Robert Nygård

AI-Assisted Lead Scoring

Master's Thesis in Information Systems
Supervisors: Asst. Prof. Jozsef Mezei
Dr. Markku Heikkilä
Faculty of Social Sciences, Business and Economics
Åbo Akademi University

Åbo 2019

Abstract

Subject: Information Systems	
Writer: Robert Nygård	
Title: AI-Assisted Lead Scoring	
Supervisor: Asst. Prof. Jozsef Mezei	**Supervisor:** Dr. Markku Heikkilä

Abstract:
Companies often gather a tremendous amount of data, such as browsing behavior, email activities and other contact data. This data can be used to estimate a contact's purchase probability using predictive analytics. The purchase probability can then be used by companies to solve different business problems, such as optimizing their sales department's call list. The purpose of this thesis is to study how machine learning can be used to perform lead scoring. Historical behavioral data is used as training data for the classification algorithm, and purchase moments are used to limit the behavioral data for the contacts that have purchased a product in the past. Different ways of aggregating time-series data are tested to ensure that limiting the activities for buyers does not result in model bias. Model performance was estimated using cross-validation. The results suggest that it is possible to estimate the purchase probability of leads using supervised learning algorithms, such as random forest, and that it is possible to obtain business insights from the results using visual analytics.

Keywords: Machine learning, Lead scoring, Purchase probability, Supervised learning, Classification, Predictive analytics, Random forest, Decision tree, Logistic regression, Neural networks, Data science, Data visualization, Marketing automation, CRM

Date: 10.05.2019	**Number of pages:** 78

1. Introduction **4**

1.1 Background 4

1.2 Manual lead scoring 5

1.3 Predictive analytics 7

1.4 Research questions 9

2. Literature review **10**

2.1 Lead scoring 10

2.1.1 The importance of customer attributes in lead scoring 11

2.1.2 Lead scoring conversion rate enablers 12

2.2 Machine learning 14

2.2.1 Classification tree models 14

2.2.2 Random forest 16

2.2.3 Logistic regression 17

2.2.4 Neural networks 19

2.2.5 Missing data 21

2.2.6 Data splitting and resampling 22

2.2.7 Evaluation metrics 23

2.2.8 Handling class imbalance 27

2.2.9 Practical applications of machine learning on CRM 28

3. Methodology **31**

3.1 Methodology selection 31

3.2 Quantitative methodology 33

3.3 Empirical study structure 35

4. Empirical study: AI assisted lead scoring **37**

4.1 Business objective 37

4.2 Data understanding 38

4.2.1 General overview 39
4.2.2 Preliminary variable selection 39
4.3 Data preparation 41
4.3.1 Contact data filtering 41
4.3.2 Creating the label 42
4.3.3 Activity data filtering 43
4.3.4 Activity data aggregation 43
4.3.5 Final variable selection 44
4.4 Model building 45
4.4.1 Aggregation 1 46
4.4.2 Aggregation 2 47
4.4.3 Aggregation 3 48
4.4.4 Aggregation 4 49
4.4.5 Aggregation 5 51
4.5 Model evaluation 52
4.5.1 Choosing an aggregation method 52
4.5.2 Model comparison 53
5. Discussion 56
5.1 Visual analysis 56
5.2 Empirical study discussion 61
6. Conclusion 63
6.1 Research questions 63
6.2 Future research 64
7. Svensk sammanfattning 65
7.1 Metod och empirisk studie 66
7.2 Diskussion och slutsats 67
References 70
Appendix 72

1. Introduction

The purpose of this chapter is to familiarize the reader with the subject of the thesis. In chapter 1.1 the core problematics are introduced. In chapter 1.2 manual lead scoring is described and its problematics are discussed. Chapter 1.3 aims to explain what predictive analytics and machine learning are and how they can be used in the context of lead scoring. In chapter 1.4 the research questions, which the thesis aims to answer, are stated.

1.1 Background

Ever since the dawn of IT, companies have used the surge of available data to improve their businesses. In an article from Harvard Business Review (2012), Andrew McAfee and Erik Brynjolfsson claim that *"Data-driven decisions tend to be better decisions. Leaders will either embrace this fact or be replaced by others who do."* Using data to solve business problems and to support business decisions is going to become the norm in the future.

Some of the most critical business decisions relate to customer acquisition. For example, decisions regarding which customers the sales department should focus on. During the acquisition phase of the customer life cycle companies try to convert leads into customers through different methods, for example by calling them or sending them an email. However, not all leads are created equal, as some are more likely to become customers than others. Companies generally do not want to waste time on the so-called bad leads since salespeople are expensive. This leads to the question: how should companies distinguish between good and bad customer leads?

Furthermore, companies want to know the reasons why customers engage with them so that they can make better marketing decisions. Companies spend varying amounts of revenue on advertising each year, and yet the customer journey may seem like a black box to

most of them. Knowing what marketing channels, customer touch points and customer journey paths lead to successful sales would allow companies to optimize their spending by affecting the root cause of what leads to a sales transaction.

1.2 Manual lead scoring

Lead scoring is used to guide companies in prioritizing which leads to target. As a starting point, companies may choose to start scoring leads according to the data they have on them. For example, if a contact has visited the website, they may be awarded 5 points, but if the contact has sent an email, that contact might be given up to 25 points. The idea here is that salespeople should only spend their time on contacts that have a high lead score, which, assuming a reliable scoring procedure, implies that they will also have a high sales conversion probability.

However, there is an apparent problem in scoring people's behavior based only on gut feeling and purchase process knowledge. Marion (2016) describes some of the central issues with manual lead scoring. The first and foremost issue is that manual lead scoring lacks access to proper statistical support. Setting the right value for an activity relies on having access to behavioral data as well as demographics or firmographics data. In manual lead scoring, the scores may be assigned based on a set of fixed rules.

Activity	Points
Form/Landing Page Submission	+ 5
Submitted "Contact Me" Form	+25
Received an Email	0
Email Open	+1
Email Clickthrough	+3
Registered for Webinar	+3
Attended Webinar	+10
Downloaded a Document	+5
Visited a Landing Page	+2
Unsubscribed from Newsletter	-2
Watched a Demo	+8
Contact is a CXO	+5
Visited Trade Show Booth	+3
Visited Pricing Page	+10

Figure 1.1. An example of a manual lead scoring matrix (Marion, G. 2016).

The result is a scoring matrix which can be observed in Figure 1.1. (based on Marion (2016)) and is based on an experiment consisting of 800 leads scored according to manual lead scoring. They found no statistical difference between being able to convert scored leads that were determined "ready for sales" and randomly choosing leads that were not scored at all. Marion (2016) asserts that there is absolutely no way that someone without experience in statistics could score or weigh these activities properly. He claims that it is a very time-consuming process to always keep adjusting the scores and that the time used could be spent more effectively elsewhere. Bohlin (2017) also claims that manual lead scoring is not a recommended approach, even if rules and weights developed through assumptions are used together. According to her, all the recommended approaches to lead scoring should utilize some kinds of data-driven, mathematically intensive methods.

In the modern world companies are constantly evolving, which means that the lead scoring model needs to be able to adapt to these changes. Process changes, asset acquisition and the emergence of new buying behaviors are just a few examples of events that warrant changes to the lead scoring model. These kinds of events may necessitate a complete overhaul of the current lead scoring system, and, keeping in mind that manual lead scoring is a long process, this is also one of the main reasons why companies abandon manual lead scoring. (Marion, G. 2016)

The amount of data required to create an accurate model is huge. For example, while firmographic and demographic data should be included in the model, manual lead scoring is not able to incorporate all this information. Manual scoring is only able to use data that the marketer is able to perceive. This means that, for example, the customer's behavior during purchase decisions may be completely lost in the model. Often B2B decisions are made by a purchase team that consists of several individuals, but contact-based lead scoring does not reflect the whole group's opinions. There are several parties at play when a B2B decision is made and manual lead scoring is not able to account for the company politics (Marion, G. 2016).

Companies generate a tremendous amount of data, and this means one should not have to rely on gut feeling or business intuition when implementing a lead scoring solution. Instead, one should aim to make data-driven decisions, however doing so is easier said than done. The data is often large and complex enough so that the human mind cannot extract insights from it. However, a computer could be used to analyze the data. This calls for the usage of a data scientist's toolkit, or more specifically in the case of lead scoring, predictive analytics.

1.3 Predictive analytics

Artun and Levin (2015) describe predictive analytics as *"an umbrella term that covers a variety of mathematical and statistical techniques to recognize patterns in data or make predictions about the future"*. In the case of lead scoring, mathematical and statistical

techniques are used to find patterns in the data to estimate the likelihood of a lead turning into a sale. Artun and Levin (2015) mention that in addition to machine learning, predictive analytics techniques could also be called data mining, artificial intelligence or pattern recognition.

When predictive analytics is applied to the purpose of scoring leads, it is part of what Artun and Levin (2015) call predictive marketing. Predictive marketing is, according to Artun and Levin (2015), a customer-centric marketing approach that aims to enrich the customer's experience throughout the customer life cycle. The approach was developed due to the assumption that customers nowadays expect a tailor-made experience when interacting with companies. This experience is made possible due to the availability of technology that captures data previously inaccessible to the everyday marketer. Another factor that contributes to the success of predictive marketing is the dramatic decrease in computing costs. This is a crucial aspect as the tools and techniques used in predictive marketing can be very computationally expensive (Artun, O., Levin, D. 2015).

