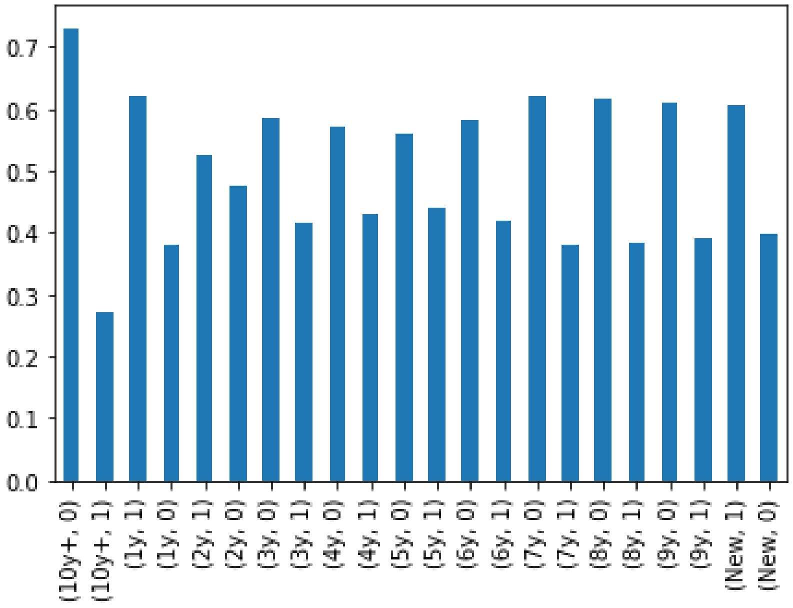


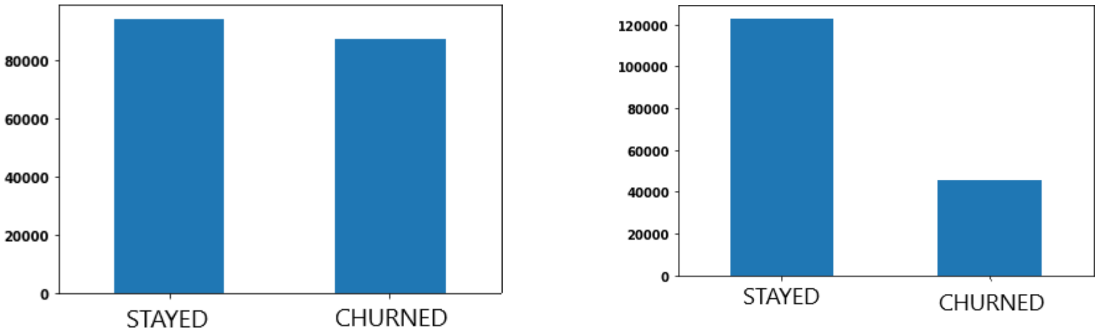
Figure 16 Distribution of churners and non-churners by customer relationship

4.4 Structure of processed data

Because one of the research questions of this study was to compare ML to traditional statistical ways of customer churn prediction. This study uses two different feature sets on top of the suggested old and new customer split. Datasets with “given features” include features provided by the case insurance company, and the datasets with “selected features” include features selected by the algorithm. Additionally, one dataset having all the rows and features is added to have a low processed comparison. All the datasets have gone through the same preprocessing, data cleaning, and dummy variable creation. Thus, this study considers seven different datasets that are:

- OC_G: Old customers with given features
- OC_S: Old customers with selected features
- NC_G: Newer customers with given features
- NC_S: Newer customers with selected features
- ALL_G: Every customer with given features
- ALL_S: Every customer with selected features
- ALL: Every customer and all features

Selected features were 42 features wide, and given features were initially 20 features wide, but after creating dummy variables, they were 37 wide. Old customer datasets were 168 000 long, new customers 181 000 long, and all customers datasets were 350 000 long. Additionally, one dataset features all customers and all features (127) to serve as a baseline for all. From Figure 17, we can see the distribution of churners between new and old customer datasets.



New customers datasets

Old customers dataset

Figure 17 Distribution of churners in new and old customer datasets

Before training and evaluation, all the datasets were split into train and test data by a 75/25 ratio. For this purpose, a function "train_test_split" from Scikit-learn was used. The function picks datapoints randomly, which helps to keep the distributions similar.

4.5 Model selection

Model selection is based on factors discussed in chapters 2 and 3, where models and churn prediction literature are discussed in more detail. Also, good coverage of different models on the ML field is considered. Based on the information provided, the following models were chosen:

1. Logistic regression
2. Support vector machines
3. Random forests
4. K-neighbors classifier
5. AdaBoost with Decision trees
6. Artificial neural network

LR was chosen because it is currently at use in the case insurance company and serves as a baseline to compare other methods. Additionally, according to Table 4, it is very commonly used in CCP; it is simple and easily interpretable. However, it is improbable that it would be the best performer from the classification algorithms, but it is a good comparison against different models.

SVM was also chosen because of the prevalence in CCP literature, even though it is not as used as much as LR. SVM's can handle non-linear relationships well. SVM also offers a good point of comparison since it belongs to another category than other models and offers the possibility to adjust weights between classes, which is a good thing with imbalanced data.

RF was another very prevalent model used in the CCP, and it has been a great performer in terms of accuracy, across different studies but lacks comprehensibility. RF is also the chosen model from the ensemble group.

KNNs were not present in the literature that was considered in the review of this study. However, it has been used in customer retention (Sahar F. Sabbeh, 2018) and has been performing well in classification. Besides, it belongs to the neighbor category, which was not represented.

AB represents boosted methods in this study and has been providing high accuracy in CCP literature. It is used with DT as the classifier, which has been showed to have a good performance in the literature (Tamaddoni, Stakhovych and Ewing, 2016). DT is good for comprehensibility, but it is also easily overfitted to the data and can become very complicated.

ANNs are very common in the CCP literature and have been shown to have high predictive performance in addition to being able to offer probabilities. However, they lack comprehensibility and can complex to build because of the sheer number of options and architectures available.

4.6 Model evaluation

Because the amount of data in this study is quite abundant, models should not get so easily biased because of the variation or not having some information in the training or test data. Hence, k-fold cross-validation is not seen necessary for every model, and a 2-fold method, e.g., splitting the data into test and train data, is used. However, 5-fold cross-validation is used in a grid search. The process starts with fitting the model into training data and evaluating the model's performance on the unseen test data. Then we can compute the accuracy of the model by comparing the actual values from the test data. The most used methods in the literature have been PCC, e.g., accuracy, CM, ROC, and AUC. Also, TDL is very often used but is related to accuracy (Burez and Van den Poel, 2009) and is not seen as necessary. In addition, other metrics such as precision, recall, and F-measure are considered. Especially F-measure, since the imbalance of the classes can lead to high accuracy rating and high precision but weak recall.

The model is supposed to be working in insurance pricing where staying customers would get a better price as they would also be staying for the coming periods. In this case, a more costly scenario for the company could be losing a customer because of not giving out a good offer. On the other hand, selling too cheap could lead to a customer that's losing money for the company. Because of the reasons above, F-measure is slightly weighted to emphasize the recall, i.e., false positives will be used. To summarize the evaluation metrics: first, AUC, accuracy, and F-measure are used to compare models more compactly.
Here, when choosing the best models, more weight is given to F-measure, then to AUC, and lastly, accuracy. After analyzing the metrics mentioned above, CM and ROC-curves are presented for two of the best models.

According to the insurance company concerned in this study, probabilities are more often used in their business. Hence, a comparison between models where it is possible to predict probabilities is carried out with different datasets with best-performing features selected.

5 Model development and results

All datasets are preprocessed with the same labeling and normalization procedures. After preprocessing, every dataset is fitted on every model that has been specified earlier with default parameter settings and tested against test data. Then, a grid search, with 5-fold cross-validation, is used to find the best possible hyperparameter options for the model. However, since there are so many datasets that are also quite large and still very similar, only one dataset is used to run grid search, because it is computationally costly, and it is running with 50 000 rows of data. The dataset chosen for the purpose is ALL_S, which features all the row information and has midmost features. Then, AUC, accuracy, and F-measure are presented for each model with default settings and with grid-searched parameters. After that, CF and ROC curves from two of the best performing models per dataset are presented. Then, the MSE scores of models that can predict probabilities are shown. Finally, in the last chapter, results are analyzed, and a comparison between models is made. Furthermore, decisions regarding best performing models are made, and answers to the second subsection of research questions are given.

5.1 Logistic regression

The most significant hyperparameters for logistic regression are solver, penalty, and regularization or C. Solver is the algorithm used in the optimization problem, penalty is the norm used in the penalization, and C is the inverse regularization parameter to the lambda parameter in the LR. The default setting for logistic regression in Scikit-learn is solver = “liblinear”, penalty = l2, regularization/C =1.0. The results for different datasets before the grid search can be found in Table 5.

Table 5 LR metrics

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.51	0.56	0.63	0.68	0.68	0.66	0.67	0.62
F-score	0.05	0.20	0.57	0.64	0.42	0.52	0.53	0.42
Accuracy	0.73	0.74	0.63	0.68	0.62	0.71	0.72	0.69

After running the grid search on the ALL_S dataset, it was concluded that the best performing values were the same values as the default values. Hence, no further optimizations are done regarding LR.

5.2 Support vector machines

Hyperparameter optimization should have a significant impact on the performance of Support vector machines (Laref *et al.*, 2019) since it directly affects how, for example, the kernel can separate the classes with the hyperplanes. However, because the dataset in this study is significant, only linear SVM kernel is computationally practical. Usually, the kernel would be very significant to optimize. However, without kernel, parameters to be optimized are penalty, loss, and C.

The default settings of linear SVM are penalty=“l2”, loss = “squared_hinge” and C=1. The values for the default model can be seen in Table 6.

Table 6 SVM metrics

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.51	0.54	0.63	0.68	0.62	0.66	0.67	0.62
F-score	0.05	0.15	0.57	0.63	0.41	0.50	0.51	0.40
Accuracy	0.73	0.74	0.63	0.68	0.68	0.71	0.72	0.70

As with LR, the grid search did not suggest any enhancements to the default settings. This could be because the linear kernel was the only feasible that could be selected and so restricted the available hyperparameter selection.

5.3 Random forests

RF's have many parameters that could be altered to find the best fit. The following parameters were tried: bootstrap, maximum tree depth, maximum features considered for a split, minimum samples per leaf, minimum samples per split, and the number of estimators.

Bootstrapping (default=true) is a Boolean value and is used to set whether bootstrapped samples or the whole dataset is used to build each tree. M*aximum tree depth* (default=None) limits the

maximum depth of the tree. *Maximum features considered for a split* is the number of features that are considered when looking for the best split. *Minimum samples per leaf* (default=1) is the minimum number of samples that are required at the leaf node. *Minimum samples per split* (default=2) specifies the number of samples that are required in a node so that it's considered for splitting. *The number of estimators* (default=10) defines how many trees there are in the forest. Results with default values are in Table 7.

Table 7 RF metrics before a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.54	0.72	0.60	0.74	0.60	0.75	0.75	0.67
F-score	0.21	0.55	0.55	0.68	0.44	0.64	0.64	0.53
Accuracy	0.71	0.82	0.60	0.74	0.65	0.78	0.79	0.72

After running grid-search with the dataset following parameters were found:

- Bootstrapping = false
- Maximum tree depth = 70
- Minimum samples per leaf = 2
- Minimum samples per split = 2
- Number of estimators = 1000

Table 8 RF metrics after a grid search (change)

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.54 (0)	0.73 (0.01)	0.63 (0.03)	0.74 (0)	0.62 (0.02)	0.77 (0.02)	0.78 (0.03)	0.69 (0.02)
F-score	0.16 (0.05)	0.58 (0.03)	0.58 (0.03)	0.68 (0)	0.44 (0.00)	0.68 (0.04)	0.69 (0.05)	0.54 (0.01)
Accuracy	0.73 (0.02)	0.84 (0.02)	0.63 (0.03)	0.74 (0)	0.68 (0.03)	0.80 (0.02)	0.81 (0.02)	0.74 (0.02)

From Table 8, we can see that with parameter optimization, almost all of the metrics at least stayed the same or improved. The most considerable improvement was in the dataset holding all of the information, and no change was seen with the NC_S dataset. Also, the F-score of the OC_G was lower than before.

5.4 K-neighbors classifier

KNN also has a few parameters to optimize. Parameters were chosen to optimize where the number of neighbors, weights, and metrics. *The number of neighbors* (default=5) means how many neighbors are used in a query of finding K-neighbors of a point. *Weight* (default=uniform) is the function that is used to give weight for distances between neighbors in neighborhoods because sometimes it is better to give more weight to neighbors nearby. *Metric* (default=minkowski) is the distance metric used for each neighbor. The results with default parameters can be seen in Table 9.

Table 9 KNN metrics before a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.55	0.63	0.59	0.66	0.60	0.67	0.67	0.62
F-score	0.25	0.41	0.56	0.65	0.46	0.57	0.56	0.49
Accuracy	0.70	0.74	0.60	0.66	0.65	0.70	0.70	0.68

After running the grid search, the following parameters were chosen:

- Number of neighbors = 19
- Weight = distance
- Metric = Manhattan

Table 10 KNN metrics after a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.54 (0.01)	0.65 (0.02)	0.61 (0.02)	0.70 (0.04)	0.61 (0.08)	0.71 (0.04)	0.70 (0.03)	0.65 (0.03)
F-score	0.20 (0.05)	0.42 (0.01)	0.5	0.69 (0.04)	0.45 (0.09)	0.61 (0.04)	0.58 (0.02)	0.50 (0.01)

Accuracy	0.72 (0.02)	0.78 (0.04)	0.61 (0.01)	0.70 (0.04)	0.66 (0.09)	0.74 (0.04)	0.73 (0.03)	0.71 (0.03)

In terms of performance, increase KNN scores improved significantly across the board after the grid search. ALL_G scores were most affected and improved by 0.09, almost in all metrics.