As mentioned before, predictive analytics is best described as an array of techniques used to generate insights from data. These techniques often take the form of mathematical/statistical algorithms, or machine learning algorithms. These algorithms are often sorted into three categories: supervised, unsupervised and reinforcement learning. Supervised learning algorithms estimate an output from the input, for example by estimating the likelihood that a customer will engage with a company. Unsupervised learning algorithms try to find patterns in data without having an explicit output, for example by looking at a customer base and finding customer groups that are different from each other. Reinforcement learning algorithms look for hidden patterns in the data to recommend the next best action. A reinforcement learning algorithm could be used, for example, to recommend products or other content to customers based on the whole customer base's purchase history or other preferences (Artun, O., Levin, D. 2015).

The purpose of lead scoring is to obtain a value that describes the likelihood of a customer lead turning into a sale by using data on the customer. In this process, the input is the data, and the output is the value representing the customer's lead-to-sale conversion probability. Thus, supervised learning algorithms are suitable for this task. However, other

types of algorithms might also be able to assist in more complex lead-scoring cases. For example, if one aims to perform lead scoring on a customer segment, one could use unsupervised learning first to create the segments.

1.4 Research questions

The primary purpose of this thesis is to study how machine learning can assist in the lead scoring process, both in B2C and B2B contexts. This entails acquiring and preprocessing the data using different kinds of data manipulation techniques. Afterwards, the data is used in training and comparing different machine learning algorithms to estimate the lead score. Furthermore, the secondary purpose of this thesis is to analyze the lead scoring results and attempt to uncover business insights, such as the importance of different customer touch points, customer journey paths and what kinds of customers to target.

As such, a couple of research questions can be identified. The overarching research objective of the thesis is to understand *"How machine learning can be used to perform lead scoring"*. This thesis expects to find answers to the following, more specific research questions in order to tackle the main objective:

- *Which machine learning algorithm gives the best performance in lead scoring?*
- *What business insights are to be derived from the lead scoring results?*

To answer the research questions the thesis will include a literature review and an empirical study where a client of ID BBN's data will be used to create several lead scoring models to test various machine learning algorithms. The models will be evaluated and the best one will be examined to see whether it offers any further business insights on top of accurate predictions.

2. Literature review

This chapter aims to provide an adequate understanding of lead scoring and machine learning to support the completion of the empirical study. The first part of the chapter will tackle lead scoring while the second will focus on machine learning concepts and algorithms.

2.1 Lead scoring

Lead scoring is a subtask of customer relationship management (CRM). The concept of lead scoring could be explained as calculating and assigning a lead score to a company's contacts. The score is calculated from characteristics data or behavioral data. Characteristics data includes variables such as industry, company size and responsibility level of the contact. Behavioral data encompasses website visits, contact history and the type of request if such a request was submitted (Benhaddou, Y., & Leray, P., 2017).

A higher lead score implies that the contact, or lead, is more likely to engage with the company. Benhaddou and Leray (2017) elaborate on this by claiming that the score could also reflect the position of the lead in the purchase cycle. This score allows companies to prioritize their sales by engaging with customers with a high lead score. Furthermore, Benhaddou and Leray (2017) claim that the score could also be used to personalize marketing actions. Michiels (2008) suggests that high priority leads should be passed on to sales and low priority leads should be engaged in lead nurturing campaigns.

There is a large variety of different CRM tools and systems available. According to Benhaddou and Leray (2017), the tools are designed to optimize digital channel integration to enable effective data collection as well as to be able to carry out, target and customize campaigns. Rosenbröijer (2014) mentions a few common CRM software companies, including Microsoft, Oracle, Siebel, Baan and Salesforce.

2.1.1 The importance of customer attributes in lead scoring

In a study by Aberdeen, Michiels (2008) showcases the importance of explicit and implicit attributes in lead scoring models. He explains implicit attributes as such attributes that are obtained from contact behavior, such as website visits, while explicit attributes include attributes that are obtained from the customer's own input, for example survey questions.

Table 2.1 shows the importances of explicit attributes in scoring models and Table 2.2 describes the importance of implicit attributes in lead scoring models based on a number survey. The tables include the percentage of participants in the study who did not use the attribute in their scoring models (Michiels, I. 2008).

Ranking Key				
1	2	3	4	5
Not Important				Very Important
			Median Rank (Importance to Scoring Model)	Percent Not Using Attribute in Scoring Model
Survey questions (purchase decision, budget, etc.)			5.0	5%
Sales activity (discussions, voicemails, etc.)			5.0	13%
Demographic profiles			5.0	0%
Profile data from prospects (landing page, forms, etc.)			4.5	10%
Comparison to profile information from existing customers			4.2	24%
Quality of lead's contact information			4.0	19%

Table 2.1. Explicit attribute importances (Michiels, I. 2008, p. 8)

Ranking Key				
1	2	3	4	5
Not Important				**Very Important**
			Median Rank (Importance to Scoring Model)	**Percent Not Using Attribute in Scoring Model**
Webinars attended			5.0	19%
Purchase propensity scores			4.5	14%
Email click-throughs			4.0	14%
Website activity (pages visited and recency)			4.0	24%
Website activity (type of activity)			3.5	24%
Keywords clicked			3.5	24%
Website activity (length of time each page was visited)			3.0	29%
Attitudinal and lifestyle information			3.0	20%

Table 2.2. Implicit attribute importances (Michiels, I. 2008, p. 9)

It is apparent that the explicit variables are included more often than implicit ones. The same observation is noted by Michiels (2008), who suggests that a reason for this could be that companies often experience difficulties in incorporating implicit attributes into their lead scoring models. The study by Aberdeen (Michiels, 2008) showed that the best performing companies usually included three or more implicit attributes in their lead scoring model. In addition, the highest performing companies had a more complex scoring model than others (Michiels, 2008).

2.1.2 Lead scoring conversion rate enablers

In the study by Aberdeen, five metrics are identified as important factors for effective lead scoring. These metrics are called process, organization, knowledge, technology and performance. There was a significant difference in the metrics between the high-performing and low-performing companies in the study (Michiels, 2008).

The process metric measured whether companies allowed leads to pass between marketing and sales. In addition, the structural quality of business functions was examined to see if they had proper processes that allow for high priority leads to be contacted first (Michiels, 2008).

The organization metric measured whether marketing was held accountable for the quality of the leads that are given to the sales staff. Furthermore, it measured whether individuals were accountable for lead management technique optimization (Michiels, 2008).

The knowledge metric measured whether companies had sales and marketing staff that had a common view on customer data and shared their definition of what a lead and qualified lead were (Michiels, 2008).

The technology metric measured whether the companies had systems that support lead management. These systems include a customer database, a lead management or demand generation, customer segmentation and targeting as well as lead scoring and prioritization tools (Michiels, 2008).

The performance metric measured whether companies were able to produce periodic dashboards or reports used to identify conversion rates and lead activity. In addition, it measured whether marketing campaigns were linked to successful sales and there were either weekly or bi-weekly reviews of the sales and marketing pipelines (Michiels, 2008).

Michiels (2008) recommends that companies looking to increase their lead conversion performance meet the requirements specified by the metrics used in the study. Furthermore, he recommends including implicit attributes and increasing the complexity of the lead scoring model. He also points out the importance of keeping the model up to date by continuously optimizing it. Michiels notes that the best performers in the study were able to interact quickly with new leads, regardless of how the lead interacted with the company. They were also more likely to use advanced automated tools for lead scoring.

2.2 Machine learning

As mentioned in the introductory chapter, supervised learning methods are a category of machine learning methods that can be applied to estimate a set outcome variable. In the case of lead scoring, the outcome to be estimated is the likelihood that a contact will purchase a product.

Supervised learning methods can be separated into two different categories: classification and regression methods. The difference between these two types is that regression is used to estimate continuous values and classification is used to estimate categorical outcomes. Furthermore, classification models have two outputs, one is a value between 0 and 1, representing the likelihood that an example belongs to the determined category, and the other is the discrete category (Kuhn, M., & Johnson, K., 2013).

Lead scoring can be treated as a classification problem. Thus, the focus of this chapter is to provide a basic understanding of applying supervised learning methods to solve classification problems.

2.2.1 Classification tree models

Classification trees are part of a family of tree-based models. They consist of nested if-then statements derived from the variables found in the data set. One example of a simple classification tree can be seen in Figure 2.1. (Kuhn, M., & Johnson, K., 2013)

```
if Predictor B >= 0.197 then
|   if Predictor A >= 0.13 then Class = 1
|   else Class = 2
else Class = 2
```

Figure 2.1. Simple classification tree (Kuhn, M., & Johnson, K., 2013, p. 369)

In this figure, three terminal nodes and two splits can be identified. The terminal nodes, or leaves, are the nodes that conclude the tree and contain the outcome. Possible outcomes in Figure 2.1 include "Class 1" and "Class 2".

The splits found in Figure 2.1 are the points where the decision tree makes decisions. The first split checks if Predictor B is larger than or equal to 0.197, and if it is then the decision tree checks if Predictor A is as large or larger than 0.13. If Predictor A is larger than 0.13, then the decision tree will arrive at the terminal node which concludes that the result will be "Class 1" (Kuhn, M., & Johnson, K., 2013).

The process how the decision tree decides where to split the tree and what rules to use depends on different kinds of optimization criteria. Generally, the tree splits the data into smaller groups of homogeneous data, which in this case means splitting the data into groups that contain a larger proportion of data points belonging to one class than the other. Some criteria that could be used include accuracy, Gini index and cross entropy. The splitting process continues until the criterion is minimized, increasing the depth of the tree. Alternatively, if a maximum tree depth limit is set, the splitting process stops upon reaching that limit (Kuhn, M., & Johnson, K., 2013; Gokgoz, E., & Subsi, A., 2015).

Pruning is applied to combat over-fitting the decision tree. Simply put, pruning removes poorly performing branches of the decision tree. If the performance increase caused by the inclusion of a branch does not meet the set confidence level set, it is pruned (Kuhn, M., & Johnson, K., 2013).