5.5 AdaBoost with Decision trees

As AdaBoost is a booster classifier, it uses some underlying classifier that is fitted on the original dataset. Then the more of the same classifier are fitted on the same dataset, but incorrectly classified instances are given more weight in coming fittings so that successive classifiers will focus more on severe cases. DT classifier has been chosen as the underlying classifier in this study. However, since there are not many parameters that can be given for AdaBoost, the grid search is done on DT. The following parameters are considered: criterion, minimum samples per split, maximum depth of trees, minimum samples per leaf, and the maximum number of leaf nodes.

Criterion (default=gini) is the function used to measure the quality of a split. *Minimum samples per split* (default=2), *maximum depth of trees* (default=None), *minimum samples per leaf* (default=1) were explained in chapter 5.3. *The maximum number of leaf nodes* (default=None) is the maximum number of leaves a tree can grow to. The results with default values can be found in Table 11.

Table 11 AB metrics before a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.52	0.70	0.64	0.73	0.62	0.74	0.74	0.67
F-score	0.08	0.52	0.58	0.70	0.43	0.64	0.65	0.51
Accuracy	0.74	0.82	0.64	0.73	0.68	0.77	0.77	0.74

After running the grid search, the following settings were found:

- Criterion = gini
- Minimum samples per split=10
- Maximum depth of trees=10
- Minimum samples per leaf=2
- Maximum number of leaf nodes=None

Table 12 AB metrics after a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.54 (-0.02)	0.70 (0)	0.61 (-0.03)	0.71 (-0.02)	0.61 (-0.01)	0.73 (-0.01)	0.73 (-0.01)	0.66 (-0.01)
F-score	0.21 (0.13)	0.54 (0.02)	0.56 (-0.02)	0.69 (-0.01)	0.45 (0.02)	0.64 (0)	0.66 (-0.01)	0.55 (0.04)
Accuracy	0.71 (-0.03)	0.78 (0.04)	0.61 (-0.03)	0.71 (-0.02)	0.66 (-0.02)	0.75 (-0.02)	0.76 (-0.01)	0.72 (-0.02)

With the AB classifier, the grid search yielded not great results. Almost all of the metrics except ALL_G, fell except for the rise in F-score in OC_G. It could be that the grid search in AB is more dependent on the correct dataset onto which it is fitted since ALL_G's performance was improved.

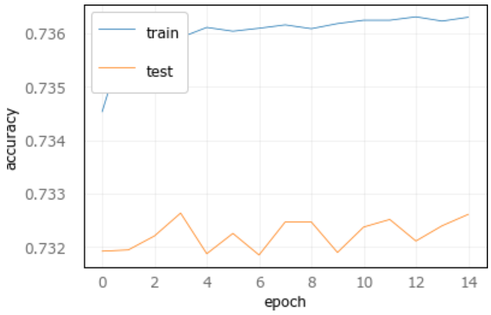
5.6 Artificial neural network

Because ANN provides the highest number of settings, features, and different ways of structuring the model, first, a novel baseline architecture is used as a default. It will have the basic structure that was presented in Figure 5, which means one input layer, one hidden layer, and one output layer. The number of nodes in the hidden layer was decided using a general rule: "mean of nodes in the input and output layer." This means, for example, for the NC_S dataset containing 42 features, 21 nodes were used. Then, the loss function used was "binary cross-entropy," the optimizer was "adam," and activation was "linear" in the first layer and "sigmoid" in the last, which produces outputs between one and zero. Then for the epochs, i.e., how many times the ANN iterates trough the dataset to learn, 15 was chosen. Results can be seen from Table 13, and the metrics by epoch from Table 14. We can see that the accuracy in most cases still would have a rising trend, which would suggest using more epochs.

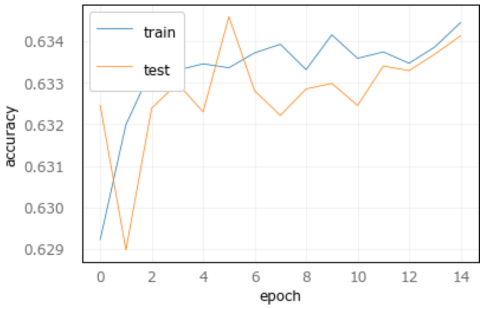
Table 13 ANN metrics before enhancements

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	AL L	AV G
AUC	0.52	0.56	0.63	0.68	0.62	0.67	0.68	0.62
F-score	0.05	0.20	0.58	0.62	0.42	0.52	0.55	0.42
Accuracy	0.73	0.74	0.63	0.68	0.68	0.71	0.71	0.70

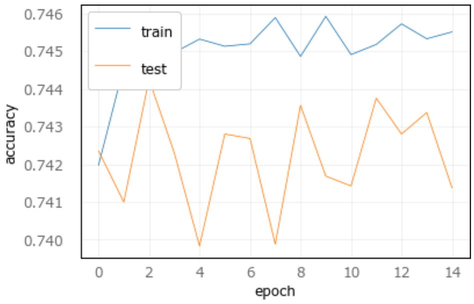
Table 14 ANN accuracy by epoch before grid search



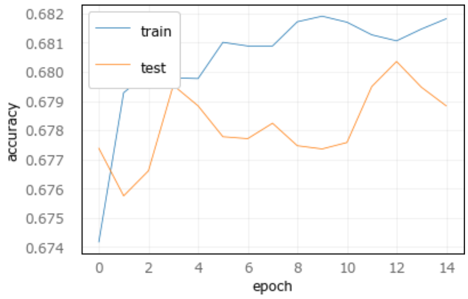
OC_G



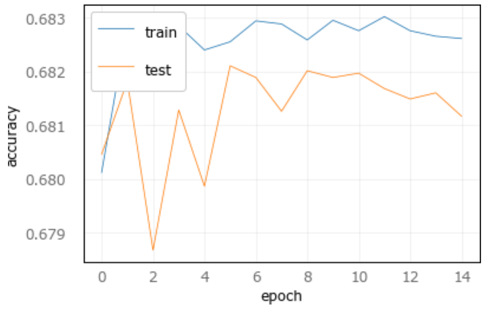
NC_G



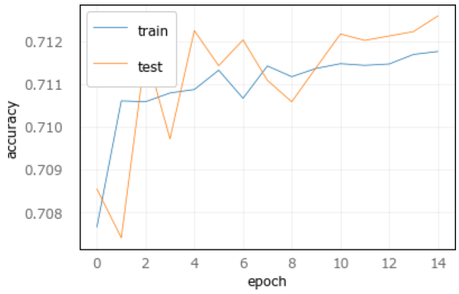
OC_S



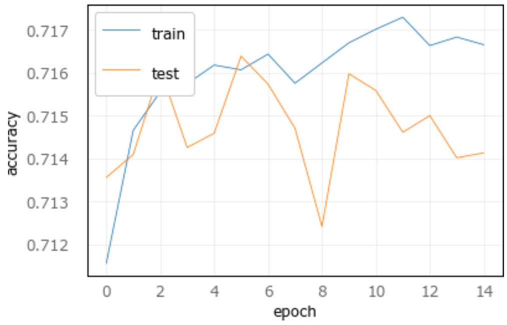
NC_S



ALL_G



ALL_S



ALL

ANN provides the highest number of different variables, which makes the grid search even more computationally heavy. Because of this reason, only 20 000 rows of ALL_S were used. Different parameters that were tested and found were:

- Batch size: 500
- Epochs: 100
- Optimizer: “SGD”
- Learning rate: 0.3
- Momentum: 0.9
- Initialization mode: normal
- Activation algorithm of the first layer: “relu”
- Kernel constraint: 3
- Dropout rate: 0.2

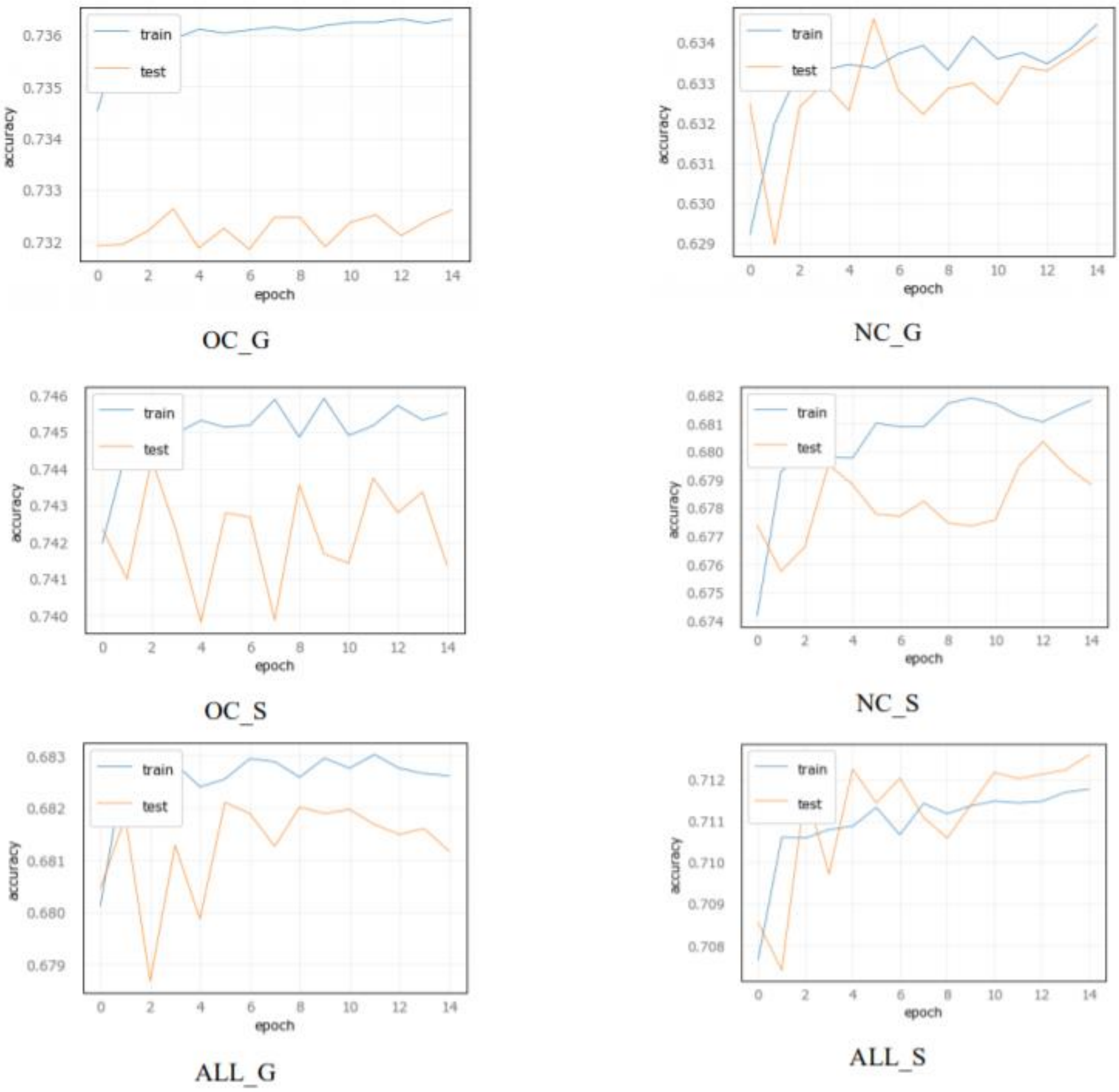
Batch size is the number of samples per gradient update. *Optimizer* is an algorithm that tries to minimize or maximize the objective or error function. *Learning rate* is the amount by how much the weights of the nodes are updated during training. *Momentum* is used to take past gradients into account and smooth out the steps of gradient descent. *Initialization mode* defines how the initial random weights on layers are set. *Activation algorithm* is used to convert the input signal to an output signal. *Kernel constraint* allows setting constraints on network parameters during optimization. *Dropout rate* helps with the overfitting problem and drops out values randomly during the training phase. (Chollet and others, 2015) The architecture of the ANN was also changed to having one more hidden layer with a number equal to one-fourth of the features in a dataset. In addition, two random dropout layers were added between the two hidden layers,

with the founded dropout rate, to prevent overfitting. From Table 15 and Table 16, we can see that the results improved after enhancements to the infrastructure and hyperparameters.

Table 15 ANN metrics after a grid search

	OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL	AVG
AUC	0.52 (0)	0.65 (0.09)	0.64 (0.01)	0.72 (0.04)	0.62 (0)	0.70 (0.03)	0.73 (0.05)	0.65 (0.03)
F-score	0.08 (0.03)	0.40 (0.20)	0.58 (0)	0.66 (0.04)	0.39 (0.03)	0.54 (-0.02)	0.60 (0.05)	0.46 (0.04)
Accuracy	0.74 (0.01)	0.80 (0.06)	0.64 (0.01)	0.73 (0.05)	0.68 (0)	0.76 (-0.05)	0.79 (0.08)	0.73 (0.03)