The strength of decision trees lies in their high interpretability, capability to process several different types of predictors and even the ability to handle missing data. Weaknesses include instability in the final model and arguably low general predictive effectiveness (Kuhn, M., & Johnson, K., 2013).

2.2.2 Random forest

Random forest algorithms attempt to alleviate the classification tree algorithm's variance or instability problems through a modified version of bagging. Bagging is a method applied to reduce variance of estimated prediction functions. In essence, random forest creates several de-correlated decision tree models and calculates their average as the basis of predicting the output (Hastie, T., Tibshirani, R., & Friedman, J. 2017; Chen, W., Xie, X., Wang, J., Pradhan, B., Hong, H., Bui, D. T., ... & Ma, J. 2017).

The general random forest algorithm was created by Breiman in 2001 after evaluating different approaches to introducing randomness to the tree construction process. The reason why the element of randomness was introduced was to increase the decision tree model's performance through de-correlation. The pseudo-code for a general random forest algorithm can be seen in Figure 2.2. (Kuhn, M., & Johnson, K., 2013).

```
Select the number of models to build, m
for i = 1 to m do
    Generate a bootstrap sample of the original data
    Train a tree model on this sample
    for each split do
        Randomly select k (< P) of the original predictors
        Select the best predictor among the k predictors and
        partition the data
    end
    Use typical tree model stopping criteria to determine when a
    tree is complete (but do not prune)
end
```

Figure 2.2. Random forest (Kuhn, M., & Johnson, K., 2013, p. 200)

There are a few parameters that can be adjusted in the random forest algorithm. One of these parameters is the amount of randomly selected predictors to include in any given tree. For classification purposes, it is recommended to pick the square root of the number of predictors as the number of randomly selected predictors to choose. Another parameter to be

set is the number of decision trees that should be built. The algorithm is unable to over-fit the model, so increasing the number of trees to be built will not reduce the performance of the model. A good number of trees to build initially is 1000 (Kuhn, M., & Johnson, K., 2013).

Predictor importance is calculated by aggregating the improvement in performance for each predictor. This performance is calculated in different ways depending on the chosen criterion, one of which is the Gini index (Kuhn, M., & Johnson, K., 2013).

Some properties of random forests include insensitivity to different values of randomly selected predictors, minimal pre-processing requirements and ability to calculate out-of-bag performance measures (Kuhn, M., & Johnson, K., 2013).

2.2.3 Logistic regression

Logistic regression models predict the likelihood that an input belongs to a specific class in a binary classification problem (i.e. two possible output classes). This means that the output needs to stay between 0 and 1 for both classes and that their sum needs to equal 1. The output can be interpreted as the probability that estimates the likelihood of an outcome.

Logistic regression is a simple yet popular model which belongs to a family of generalized linear models. It can also be used to make inferential statements about the predictors used in the model, such as assessing whether a predictor has a statistically significant relationship to the probability of the outcome (Kuhn, M., & Johnson, K., 2013; Dreiseitl, S., & Ohno-Machado, L., 2002).

The model is often applied in biostatistical applications where there are two possible outcomes, or classes. Examples of these types of classification problems include determining whether the patient has an illness or not (Hastie, T., Tibshirani, R., & Friedman, J. 2017).

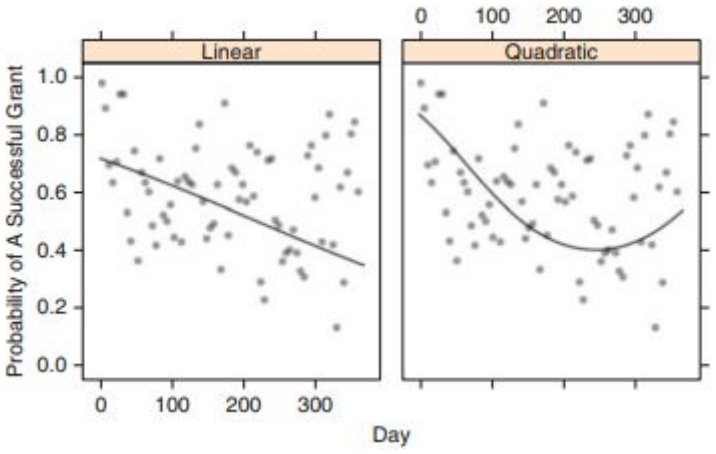


Figure 2.3. Comparing linear and quadratic models (Kuhn, M., & Johnson, K., 2013, p. 284)

As an application of this model, Kuhn and Johnson (2013) test logistic regression in a case where grant admission probability is estimated. This case has two possible outcomes or classes, which is why it can be modeled using a binomial distribution. The logistic regression model learns from the training data and attempts to separate the two classes using the supplied predictors. In the example model, the day of the year was used a parameter. The initial model performed poorly, however after adding additional parameters, such as the squared day, the model's performance improved. The left plot on Figure 2.3 shows the decision boundary using only one predictor and the right plot depicts the boundary after adding the squared day predictor (Kuhn, M., & Johnson, K., 2013).

However, in this case (and in general) adding more variables did not necessarily mean that the model performance keeps increasing. When all variables were used, Kuhn and Johnson (2013) noticed that some predictors had data points in the extremes of the distribution. This caused a decrease in model effectiveness, and when these predictors were excluded the model performed better (Kuhn, M., & Johnson, K., 2013).

2.2.4 Neural networks

Neural network is a general term that includes several non-linear algorithms and models. The most common and simplest neural network algorithm uses back-propagation and has a single hidden layer. The term "neural network" has its roots in attempting to model the human brain. The units represent neurons and the connections between units represent synapses. Neural networks can handle both regression and classification problems. The network functions by finding values for its unknown parameters, called weights, that will fit the training data well (Hastie, T., Tibshirani, R., & Friedman, J. 2017).

As seen in Figure 2.4, neural networks are comprised of several connected units in different layers. The top layer depicted in Figure 2.4 is called the input layer which consists of all predictor variables. The hidden layer contains hidden units, each of which contain a linear combination of the values of predictor variables. In calculations, logistic or sigmoidal, function is applied to these values to keep them between 0 and 1. The bottom layer represents the output layer with the possible output classes. As the final output of the algorithm, the class with the highest value becomes the predicted class. The sigmoidal function is applied again on the units in the output layer. Additionally, a softmax transformation is performed to tweak the output values so that they represent percentages (Kuhn, M., & Johnson, K., 2013).

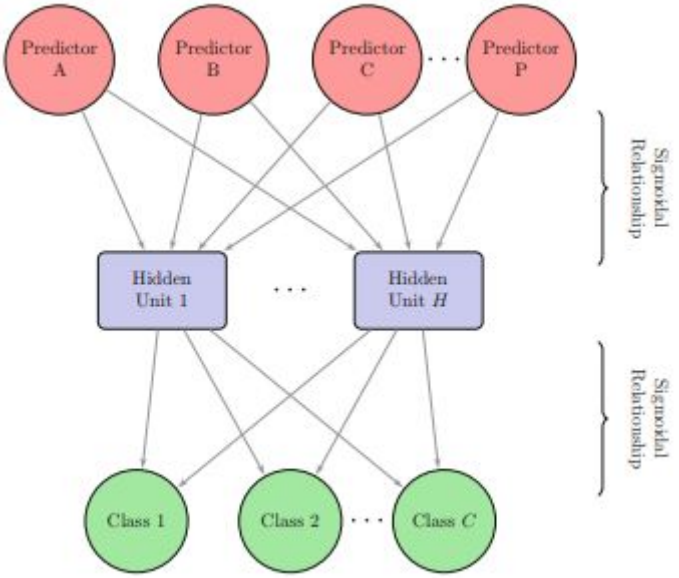


Figure 2.4. Classification neural network with 1 hidden layer (Kuhn, M., & Johnson, K., 2013, p. 334)

Neural network models attempt to optimize the sum of the squared errors across all samples and classes to increase the model's performance (Kuhn, M., & Johnson, K., 2013; Wang, L., Zeng, Y., & Chen, T. 2015). This is usually achieved by applying the highly efficient back-propagation algorithm which utilizes derivatives to locate the optimal parameters (Kuhn, M., & Johnson, K., 2013). Learning rate adjusts how large the changes to the weights are when adjusting the weights of the network (Hastie, T., Tibshirani, R., & Friedman, J. 2017).

Unlike the random forest algorithm, neural networks are incredibly prone to over-fitting. Weight decay, which reduces the size of the parameter estimates, can be used to combat over-fitting. In addition, it is recommended to combine nonlinear classification models, such as neural networks, with feature selection tools, as neural networks can be negatively affected by predictors that lack predictive power (Kuhn, M., & Johnson, K., 2013).

Another way to combat over-fitting is to set a minimum error to reach. This early stopping method prevents over-fitting by stopping the model once it reaches the set minimum error. Scaling the inputs has an effect on the resulting model, as does modifying the number of hidden units and layers. Too many hidden units could decrease model performance, and

too few will prevent it from developing properly (Hastie, T., Tibshirani, R., & Friedman, J. 2017).

2.2.5 Missing data

When a value is not present in a data set, it is a missing value. A value could be structurally missing, for example if a variable described the number of births given but the data set included both sexes, all males would either have missing values or the value 0. If these values were missing, they would be structurally missing. (Kuhn, M., & Johnson, K., 2013)

Understanding why data is missing is important. If a missing value is related to the outcome, it is called informative missingness, and it may cause the model to be significantly biased. For example, customers often rate products when they either love or hate it, causing ratings to be populated by either very high or very low values and few middle values. This is informative missingness because customers who would have rated the product as average simply chose to not rate it at all (Kuhn, M., & Johnson, K., 2013).