Table 16 ANN Accuracy by epoch after a grid search



OC_G

NC_G

OC_S

NC_S

ALL_G

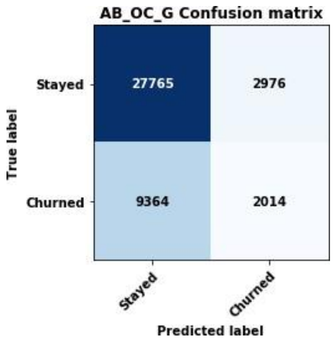
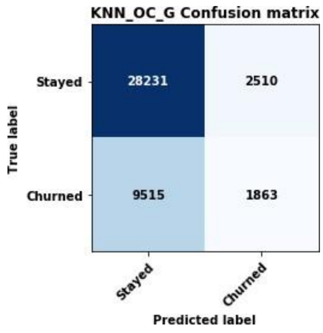
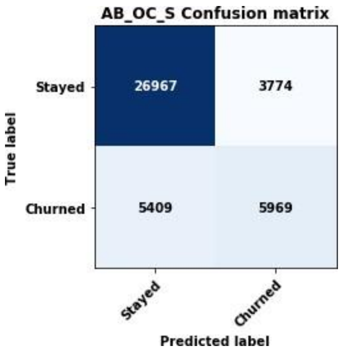
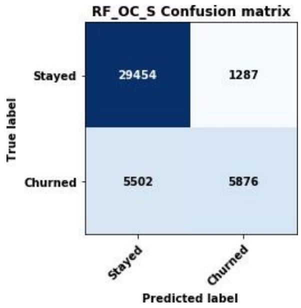
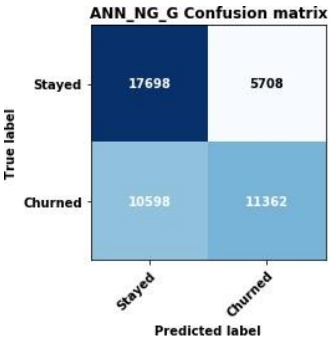
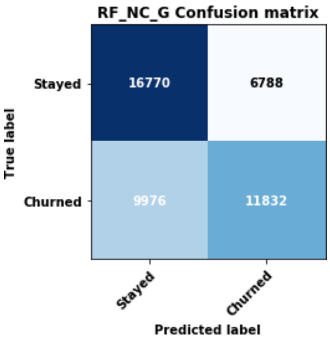
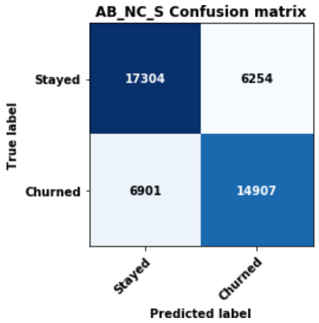
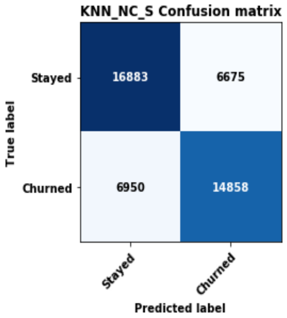
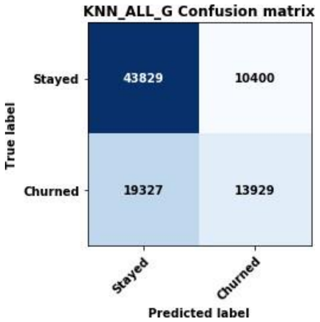
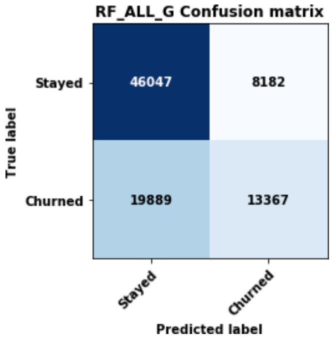
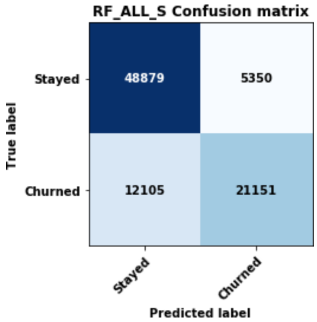
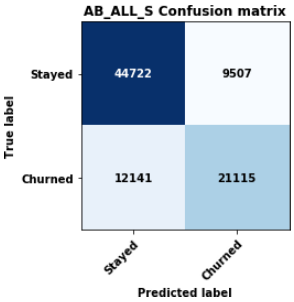
ALL_S

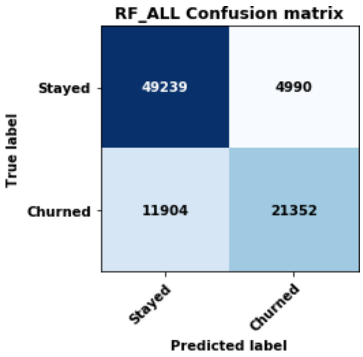
5.7 Confusion matrixes

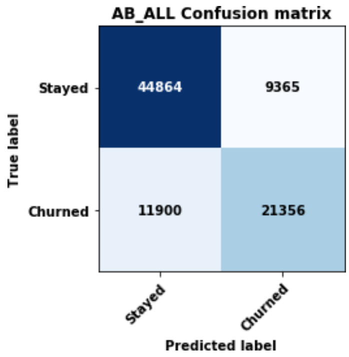
As said, the CFs from only two of the best performing models by dataset would be represented because the number of different CFs there would be quite large, and the representation would not be necessary since their performance can quite well be seen from metrics represented in their chapters. The models to be shown in Table 17 were chosen by comparing measurements in a manner explained in the model evaluation chapter.

From the CFS (Table 17), we can see three trends. Firstly, datasets with selected features are performing significantly better when compared against features given by the insurance company. Secondly, the false-negative rate seems to be much higher with datasets containing old customers. Lastly, the datasets with all rows and features seemed to be performing quite well even though not as well as the new customers with selected features datasets.

Table 17 Confusion matrixes

			
OC_G		OC_S	
			
NG_G		NC_S	
			
ALL_G		ALL_S	

<table>
<tr>
<td>


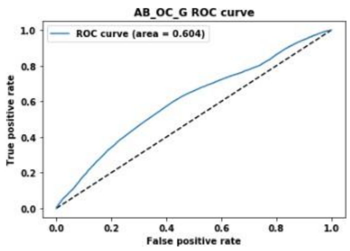
</td>
<td>


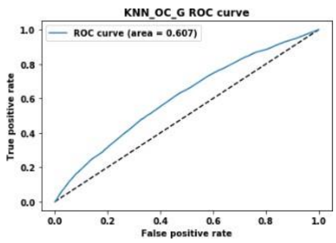
</td>
</tr>
<tr>
<td colspan="2">ALL</td>
</tr>
</table>

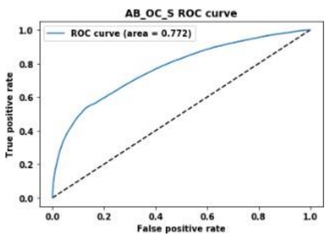
5.8 Receiver operating characteristic curves

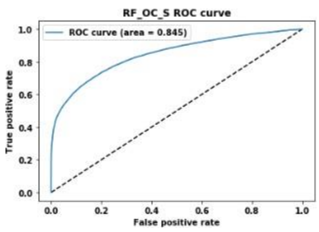
Just as explained in the previous chapter, the two best performing models by dataset have been gathered into Table 18. From the ROC-curve, we can see that all models did at least slightly better when compared to the random dotted line. There are again significant differences between the results from selected features and given features, where given feature datasets performed worse compared to selected features datasets. However, the differences between old and new customer models are not that clear by looking at the pictures or AUC-scores. ALL datasets performed again at least as well as datasets with selected features, but the difference between ALL_S and ALL was not that different, having only 0.009 difference in AUC-score.

Table 18 ROC-curves

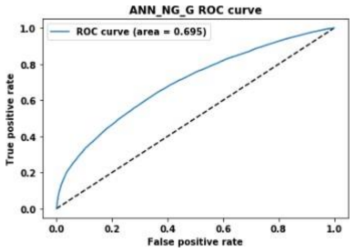
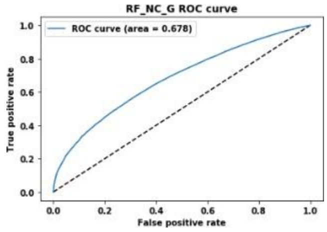
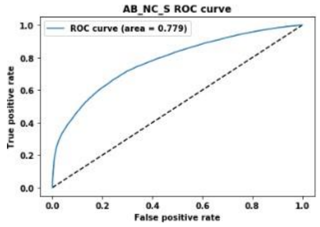
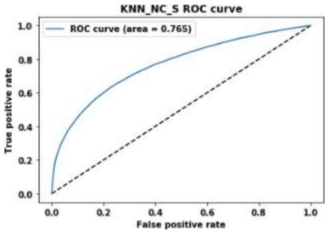
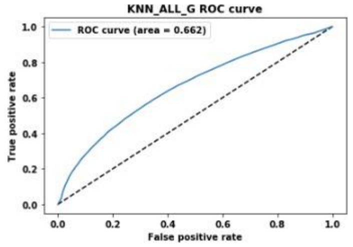
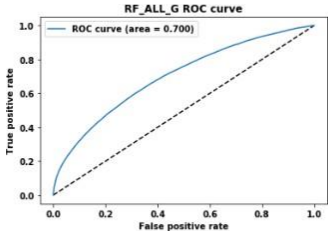
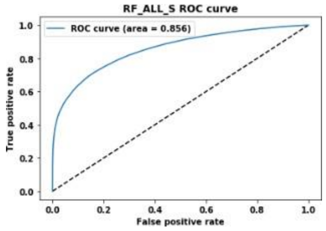
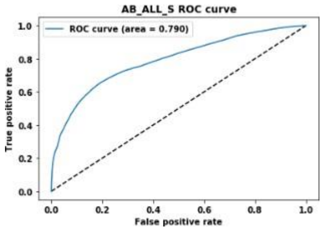
<table>
<tr>
<td>


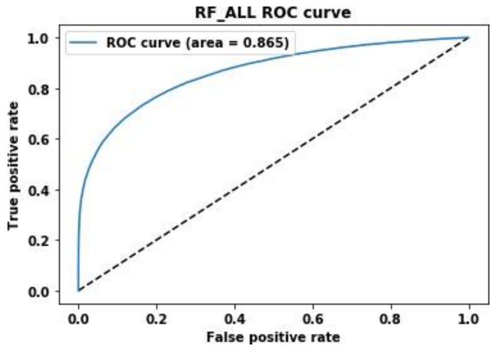
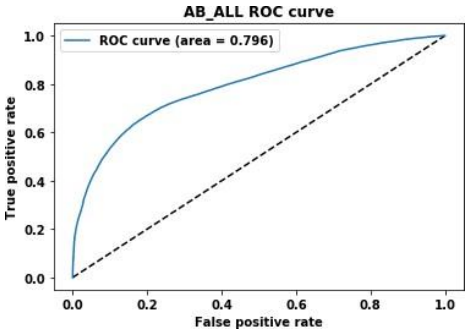
</td>
<td>


</td>
<td>


</td>
<td>


</td>
</tr>
<tr>
<td colspan="2">OC_G</td>
<td colspan="2">OC_S</td>
</tr>
</table>

			
NG_G		NC_S	
			
ALL_G		ALL_S	

	
ALL	

5.9 Probability modeling

From the chosen models LR, RF, KNN, AB, and ANN were able to predict probabilities between classes. The results of different models can be seen in Table 19.

Table 19 MSE of different models on datasets

	OLD	NEW	ALL	AVG
LR	0.18	0.21	0.19	0.19
RF	0.12	0.16	0.14	0.14
KNN	0.16	0.20	0.18	0.18
AB	0.21	0.23	0.22	0.22
ANN	0.15	0.18	0.17	0.17

All of the models were relatively close to each other when comparing the average performance on datasets. Still, RF was the best performer in the probability modeling across all the datasets. ANN was the second and KNN, the third-best using MSE metric. AB, however, did not perform that well in this section.

5.10 Summary and analysis of the results

Based on the results, the most performing models seem to be RF, AB, and KNN. The results compared to the literature review are not surprising, as a similar performance of RF and AB has been demonstrated in the literature (Vafeiadis *et al.*, 2015; Tamaddoni, Stakhovych and Ewing, 2016; Faris, 2018). However, KNN has not been that prevalent in the literature or has not performed that well (Keramati *et al.*, 2014) its performance was very close with ANN but was chosen as a better model because of slightly higher F-score. In the end, it should be mentioned that the spreads between performance metrics between all of the models were only slightly different.

Then, the sub-question 2a can be answered, which was: *"How different methods compare to one another?"*.

The performance of the SVM was expected since SVMs do not fit well into more massive datasets (Cervantes *et al.*, 2008) and its performance in simple form has been seen to show average results (Coussement and Van den Poel, 2008; Coussement, Lessmann and Verstraeten, 2017; Faris, 2018). RF has been shown to outperform SVM (Coussement and Van den Poel, 2008) which makes the results to be aligned with the literature. However, ANN was performing quite poorly compared to literature (Sharma, Panigrahi and Kumar, 2011; Keramati *et al.,* 2014). It could be that either the architecture of the model was not good enough because very high performance has been achieved with hybrid models (Mohammadi, Tavakkoli-Moghaddam and Mohammadi, 2013; Sivasankar and Vijaya, 2018) or the amount of data still was not enough for the model to learn. LR's performance was also as suspected as it has been shown that forest techniques are able outperform it (Larivière and Van Den Poel, 2005).

RF seems to perform well across different datasets by having the best average score in all of the metrics except the F-score, where AB bested it by 0.01. Looking at CMs, RF has lower amounts of false positives across all datasets, which has been defined as a more weighted metric in this thesis. Additionally, when looking at ROC-curves between AB and RF, we can see that RF seems to perform better. However, RF was seen to be a computationally heavy model and using significantly more memory compared to other models. Complex and big models could imply very low entropy and hence, very complex models, which could also lead to lousy generalizability and overfitting.