In addition to missing data, there is also censored data. Data is censored when the value is not missing, but the value is not correct either. For example, if a company that rents movies has a variable that measures the time that a customer has rented a movie then the variable's data would be censored for those rented movies that have yet to be returned. This is because it is not possible to know exactly how long the movie will be rented, but it is at least up to the current date. In predictive modeling, censored data is often treated as either missing data or as the censored data value (Kuhn, M., & Johnson, K., 2013).

Missing data can be found in specific variables or in subsets of the data. If a large amount of values in a variable are missing, it may be a good idea to exclude it. In cases where missing values occur in a subset of the data set, it is a common practice to consider the size of the data set. In large data sets it may be a good decision to exclude the examples with missing values, however in smaller data sets removing the examples is usually a bad idea (Kuhn, M., & Johnson, K., 2013).

Instead of removing examples, one may choose to utilize missing value imputation. Imputation is the act of replacing missing values in data. This is often done by replacing the missing values with ‘estimated’ values using assumptions that are based on other variable values. Because imputation adds a degree of uncertainty to the model, it should also be performed in the resampling or cross-validation process to mitigate the uncertainty. As the most frequently used procedure, the K-nearest neighbor algorithm can be trained to impute missing values based on other variable values. Research results by Batista and Monard (2002) show that it is an effective imputation method even with large amounts of missing data (Kuhn, M., & Johnson, K., 2013; Liu, Z. G., Pan, Q., Dezert, J., & Martin, A., 2016).

2.2.6 Data splitting and resampling

When working with a data set, there is a given number of examples or data points to work with. It is important to note that when evaluating the performance of a model one should use examples that have not been used in the creation of the model. This is done to obtain an unbiased performance evaluation. The set used to train the model is called the "training" data set and the data set used to estimate the model's performance is called the "test" or "validation" data set. Splitting the data into these sets can be done by taking random samples from the original set (Kuhn, M., & Johnson, K., 2013).

Over-fitting occurs when a model has learned, in addition to the general patterns in the data, the individual noise present in the training examples. This causes the model to become worse at predicting new, unseen examples, since it is tailor-made to predict the examples that were included in the training phase (Kuhn, M., & Johnson, K., 2013). Resampling techniques are applied to avoid bias and provide an effective performance evaluation when training a supervised learning model. They are often used to effectively estimate a model's performance. Common resampling techniques include k-fold cross-validation and bootstrapping (Kuhn, M., & Johnson, K., 2013).

K-fold cross-validation is a procedure that splits the data into a set number of samples that are roughly equal in size. Every sample except one is used to train the machine learning

model and the performance is then tested on the excluded sample. This process is then repeated until every sample has been excluded and tested against. The performance of each model is summarized, often using mean and standard deviation, to gain an understanding on its general performance (Kuhn, M., & Johnson, K., 2013).

Bootstrapping is a technique that selects random examples from the data set but does not exclude an example after it has been selected once. The result is that some examples are represented more than once while some can be completely excluded. The resulting data set is called a bootstrap and it is as large as the original data set. A model is created using the bootstrap, which is then tested against a sample set consisting of all examples not included in the bootstrap. As with k-fold cross-validation, the performance of each created model is summarized (Kuhn, M., & Johnson, K., 2013).

2.2.7 Evaluation metrics

As their output, classification models generate the predicted class and a value between 0 and 1 which represents the probability that the example belongs to that class. In most cases, for example in e-mail spam filtering, the predicted class is used in further decision processes instead of the probability. The probability estimates can be used to estimate the model's confidence about the prediction. An example of this would be when an insurance company investigates fraudulent claims, where they would combine the probability of fraud with the cost of the investigation and other monetary losses to see whether further investigation is profitable or not (Kuhn, M., & Johnson, K., 2013).

One way to describe the performance of a model is through a confusion matrix, depicted in Figure 2.5. The confusion matrix presents the observed values as columns and predicted values as rows. Diagonal cells are values that were correctly predicted, written as TP and TN, and the off-diagonal cells denote the cells that were wrongly predicted, written as FP and FN. TP stands for true positive, TN for true negative, FP for false positive and FN for

false negative (Kuhn, M., & Johnson, K., 2013; Luque, A., Carrasco, A., Martín, A., & de las Heras, A. 2019).

Predicted	Observed	
	Event	Nonevent
Event	TP	FP
Nonevent	FN	TN

Figure 2.5. Confusion matrix for 2-class problems (Kuhn, M., & Johnson, K., 2013, p. 254)

Performance metrics are used to assess how well the model performs. Accuracy, which represents the percentage of correct predictions made, is one of the simplest and widely used performance metrics, but it has some disadvantages. It does not distinguish between error types, which is important if some errors are more important to avoid than others. In addition, accuracy does not consider class imbalance. For example, if a dataset has 10 examples with class 1 and 1000 examples with class 2 the model could predict every class to be class 2 and achieve a very high accuracy (Kuhn, M., & Johnson, K., 2013).

There are several alternative performance measures which allow for considering one class as more important than the other. Sensitivity and specificity consider which class is more interesting. Sensitivity is calculated as the number of correctly identified sample from the more important class, divided by the total number of samples from the more important class. Specificity is calculated as the number of correctly identified samples from the less important class divided by the total number of samples from the less important class. Sensitivity rises when the number of predicted examples from the more important class rises, but this often causes specificity to decrease. If predicting one class is more important than the other, these metrics can be used to model potential trade-offs. Sensitivity is also known as true positive rate and the false positive rate is calculated by subtracting the specificity value from 1 (Kuhn, M., & Johnson, K., 2013).

One can also utilize a metric that combines both specificity and sensitivity. This metric is called Youden's J Index, and it adds sensitivity and specificity and subtracts it by 1 (Kuhn, M., & Johnson, K., 2013).

Another more common way to summarize the magnitude of errors is by using the receiver operating characteristic (ROC) curve. This method plots the true positive rate (TPR) against the false positive rate (FPR) across different probability thresholds. An example ROC curve is depicted in Figure 2.6, where the threshold for determining the class is 50% at the red dot and 30% at the green square. The threshold determines how sure the machine learning model needs to be to assign the more important class to an example (Kuhn, M., & Johnson, K., 2013; Carter, J. V., Pan, J., Rai, S. N., & Galandiuk, S., 2016).

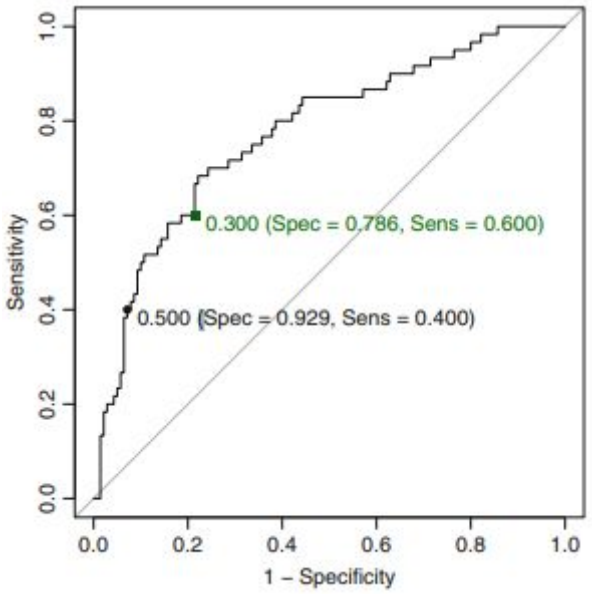


Figure 2.6. Receiver operator characteristic (ROC) curve (Kuhn, M., & Johnson, K., 2013, p. 263)

If the curve's trajectory is steep, as shown by the dot marker in Figure 2.6, sensitivity is increasing at a larger rate than the decrease in specificity. Specificity starts decreasing more at around 0.7 sensitivity. Using this method makes it possible to maximize the trade-off between sensitivity and specificity (Kuhn, M., & Johnson, K., 2013).

Another use of the chart is to calculate the area under the curve (AUC). A perfect model would have 100% sensitivity and specificity, meaning the area under the curve would be 1.0, filling the whole graph. A useless model would draw a line diagonally through the graph, having an area under the curve of 0.5. The closer the curve is to the upper left corner of the plot, the better the model would be. A higher area under the curve is also indicative of a better model. It is possible to compare the ROC curves of different models by plotting them on the same graph and calculating their area under the curve (Kuhn, M., & Johnson, K., 2013; Baker, A. M., Hsu, F. C., & Gayzik, F. S. 2018).

An advantage of ROC curves is that it is insensitive to class imbalances. This is due to the ROC curve being a function of sensitivity and specificity. A disadvantage is that it obscures information. When comparing ROC curves, it is often the case that no curve is strictly better than another. Another thing to note is that sometimes specificity and sensitivity are valued differently, meaning different areas of the curve are of higher importance (Kuhn, M., & Johnson, K., 2013).

Lift charts are used to visualize the model's capacity to sort the examples into the correct class. When the probability scores are sorted in descending order, a model that is effective at ranking the examples would calculate higher probability scores for the positive class than the negative class. A perfect model would mean that the positive classes would all have higher scores than the negative classes. Lift is a metric that represents the amount of correctly classified examples over random guessing. The chart creates a line by plotting the cumulative gain, or lift, and the cumulative percentage of screened samples. Figure 2.7 depicts lift curves made on a data set with an even distribution of classes. A perfect model would generate the green lift curve and the purple curve represents a random choice (Kuhn, M., & Johnson, K., 2013; Tamaddoni, A., Stakhovych, S., & Ewing, M. 2016).