AB can be seen as the second-best model across the datasets. The differences between AUC and accuracy metrics on average are not that considerable, even though the differences in ROC- curves are noticeable. By looking at CMs, AB seems to be introducing more false positives regarding churning customers when compared to RF but would seem to be tied with KNN in that regard. The good thing about AB is that the model was not computationally very demanding. The memory consumption was tolerable, and predictions worked fast.

KNN can be seen as the third-best model. It is almost tied with ANN regarding the average scores but can have better performance in F-score. KNN performs quite similarly than AB in multiple datasets, for example, NC_S, but suffers from significantly worse performance in some datasets such as OC_S. KNN is also computationally very slow every time predictions have to be made, even though the memory consumption is tolerable. A comparison of average metrics can be seen in Table 20.

Table 20 Summary of average metrics

	RF	AB	KNN	ANN	SVM	LR
AUC	0.69	0.66	0.65	0.65	0.62	0.62
F-Score	0.54	0.55	0.5	0.46	0.4	0.42
Accuracy	0.74	0.72	0.71	0.73	0.7	0.69

Then, the second sub-question 2b, "*How does machine learning compare against current methods used in the insurance company?*".

Currently, the insurance company referred to in this study, is using logistic regression as a basis which it is using to model which variables are significant when trying to predict customer churn. However, any classification predictions regarding churning are not being made. Hence, this study proves at least two things. Firstly, by using ML methods, it is possible to predict the churning of their customers with quite reliable accuracy, especially amongst new customers. However, all of the ML methods were only slightly better than LR. Secondly, a feature selection algorithm was able to select variables that made predictions more accurate by introducing only five more features. From Table 21, we can see that selected datasets were able to achieve, on average, almost 0.1 better scores while still performing at least as well as the dataset containing all of the features without adding unnecessary complexity.

Additionally, in chapter 5.9 the performances of different models on probability prediction were introduced, where the currently used LR method was outperformed by multiple ML methods, but only slightly. However, probability prediction is not the primary research objective of this study.

Table 21 Average prediction scores by dataset

OC_G	OC_S	NG_G	NC_S	ALL_G	ALL_S	ALL
0.46	0.60	0.61	0.69	0.57	0.68	0.69

Lastly, the second research question was: "*What is the most suitable machine learning model to be used to predict future customer churn for the given dataset on customer feature data?*".

According to the previous discussion and the results of this study, the most suitable method that could be suggested would be RF, which is in line with the literature. RF was able to perform best, from chosen models, on imbalanced datasets and less imbalanced datasets. It was able to capture the variance of the data but had its limitations regarding memory consumption and possible unnecessary complexity of the model. Hence, also AB could be suggested as a second option, as its performance was almost as good as RF's but without the significant memory impact.

6 Conclusions

This chapter concludes the results and implications emerging from the results of this study. Furthermore, limitations and future research suggestions are discussed. Firstly, customer churn in the insurance company was possible to be predicted using ML methods with a quite good performance and accuracy (RF: 0.74, AB:0.72 and KNN: 0.71) by using the dataset provided, and at least slightly better performance in comparison to current methods in the case insurance company, was shown. Secondly, according to the results drawn from the previous chapter, it would seem that both RF and AB seem to be good performing models, but RF being the preferred method. The result seems logical because RF's and AB have been performing well in the CCP field (De Caigny, Coussement and De Bock, 2018; Sahar F. Sabbeh, 2018).

6.1 Analysis of results

The results were in line with the current literature. It was shown that RF and AB were both performing the best on the datasets of this study and are also top performers in other studies. Also, CCP can be done with the current dataset with some reliability and that the performance of ML models compared to statistical models could be at least slightly better. However, the spread in performance metrics with each model regarding evaluated metrics were only somewhat different. A more significant performance gain was seen when making comparison between the performance on features selected by an algorithm or provided by the insurance company. It was shown that by adding five more features selected by a feature selection algorithm, it was possible to enhance the performance of the model on different datasets. These results were across all datasets regardless of customer relationship duration. The algorithm was able to reduce the complexity of the model by reducing the variables from 127 to 42, while still making the models perform as well.

6.2 Limitations and future research

There are multiple considerations regarding the limitations faced in this study and future research on this field and continuing the work of this thesis. First, the data and so information in this study had to be cut down to third because only one row per customer was used. The insurance company required this limitation, but it also kept the scope of this thesis on a higher level. Limiting the scope allowed to get a better overview of the current field of CCP and concentrate more on the comparison of different ML models and current ways of working in the insurance company compared to ML methods. However, in future research, it could make sense to include the time dimension of a customer to ML models to get more information regarding a customer. Taking the time dimension into account could also relieve the imbalance problem since customers would be seen in a different light by the model.

Furthermore, future research could be conducted by using either more data, other features, or datasets from the current insurance company. Especially other features could make sense because the number of features was cut down by more than half by the feature selection algorithm. Additionally, multiple studies have been able to get high accuracy on CCP by using behavioral data, which would also be interesting.

Secondly, this study did not take comprehensibility into account, which has been in interest on the CCP field, as was presented in the literature review of this thesis. It makes sense for companies to also get the information out of the model *why* their customers are churning, not just that they are churning. From some of the models used in this study, it would be possible to extract the feature importance of different features, but it isn’t included in the scope.

Thirdly, the probability prediction was only touched on the surface of this study. It was not meant to be by no means comprehensible and served more as just a piece of information and comparison to current models used in the insurance company. Additionally, it was again shown that the more modern methods were outperforming the LR, which was used in the insurance company. However, the models were performing on average quite similarly, which could imply that the tradeoff in rising complexity using the suggested ML models in comparison to LR is not necessarily worth it regarding the probability predictions.

Fourthly, future research could include other, more complex models to predict the churn. Some examples could include weighted random forests (Burez and Van den Poel, 2009), hybrid models (Farquad, Ravi and Raju, 2014; Sivasankar and Vijaya, 2018), or unsupervised methods such as clustering that could be used on unstructured data. This way, it would be possible to mine out features that could be used in future CCP research in the insurance company. Other ways the models could be improved would be using hybrid models that were achieving significant performance gains, as explained in the literature review.

Lastly, extensive grid searches and other optimizations on a large dataset are not feasible on a home computer because they require multiple iterations to find optimal settings, which could affect the performance of the models significantly. Hence, in future research, either more performant computers are suggested to be used or for example, cloud computing.

7 References

Ahmed, A. and Linen, D. M. (2017) ‘A review and analysis of churn prediction methods for customer retention in telecom industries’, in *2017 4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017*. IEEE, pp. 1–7. doi: 10.1109/ICACCS.2017.8014605.