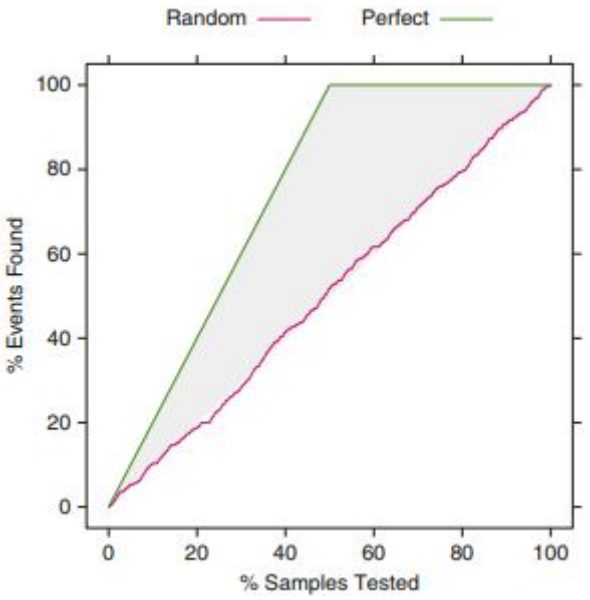


Figure 2.7. Lift plot (Kuhn, M., & Johnson, K., 2013, p. 266)

2.2.8 Handling class imbalance

Class imbalance occurs when one or more classes have a lower proportion than the other classes in the (training) data. For example, class imbalance may occur in online advertising, where each example is an advertisement shown to a viewer and the two classes correspond to viewers who (i) clicked and (ii) did not click on it. In a recent study, the percentage of clicked ads was about 2.4% (approximately 1 in 40 cases), which demonstrates how severe class imbalance can become. Class imbalance often leads to highly skewed results, meaning the model may predict a very small proportion of the minority class correctly. The probability scores for the minority class are often very low as well (Kuhn, M., & Johnson, K., 2013).

There are different ways to handle class imbalance. One of these ways is to simply tune the model to maximize the accuracy of the minority class or the model's sensitivity. Another way is to modify the threshold for classifying the minority class, resulting in a trade-off between sensitivity and specificity. This can be accomplished with the help of the ROC curve. Determining the threshold can be done, for example, either by simply choosing the point that is closest to the top-left of the plot or by calculating Youden's J index for each

threshold and selecting the threshold that has the highest value. When determining the new cutoff point, it is important to use an independent dataset since using the training data would likely lead to optimistic bias (Kuhn, M., & Johnson, K., 2013).

Sampling methods can be used to manipulate the training data to represent an equal distribution of classes. However, it is important that the test set still representative of the actual state of the data. Down-sampling and up-sampling are techniques that can be used to improve class balance. In general, up-sampling creates additional data points from the minority class and down-sampling removes data points from the majority class (Kuhn, M., & Johnson, K., 2013).

There are several different kinds of sampling techniques, one of the most widely used being the synthetic minority over-sampling technique (SMOTE). This technique uses up-sampling and down-sampling to balance the classes. To up-sample the minority class, the technique chooses a random example from the minority class and determines its K-nearest neighbors. K-nearest neighbors is a method that determines a predetermined number of examples that resemble the target example the most. A new example is then created using a random combination of values found in the chosen example's and its neighbors' variables. Down-sampling of the majority class is also possible (Kuhn, M., & Johnson, K., 2013; Prati, C., Batista, G. E., & Monard, M. C., 2009; Buda, M., Maki, A., & Mazurowski, M. A., 2018)

2.2.9 Practical applications of machine learning on CRM

Ngai, Xiu and Chau (2009) discuss a collection of literature regarding the application of machine learning to customer relationship management. They split the customer journey into four stages and identify seven types of machine learning, or data mining, methods used in the literature. The four stages of the customer journey are customer identification, attraction, retention and development. The machine learning methods are association, classification, clustering, forecasting, regression, sequence discovery and visualization. The most common machine learning algorithms used include association rule, decision tree, genetic algorithm, neural networks, K-nearest neighbor and linear as well as logistic regression (Ngai, E. W.,

Xiu, L., & Chau, D. C., 2009). In the following paragraphs some interesting and relevant applications from the literature are discussed.

Chen, Hsu and Chou (2003) use a decision tree algorithm to perform a target customer analysis for a tour company. The model is built to predict which tour any given customer may take. Since the company offered more than two tours, there was a need to create a model that could accommodate several output classes. The variables used in the model included the customer's marital status, income, gender and hobby. The resulting model was effective in predicting what tours the customer would take (Chen, Y. L., Hsu, C. L., & Chou, S. C. 2003).

Buckinx and Poel (2005) create models that predict customer loyalty in a non-contractual setting using random forest, logistic regression and neural networks. Logistic regression was included as a comparison point for the more advanced models. The neural network used was of the automatic relevance determination (ARD) type. One reason for using this type of neural network was to obtain variable importances. The random forest algorithm was used in lieu of a decision tree algorithm due to their robustness and greater performance. In addition, the random forest algorithm chosen was also capable of producing variable importances. The model was evaluated using accuracy and AUC. The model was successful in detecting future partial defection and there were no noticeable differences in the models created by the three algorithms. Partial defection means that the customer switches some of their purchases to another store. Findings suggest that the most important variables relate to the recency, frequency and monetary value of the customers (Buckinx, W., & Poel, D. V. D. 2005).

Kim and Street (2004) apply a genetic algorithm and an artificial neural network to maximize expected profit from direct mailing. The genetic algorithm is used to select different subsets of variables to pass on to the neural network, the results are evaluated, and the best subset is then chosen for the final analysis. This is done to minimize the number of variables to increase the interpretability of the neural network model, which potentially allows marketers to extract key drivers of consumer response. However, reducing the number of variables could lead to a decrease in accuracy. The method produced a model that considers campaign costs and profit per additional customer, maximizing the expected profit

and having a higher interpretability due to using a smaller set of features (Kim, Y., & Street, W. N., 2004).

3. Methodology

This chapter will discuss the applied methodology of the thesis. The thesis will apply a quantitative methodology as motivated by the nature of the chosen research problem. The other dominant research methodology, qualitative research, often has an emphasis on words and interpretations, trying to explain how or why something is occurring, whereas quantitative research focuses on numbers as it is the case in present research.

The discussion of selecting different types of research approaches is discussed in chapter 3.1. Chapter 3.2 presents Schmueli's and Koppius's (2011) schematic for building empirical models. In chapter 3.3 a slightly modified version of the schematic, which is applied in the empirical study, is described.

3.1 Methodology selection

A taxonomy of research approaches by Järvinen (2004) is shown in Figure 3.1. The figure depicts an attempt to categorize different types of studies based on methods and research questions typical for information systems research. Mathematical approaches strive to prove an assertion, for example a mathematical formula, to be true. Conceptual-analytical approaches study reality by using previous knowledge and apply it using logical reasoning. Theory-testing studies include the usage of experiments or field studies while theory-creating approaches aim to create a theory. Innovation-building approaches attempt to develop something new and the innovation-evaluating approach is applied when comparing the result to the goal. These types of research approaches are focused on the utility of the created innovations while the theory-focused approaches are meant to study reality (Järvinen, P. 2004).

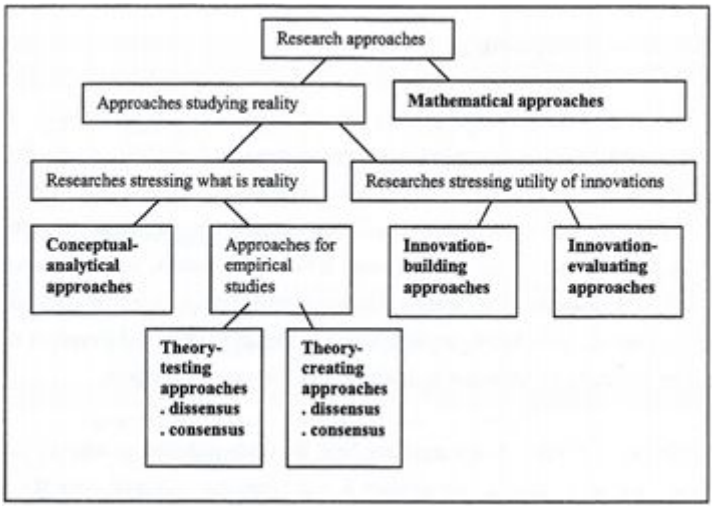


Figure 3.1. Taxonomy of research approaches (Järvinen, P. 2004)

The objective of this thesis is to study how machine learning can be used to perform lead scoring. This is done by seeking to answer the research questions through an empirical study, which is supported by a literature review. Thus, the thesis applies a theory-testing research approach. However, one could argue that the study applies an innovation-building or evaluating approach since it includes building and evaluating a machine learning model that scores leads.

According to Schmueli and Koppius (2011), predictive analytics is not often a part of mainstream information systems research. They claim that most modeling in information systems has been causal-explanatory statistical modeling. Machine learning is part of predictive analytics, and thus should be considered part of information systems as well. Schmueli and Koppius (2011) encourage researchers to use predictive analytics in future research.

One could argue that CRM is a large information system. Lead scoring is, in essence, an attempt to improve CRM functionality by prioritizing leads. Thus, lead scoring using machine learning is part of an arguably significant subject area within information systems.

3.2 Quantitative methodology

Schmueli and Koppius (2011) present their schematic for steps required to build a predictive model information systems research. The overall schematic can be used for building both exploratory and predictive models, however the steps themselves differ. The steps are depicted in Figure 3.2. (Shmueli, G., & Koppius, O. R. 2011).

Figure 3.2. Schematic of the Steps in Building an Empirical Model (Predictive or Explanatory) (Shmueli, G., & Koppius, O. R. 2011)

During the first step, a specific goal is defined. For the purposes of predictive analytics, a goal could be to predict an outcome from a set of observations. If the goal is a numerical value, it is called a prediction and if it is a categorical one, it is called a classification. Another type of goal, which is termed ranking, would be to rank the observations according to the probability that they belong to a certain class (Shmueli, G., & Koppius, O. R. 2011).

In the second, step the data collection and study design is determined. A choice between an experimental or observational setting is made. For predictions, observational data is preferred in case it represents the real-world situation better than experimental data. When considering the data collection instrument, it is important to ensure that the data used for both modeling and prediction contains the same variables and originates from the same population. For predictive analytics, the sample size should be larger than for explanatory modeling. This is due to a few reasons, such as higher uncertainty for predicting individual observations and the model being based on the data instead of theory. In addition, a larger sample size reduces the bias and sampling variance of the model, which is important since

predictive models are often used to capture complicated relations present in the data. A larger data set also allows for the use of holdout sets, which are needed when testing the final model performance. The dimensionality of the data is considered by justifying the existing variables. For predictive analytics, several variables are often introduced to capture their relationships. In predictive modeling, the hierarchical design of the data collection often entails concentrating on a specific population group instead of trying to capture a little bit of everything (Shmueli, G., & Koppius, O. R. 2011).

The third step is data preparation. Dealing with missing values and data partitioning is included in this step. Choosing how to treat missing values in data sets depends on whether the missing values are informative by themselves and whether the missing values are in both the training sample and the sample to be predicted. There are several methods for handling missing values, such as removing the examples or variables containing the missing values. Other methods include replacing them with other values, such as a value that indicates that the value is missing, or a value chosen by machine learning algorithms. Then the data is partitioned into a training and a holdout set. The training set is used to create the model and often includes a validation set. A model is chosen based on its performance on the validation set. The holdout set is used to evaluate the final performance of the model. (Shmueli, G., & Koppius, O. R. 2011).

In the fourth step exploratory data analysis is applied. This step includes the numerical and graphical summarization of the data, dimensionality reduction and outlier handling. Visualisation is used to gain more insights into the nature of the data. Dimensionality reduction is applied, often by using principal components analysis (PCA), to reduce the number of variables and sometimes increase predictive accuracy (Shmueli, G., & Koppius, O. R. 2011).

Choosing variables for modeling is done in the fifth step. When choosing variables, both the predictor and response variables are chosen based on their current availability and measurement quality. In addition, when choosing the response variable, the goal of the predictive model is kept in mind. Predictive models often include more variables than explanatory models to allow for the discovery of relationships between variables (Shmueli, G., & Koppius, O. R. 2011).

The sixth step includes choosing potential methods. The primary decision to be made here is to choose between data-driven algorithms, shrinkage methods or ensembles. Data-driven algorithms include models such as classification trees and neural networks. These models can capture complex relationships between variables. Shrinkage methods include techniques such as principal components regression and ridge regression. These models shrink the predictors, leading to potentially increased prediction accuracy but also increasing model bias. Ensemble methods use the average of several models to produce higher accuracy models. Random forest is an example of an ensemble model (Shmueli, G., & Koppius, O. R. 2011).

The seventh step includes evaluation, validation and model selection. During evaluation, the predictive accuracy of a model is measured by applying it to the holdout set. Model validation is done to reduce the risk of over-fitting, which means that the model learns too much from the training set leading to worse predictive accuracy on the holdout set. A model is selected by finding a model with the best balance between bias and variance leading to a high accuracy (Shmueli, G., & Koppius, O. R. 2011).

The last step includes model use and reporting. Different performance measures and plots are used to represent the model's effectiveness. For example, classification matrices and ROC curves could be included in the report. Over-fitting is discussed as well as comparing the model to simpler possible solutions (Shmueli, G., & Koppius, O. R. 2011).

3.3 Empirical study structure

The structure that will be implemented in the empirical study is a simplified version of the one described in chapter 3.2. The purpose of the simplification is to be able to convey the status of the project with greater ease.

The first stage focuses on the business objective. In this stage the business objective is clearly defined, and the client's business is analyzed. In the second stage, data understanding, the client's data is examined, and potential problems are detected and corrected. The data is

prepared in the third stage. During data preparation, several data transformation actions are taken. This includes dealing with missing values, transforming the data to be suitable for machine learning, feature extraction, example filtering and variable choice. The fourth stage is model building. During this stage models are built using different machine learning algorithms and aggregation methods. During the fifth stage, which is model evaluation, the models are evaluated, and the optimal model is identified. The sixth stage is model deployment. During the deployment stage, a report is made, and the model is prepared for deployment for the client. For the purposes of this thesis, the sixth stage will only include a report of the model and has therefore been moved to chapter 5.

4. Empirical study: AI assisted lead scoring

This chapter will present the details of the process of the empirical study. Chapter 4.1 analyzes the business problem of the client. Chapter 4.2 details the process of understanding the available data and choosing what data to include in the model. Chapter 4.3 describes the steps required for preparing the data into a format suitable for machine learning. In chapter 4.4, machine learning models are built using different aggregation methods. In chapter 4.5, one aggregation method is chosen and the different machine learning models using this type of aggregation are compared against one another. The best model is identified and later analyzed in chapter 5.

4.1 Business objective

In the project, the main task is specified as creating a lead scoring model for the client so that they can prioritize their salespeople's time. The result of this study will be a model that can identify a list of leads along with their lead scores so that the client can select which leads to contact. The client is an international company; however, this study will only include contacts from Finland in the empirical analysis.

The client's product is a large purchase decision. Their business encompasses both B2B and B2C transactions. At the starting stage of the modeling, the specific steps of modeling are not clearly identified as it may be necessary to train two different machine learning models, one for B2B and one for B2C, since the available data may differ.

The client's business is structured in such a way that once the client becomes aware of a lead, either through a web form or other means, they notify a local contractor based on where the potential buyer is located. The contractor then takes the matter into his or her own

hands and contacts the lead when they see fit. After a lead has been contacted, the contractor sends the lead's status back to the client, after which it is processed in their internal systems.

A customer may contact the client through the client's website. The landing page of the website allows the visitor to choose between B2C and B2B business, both of which have an easily accessible contact form directly available. This contact form requests basic information from the contact and allows the contact to fill in a message or question.

4.2 Data understanding

In this chapter, the contents of the data files provided by the client are checked and each attribute, or variable, is examined to determine which attributes will be used in the machine learning model construction. To accomplish this in an effective way, a spreadsheet detailing the properties of all attributes is created for each data file. Most of the data is available on the client's Eloqua server hosted by ID BBN. As a result, it is possible to pick and choose what kinds of variables to export from the Eloqua server.

The data is explored in RapidMiner. First, the data must be imported into the program. While importing the data, it is important to choose the correct encoding for the data file and the correct data types for the attributes. RapidMiner only looks at the first few examples when determining the type of the attribute, so it is crucial to examine each attribute and to manually check whether the assigned data type is correct. For example, if a data field contains customer IDs, which may only consist of numeric characters, the program may determine that the attribute's data type is integer instead of polynomial. If this happens, any of the customer IDs starting with the number zero will have their value altered due to the numerical data type dropping unnecessary zeros at the beginning of the value. This alters the data set significantly and will impose problems later down the line when joining different data sets.

4.2.1 General overview

Two data sets are used for this analysis. The main data set contains contact-level data from the client's internal systems. Examples of variables from this data include name, country, location, contact type, email address, the source of the lead and whether the lead has made a purchase. In addition to these variables, there are several variables which are not suited for machine learning. Reasons for variables not being suited for machine learning include having too many missing values or lack of predictive power. This data set is used as the foundation on which the machine learning model is built.

The other data set contains the contact's ID and activity data, including information such as website visits, email sends, email opens, email clickthroughs and form submits. The data set is structured in a way that is unsuitable for machine learning. To remedy this, data preparation steps need to be taken in order to aggregate the values without losing much of their predictive power.

Since the client's product is a large-scale purchase decision, some degree of class imbalance is to be expected. There are more contacts in the client's data set that have not made a purchase than contacts that have made a purchase.

4.2.2 Preliminary variable selection

The main data file includes several redundant or useless variables. This is to be expected since the data set has almost 200 variables. RapidMiner suggests the following criteria when deciding if a variable should be included in the model or not:

- Variables that correlate with the label
- Variables where most values are different
- Variables where most values are identical
- Variables with many missing values

During this step only those variables that are useless will be excluded. Final variable selection will be performed at a later stage. The reason for this is that some data preparation steps require the use of variables which will not be used in the final analysis. Another variable selection step will be performed after the data preparation stage. Table 4.1 describes the variable types chosen from the main contact data set (see Appendix A for RapidMiner process).

Type	Description
Identifier	Used to link the same contacts in different data sets
Location	Filtering by region
Marketing unit	Filtering by marketing unit location
Date created & modified	Filtering by time
Email address domain	Filtering by email domain
Contact status & time	Used to identify buyers and their purchase moment

Table 4.1. Preliminary variable selection

The activity data set does not contain redundant variables. However, the data set must be transformed into a format suitable for machine learning before it can be used. Table 4.2 shows the variables in the activity data before being transformed.

Type	Description
Contact	Identifies who did the activity
Activity	Specifies which activity was done
Date	Indicates when activity took place

Table 4.2. Activity data, raw format

4.3 Data preparation

As stated before, the objective of the research is to obtain a machine learning model that predicts the probability that a customer lead will turn into a sale. Consequently, it is not feasible to use all the available data from customers who have made a purchase in the past. For those customers who have purchased something in the past, only the data before the moment of purchase should be used. This is done so that the machine learning algorithm learns the patterns leading up to a purchase instead of learning the behavior of an already existing customer. The process flowchart depicting the steps taken from the data importation stage to the model creation stage is shown in Figure 4.1.