Aksoy, S. and Haralick, R. M. (2001) ‘Feature normalization and likelihood-based similarity measures for image retrieval’, *Pattern Recognition Letters*, 22(5), pp. 563–582. doi: 10.1016/S0167-8655(00)00112-4.

Amin, A. *et al.* (2019) ‘Customer churn prediction in telecommunication industry using data certainty’, *Journal of Business Research*, 94, pp. 290–301. doi: 10.1016/j.jbusres.2018.03.003.

Au, W. H., Chan, C. C. and Yao, X. (2003) ‘A novel evolutionary data mining algorithm with applications to churn prediction’, *IEEE Transactions on Evolutionary Computation*, 7(6), pp. 532–544. doi: 10.1109/TEVC.2003.819264.

Ballings, M. and Van Den Poel, D. (2012) ‘Customer event history for churn prediction: How long is long enough?’, *Expert Systems with Applications*, 39(18), pp. 13517–13522. doi: 10.1016/j.eswa.2012.07.006.

Bayerstadler, A., Van Dijk, L. and Winter, F. (2016) ‘Bayesian multinomial latent variable modeling for fraud and abuse detection in health insurance’, *Insurance: Mathematics and Economics*, 71, pp. 244–252. doi: 10.1016/j.insmatheco.2016.09.013.

Bergstra, J. and Bengio, Y. (2012) ‘Random Search for Hyper-parameter Optimization’, *J. Mach. Learn. Res.* JMLR.org, 13, pp. 281–305. Available at: http://dl.acm.org/citation.cfm?id=2188385.2188395.

Beyramysoltan, S., Rajkó, R. and Abdollahi, H. (2013) ‘Investigation of the equality constraint effect on the reduction of the rotational ambiguity in three-component system using a novel grid search method', *Analytica Chimica Acta,* 791, pp. 25–35. doi: https://doi.org/10.1016/j.aca.2013.06.043.

Bradley, A. P. (1997) 'The use of the area under the ROC curve in the evaluation of machine learning algorithms', *Pattern Recognition*. Pergamon, 30(7), pp. 1145–1159. doi: 10.1016/S0031-3203(96)00142-2.

Buckinx, W. and Van Den Poel, D. (2005) 'Customer base analysis: Partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting', *European Journal of Operational Research,* 164(1), pp. 252–268. doi: 10.1016/j.ejor.2003.12.010.

Burez, J. and Van den Poel, D. (2007) 'CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services', *Expert Systems with Applications,* 32(2), pp. 277–288. doi: 10.1016/j.eswa.2005.11.037.

Burez, J. and Van den Poel, D. (2009) 'Handling class imbalance in customer churn prediction', *Expert Systems with Applications*. Pergamon, 36(3 PART 1), pp. 4626–4636. doi: 10.1016/j.eswa.2008.05.027.

De Caigny, A., Coussement, K. and De Bock, K. W. (2018) 'A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees', *European Journal of Operational Research,* 269(2), pp. 760–772. doi: 10.1016/j.ejor.2018.02.009.

Cervantes, J. *et al.* (2008) 'Support vector machine classification for large data sets via minimum enclosing ball clustering', *Neurocomputing,* 71(4), pp. 611–619. doi: https://doi.org/10.1016/j.neucom.2007.07.028.

Chandrashekar, G. and Sahin, F. (2014) 'A survey on feature selection methods', *Computers and Electrical Engineering,* 40(1), pp. 16–28. doi: 10.1016/j.compeleceng.2013.11.024.

Chawla, N. V *et al.* (2002) *SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research*. Available at: https://arxiv.org/pdf/1106.1813.pdf (Accessed: 14 August 2019).

Chen, Z.-Y., Fan, Z.-P. and Sun, M. (2012) ‘Decision support A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data’, *European Journal of Operational Research*, 223, pp. 461–472. doi: 10.1016/j.ejor.2012.06.040.

Chollet, F. and others (2015) ‘Keras’.

Claesen, M. and De Moor, B. (2015) ‘Hyperparameter Search in Machine Learning’. Available at: http://arxiv.org/abs/1502.02127 (Accessed: 15 September 2019).

Cortes, C. and Vapnik, V. (1995) ‘Supprot-Vector Networks’, *Machine Learning*, 20(20), pp. 273–297. doi: 10.1111/j.1747-0285.2009.00840.x.

Coussement, K., Lessmann, S. and Verstraeten, G. (2017) ‘A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry’, *Decision Support Systems*, 95, pp. 27–36. doi: 10.1016/j.dss.2016.11.007.

Coussement, K. and Van den Poel, D. (2008) ‘Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques’, *Expert Systems with Applications*, 34(1), pp. 313–327. doi: 10.1016/j.eswa.2006.09.038.

Datta, P. *et al.* (2000) ‘Automated cellular modeling and prediction on a large scale’, *Artificial Intelligence Review*. Kluwer Academic Publishers, 14(6), pp. 485–502. doi: 10.1023/A:1006643109702.

David, M. (2015) ‘A review of theoretical concepts and empirical literature of non-life insurance pricing’, *Procedia Economics and Finance*, 20, pp. 157–162. doi: 10.1016/S2212- 5671(15)00060-X.

Dietterich, T. G. (2000) ‘Ensemble Methods in Machine Learning’, in *International workshop on multiple classifier systems*. Heidelberg: Springer, Berlin, pp. 1–15. doi: 10.1007/3-540-45014-9_1.

Drummond, C. and Holte, R. C. (2003) ‘C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling’, *Physical Review Letters*. doi: 10.1103/PhysRevLett.91.039901.

Eiben, A. E., Koudijs, A. E. and Slisser, F. (1998) ‘Genetic modelling of customer retention’, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 178–186. doi: 10.1007/BFb0055937.

Elliott, T. (2019) *The State of the Octoverse: machine learning*. Available at: https://github.blog/2019-01-24-the-state-of-the-octoverse-machine-learning/ (Accessed: 5 September 2019).

Fang, K., Jiang, Y. and Song, M. (2016) ‘Customer profitability forecasting using Big Data analytics: A case study of the insurance industry’, *Computers & Industrial Engineering*, 101, pp. 552–564. doi: 10.1016/j.cie.2016.09.011.

Faris, H. (2018) ‘A hybrid swarm intelligent neural network model for customer churn prediction and identifying the influencing factors’, *Information (Switzerland)*, 9(11), p. 288. doi: 10.3390/info9110288.

Farquad, M. A. H., Ravi, V. and Raju, S. B. (2014) ‘Churn prediction using comprehensible support vector machine: An analytical CRM application’, *Applied Soft Computing Journal*, 19, pp. 31–40. doi: 10.1016/j.asoc.2014.01.031.

Fawcett, T. (2006) ‘An introduction to ROC analysis’, *Pattern Recognition Letters*, 27(8), pp. 861–874. doi: 10.1016/j.patrec.2005.10.010.

Galar, M. *et al.* (2012) ‘A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches’, *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, pp. 463–484.

doi: 10.1109/TSMCC.2011.2161285.