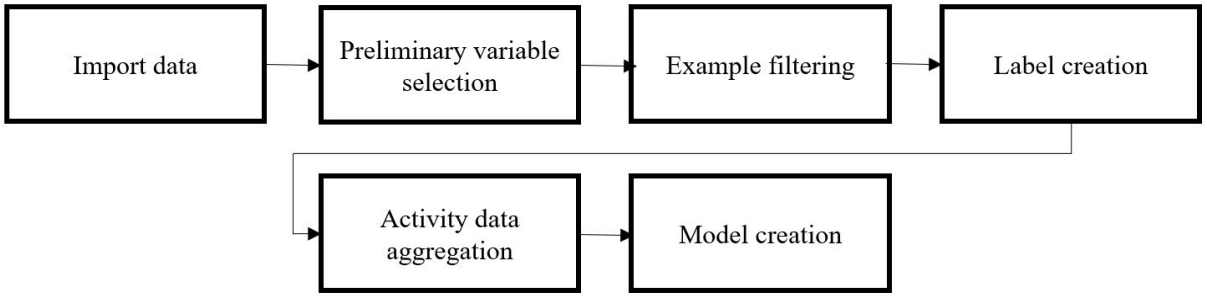


Figure 4.1. Process flowchart

4.3.1 Contact data filtering

Since only contacts from Finland are included in the analysis, all other contacts must be excluded. Using the variable Country, it is possible to exclude all countries that do not

contain the values FI or Finland. However, roughly 40% of examples have missing values in this variable (see Appendix B for RapidMiner process).

A large amount of the examples with missing Country values can be assumed to be companies since they are not missing their industry variable. In addition, an insignificantly small amount of these companies have made a purchase in the past. Most contacts have missing values in the other variables as well except for the email address. As such, contacts with missing country variables that have an email domain ending with .fi, .com and .net will also be kept in the data set (see Appendix B for RapidMiner process).

A choice regarding the inclusion of either B2B or B2C, or both types of customers needs to be made. An abnormally small amount of B2B customers resulting from the previous filtering process have made a purchase, which leads to the reasonable conclusion that the purchase moment variable applies mostly to B2C customers, and that companies may have alternative processes in place when they make a purchase. Due to this, companies are excluded by removing all contacts that have a value in the Company variable except for some standard values that indicate that the contact is a B2C customer (see Appendix B for RapidMiner process).

Data is included in the analysis from the beginning of 18.2.2018 until 16.11.2018. This is to comply allow for a set time frame, which makes it easier to handle activity data as well (see Appendix B for RapidMiner process).

4.3.2 Creating the label

The label is created using a variable that tells us when the project manager is notified of a project starting. This is done after a deal has been made, so it can be used to pinpoint the approximate time of a purchase. A new variable called hasPurchased is created from the variable. The variable takes the value 1 if the contact has made a purchase and the value 0 if the value is missing for the contact. This is a case of informative missingness, where the value is missing if a purchase has yet to be made. One could also argue that the data is

censored, since it is still unknown whether the contact will make a purchase or not (see Appendix C for RapidMiner process).

4.3.3 Activity data filtering

The activity data set has three variables which specify what type of activity was done, who did the activity and when it happened. The date variable is in the wrong format and is fixed before filtering out activities outside of the time frame (see Appendix D for RapidMiner process).

4.3.4 Activity data aggregation

In its current format the activity data is not suitable for supervised learning. To remedy this, the data must be aggregated so that each row contains the data for each contact. Different values are calculated for each activity type, including the amount of activities, the average number of days between the current activity and the last activity and the number of days between the first and last activity. Other values such as the number of activities within the past few days or weeks are also included (see Appendix E for RapidMiner process).

When aggregating the activities, the buyers are assigned an end date corresponding to their purchase moment. Alternatively, the last activity before the purchase moment can be chosen. Activities after each contact's end date are excluded from the aggregation. The reasoning behind this decision is that to correctly teach the machine learning algorithm to detect potential buyers only the activities before the purchase moment can be used. If the activities after the purchase decision were to be included, the model's ability to detect contacts exhibiting potential buyer behavior would diminish since it would be looking for contacts who are already buyers.

For non-buyers, different ways to determine the end date are tested. One way is to set the end date to be the end of the time period, which is 16.11.2018. Another way is to set it to be the non-buyer's last activity. Yet another way is to choose a random end date between the non-buyer's first and last activity. Furthermore, the last activity before the randomly chosen end date between the first and last activity can be chosen as the end date. As with buyers, any activity after the end date is excluded from the aggregation.

One must exercise caution when treating the classes differently. Due to the class imbalance present in the data set, modifying the non-buyers to represent different stages in their customer lifecycle should not cause the model to become overly biased, but it may still infer a slight amount of bias into the model. Another thing to note is that when setting a random end date between the first and last activity, the last activity will always be excluded from the non-buyer's activities unless the non-buyer only has one activity.

4.3.5 Final variable selection

The variables created by the aggregation can be seen in Table 4.3. These variables were calculated for each activity type. The activity types include web browsing sessions, email activities and other related metrics.

Type	Description
Contact	Identifies who did the activity.
daysToEnd.max	Days between the first activity and the end date.
daysToEnd.avg	Average of days between all activities and the end date.
sum	Total amount of activities.
1daySum	Amount of activities within 1 day of the end date.
3daySum	Amount of activities within 3 days of the end date.
1weekSum	Amount of activities within 1 week of the end date.
2weekSum	Amount of activities within 2 weeks of the end date.
4weekSum	Amount of activities within 4 weeks of the end date.
10percentSum	Amount of activities within 10 percent of total time since the end date.
40percentSum	Amount of activities within 40 percent of total time since the end date.
80percentSum	Amount of activities within 80 percent of total time since the end date.

Table 4.3. Activity data, aggregated values

4.4 Model building

All models were created using SMOTE up-sampling except for the neural network model. Cross validation using 10 folds was used for resampling to obtain a fair estimate of the model's performance. No activities before 18.2.2018 or after 16.11.2018 are included. Only B2C contacts from Finland have been included in the results and email domain filtering was applied to the contacts whose nationality was unknown. In the following, different ways to aggregate the data are tested to see which aggregation method produces the smallest amount of bias and the highest performance (see Appendix F for RapidMiner process).

4.4.1 Aggregation 1

These results were obtained when setting the end date for non-buyers to correspond to the end of the chosen time period, which is 16.11.2018. For buyers, the end date was set to be the same as their purchase moment.

As a result, non-buyers are going to have very different aggregated values depending on when they were active. However, buyers will still have quite similar aggregated values since their end date corresponds to their purchase moment. Of course, some non-buyers will have similar aggregated values as the buyers. The model seems to learn this difference and becomes effective at distinguishing non-buyers from buyers based on the variables related to the end date.

	true false	true true	class precision
pred. false	8346	74	99.12%
pred. true	1134	676	37.35%
class recall	88.04%	90.13%	

Figure 4.4.1 Decision tree confusion matrix 1

	true false	true true	class precision
pred. false	8395	64	99.24%
pred. true	1085	686	38.74%
class recall	88.55%	91.47%	

Figure 4.4.2 Random forest confusion matrix 1

	true false	true true	class precision
pred. false	8275	92	98.90%
pred. true	1205	658	35.32%
class recall	87.29%	87.73%	

Figure 4.4.3 Logistic regression confusion matrix 1

	true false	true true	class precision
pred. false	9163	272	97.12%
pred. true	317	478	60.13%
class recall	96.66%	63.73%	

Figure 4.4.4 Neural networks confusion matrix 1

4.4.2 Aggregation 2

In this aggregation method, the end date for non-buyers was set to be the moment of their last activity. For buyers, the purchase moment was used as the end date.

Since the end date for buyers does not coincide with an activity, like it does for non-buyers, the model learns these small differences and becomes very good at predicting buyers. This is a bias since the effectiveness mostly stems from the fact that the aggregations are calculated in slightly different ways for both classes.

	true false	true true	class precision
pred. false	9407	43	99.54%
pred. true	73	707	90.64%
class recall	99.23%	94.27%	

Figure 4.4.5 Decision tree confusion matrix 2

	true false	true true	class precision
pred. false	9468	55	99.42%
pred. true	12	695	98.30%
class recall	99.87%	92.67%	

Figure 4.4.6 Random forest confusion matrix 2

	true false	true true	class precision
pred. false	7602	90	98.83%
pred. true	1878	660	26.00%
class recall	80.19%	88.00%	

Figure 4.4.7 Logistic regression confusion matrix 2

	true false	true true	class precision
pred. false	9468	92	99.04%
pred. true	12	658	98.21%
class recall	99.87%	87.73%	

Figure 4.4.8 Neural networks confusion matrix 2

4.4.3 Aggregation 3

This aggregation method sets the end date for buyers to be the last activity before their purchase decision. For non-buyers, the last activity is set as the end date.

This method fixed the bias that occurred in aggregation methods 1 and 2, but the recall and precision values have dropped. However, this seems to be the fairest, most un-biased method of aggregating the activity data.

	true false	true true	class precision
pred. false	6551	252	96.30%
pred. true	2929	498	14.53%
class recall	69.10%	66.40%	

Figure 4.4.9 Decision tree confusion matrix 3

	true false	true true	class precision
pred. false	6544	230	96.60%
pred. true	2936	520	15.05%
class recall	69.03%	69.33%	

Figure 4.4.10 Random forest confusion matrix 3

	true false	true true	class precision
pred. false	5492	174	96.93%
pred. true	3988	576	12.62%
class recall	57.93%	76.80%	

Figure 4.4.11 Logistic regression confusion matrix 3

	true false	true true	class precision
pred. false	8553	478	94.71%
pred. true	927	272	22.69%
class recall	90.22%	36.27%	

Figure 4.4.12 Neural networks confusion matrix 3

4.4.4 Aggregation 4

These results were obtained by determining a random end date between the first and last activity for non-buyers and setting the end date as the last activity before the randomly selected end date. For buyers, the end date variable is set to be the last activity before the purchase decision.

Setting an end date between the first and last activity will always exclude the last activity for non-buyers if they have more than one activity. This may increase bias towards detecting non-buyers since they will always have fewer activities in this setting. However, selecting a random end date between the first and last activity for non-buyers is meant to

simulate non-buyers in different stages of the customer life cycle, which may altogether be a fairer way to teach the model.

	true false	true true	class precision
pred. false	7496	232	97.00%
pred. true	1984	518	20.70%
class recall	79.07%	69.07%	

Figure 4.4.13 Decision tree confusion matrix 4

	true false	true true	class precision
pred. false	7294	163	97.81%
pred. true	2186	587	21.17%
class recall	76.94%	78.27%	

Figure 4.4.14 Random forest confusion matrix 4

	true false	true true	class precision
pred. false	6066	127	97.95%
pred. true	3414	623	15.43%
class recall	63.99%	83.07%	

Figure 4.4.15 Logistic regression confusion matrix 4

	true false	true true	class precision
pred. false	8936	477	94.93%
pred. true	544	273	33.41%
class recall	94.26%	36.40%	

Figure 4.4.16 Neural networks confusion matrix 4

4.4.5 Aggregation 5

For these results the end date for non-buyers was set to be the actual randomly determined end date between the first and last activity instead of the last activity date. The buyers' end date values were set to be the purchase decision moment.

The results seem biased since buyers are using a predetermined end date corresponding to their purchase moment and the non-buyers are using a randomly generated one inside of their activity timeline. This means that the last of the non-buyers activities will always be left out. In any case, the model notices these subtle differences in the way the end dates are set which causes a rise in the class recall of the positive class at the cost of negative class recall.

	true false	true true	class precision
pred. false	8144	132	98.41%
pred. true	1336	618	31.63%
class recall	85.91%	82.40%	

Figure 4.4.17 Decision tree confusion matrix 5

	true false	true true	class precision
pred. false	8017	76	99.06%
pred. true	1463	674	31.54%
class recall	84.57%	89.87%	

Figure 4.4.18 Random forest confusion matrix 5

	true false	true true	class precision
pred. false	7814	97	98.77%
pred. true	1666	653	28.16%
class recall	82.43%	87.07%	

Figure 4.4.19 Logistic regression confusion matrix 5

	true false	true true	class precision
pred. false	9012	324	96.53%
pred. true	468	426	47.65%
class recall	95.06%	56.80%	

Figure 4.4.20 Neural networks confusion matrix 5

4.5 Model evaluation

In this chapter one of the aggregation methods shown in chapter 4.4 is selected. In addition, the models that were created using that method are compared against each other using different types of performance metrics. Through this process, the best model is identified.

4.5.1 Choosing an aggregation method

When looking at the different types of aggregation methods discussed in chapter 4.4, some of them stand out as exhibiting some sort of bias or data leak due to the classes being treated differently. Data leaks occur when there are attributes present in the data that directly or indirectly tell the model which class any given example belongs to. This prevents the model from learning to correctly predict the leads based on their behavior.

Table 4.5.1 describes how the aggregation methods differ from each other as well as their bias degree and the best model's AUC. The bias degree column values are based on the descriptions of the different aggregation types in chapter 4.4 and represents my subjective understanding of the models.

Aggregation type	Buyer end date	Non-buyer end date	Bias degree	Best model AUC
1	Purchase moment	Last date in data set (16.11.2018)	High	Random forest: 0.955
2	Purchase moment	Last activity	High	Random forest: 0.991
3	Last activity before purchase moment	Last activity	None	Random forest: 0.761
4	Last activity before purchase moment	Last activity before randomly selected end date (between first and last activity)	Low	Random forest: 0.843
5	Purchase moment	Randomly selected (between first and last activity)	Medium	Random forest: 0.935

Table 4.5.1 Aggregation type comparison

Aggregation methods 1 and 2 show a high degree of bias, method 5 shows more bias than method 4 and method 3 does not seem to show any type of bias at all. Aggregation method 3 is chosen as the most effective aggregation method due to its lack of bias and relatively high AUC compared to random selection, where the AUC would be 0.5.

4.5.2 Model comparison

Next, the models created by the chosen aggregation method are evaluated and compared. Table 4.5.2 shows performance metrics for each model. Accuracy is not a reliable metric due to the high class imbalance in the data. Instead, AUC and Youden should be used to obtain a general estimate for the model's performance. Sensitivity and specificity should be considered to understand how the models have correctly categorized the two classes.

Model	Accuracy	AUC	Sensitivity	Specificity	Youden
Decision Tree	68.91%	0.723	66.40%	69.10%	0.355
Random Forest	69.05%	0.761	69.33%	69.03%	0.384
Logistic Regression	59.32%	0.698	76.80%	57.93%	0.347
Neural Network	86.27%	0.749	36.27%	90.22%	0.265

Table 4.5.2 Model performance metrics using aggregation method 3

As expected, the decision tree model is not as effective as the random forest model. It scored lower than the random forest model on all performance metrics except specificity and the resulting model was not easy to interpret. The created decision tree has a maximum depth of 10, meaning that the maximum number of decision points that a branch could have is 10. Despite decision trees being known for their high interpretability, a depth of 10 means that there could be up to 2^10 possible final decision points which is a large amount of information to interpret. Even with pruning the resulting decision tree proved challenging to interpret and restricting the tree to smaller maximum depths could have further decreased the different performance metrics. If one were to create a tree without pruning, it would be more prone to overfitting despite resampling procedures, and the gained interpretability may not be worth the decrease in model effectiveness.

The random forest model was created using 100 decision trees and has the best overall score. It has very similar specificity, sensitivity and accuracy scores as the decision tree model but higher Youden and AUC scores. It is not possible to interpret the model in a way that is like the decision tree model, but it does have higher average performance as expected. It is also possible to produce the attribute importances of the random forest model, which can be examined to gain an insight into which predictors are more important than others.

Logistic regression was mainly included in the procedure to obtain a benchmark for what a linear classification algorithm could achieve compared to the more complex, non-linear machine learning algorithms. However, despite having lower overall scores, the linear model still succeeded in obtaining a higher Youden value than the neural network

model. The model also achieved the highest sensitivity, albeit the lowest specificity, which means it is better at identifying the more important class than other models, but at the cost of being worse at identifying the less important class. Logistic regression has high interpretability due to the attribute coefficients, however the low overall performance makes it less useful than the other models.

Despite having the highest accuracy and only slightly lower AUC, the neural network model has treated the classes very differently. This can be observed by looking at the sensitivity and specificity values of the model. It seems to have correctly guessed 90.22 % of the negative class, but only 36.27 % of the positive class. In lead scoring or marketing in general, one could argue that it is more important to be able to detect the positive than the negative class. The model has next to no interpretability since the type of neural type that was used does not have a built-in function to calculate attribute importances.

In conclusion, the random forest model is selected as the best performing model and will be analyzed in chapter 5. This decision is based on the model having the highest overall performance score and the possibility to interpret the model through attribute importances. However, if one were to assign financial values such as the cost of losing a potential lead versus the cost of contacting a lead, the value of each model could change. For example, the logistic regression model may be better than the other models if sensitivity were to have a higher value than specificity.

5. Discussion

The focus of this chapter is to analyze the best performing model and discuss the empirical study as a whole. Visual analytics is used to gain an insight into what types of activities leads usually have depending on their assigned purchase probability. The purchase probability is the same as the confidence value that the random forest model has assigned to all leads. It is a value representing the probability that the lead is exhibiting behavior like that of a buyer, moments before a purchase is made.

5.1 Visual analysis

Visual analysis is a tool used to gain insights into how the data looks like. Understanding how the model has distributed the confidence values across the population will allow us to gain insights into how leads that have a high, low and medium purchase probability behave prior to purchasing the product. It is important to understand how data is collected and modified; otherwise faulty conclusions may be drawn from the visualizations. For example, an email may be sent automatically once a form is submitted, and an email may be opened automatically by some email client's spam detection or anti-virus functions. Thus, understanding the behavior of the data is important.

Before plotting the data, it is good to look at the variable importances to obtain an understanding of which attributes to pay attention to. The heatmap shown in Figure 5.1 allows us to compare the importances of different activities and aggregated metrics. The bar at the bottom of the figure depicts the importance of the variable by color. A variable depicted with a red color has an importance of 2 and a variable depicted with a blue color has an importance of 0. If a variable is depicted with a yellow color it has an importance of 1. Colors between blue and yellow indicate that the variable has an importance of something between 0 and 1, while the colors between yellow and red indicate that the variable has an

importance between 1 and 2. Unsubscribes and bouncebacks seem to be completely irrelevant to the model, meaning removing them from future models could be a good idea. One would have assumed that unsubscribes would be used to give a lead a lower lead score, but that does not seem to be the case. Form submission related attributes seem to be the most important. This may be due to form submissions being the primary way in which a lead can signal that they are interested in purchasing the product. The importance of website browsing sessions and page views could also be, in part, attributed to the fact that form submissions are so important. This is because to submit a form, the lead must first navigate through the website. Out of all email variables, email opens seem to be the most important one. The time variables seem to both be very important predictors, which underlines the importance of using an unbiased aggregation method.

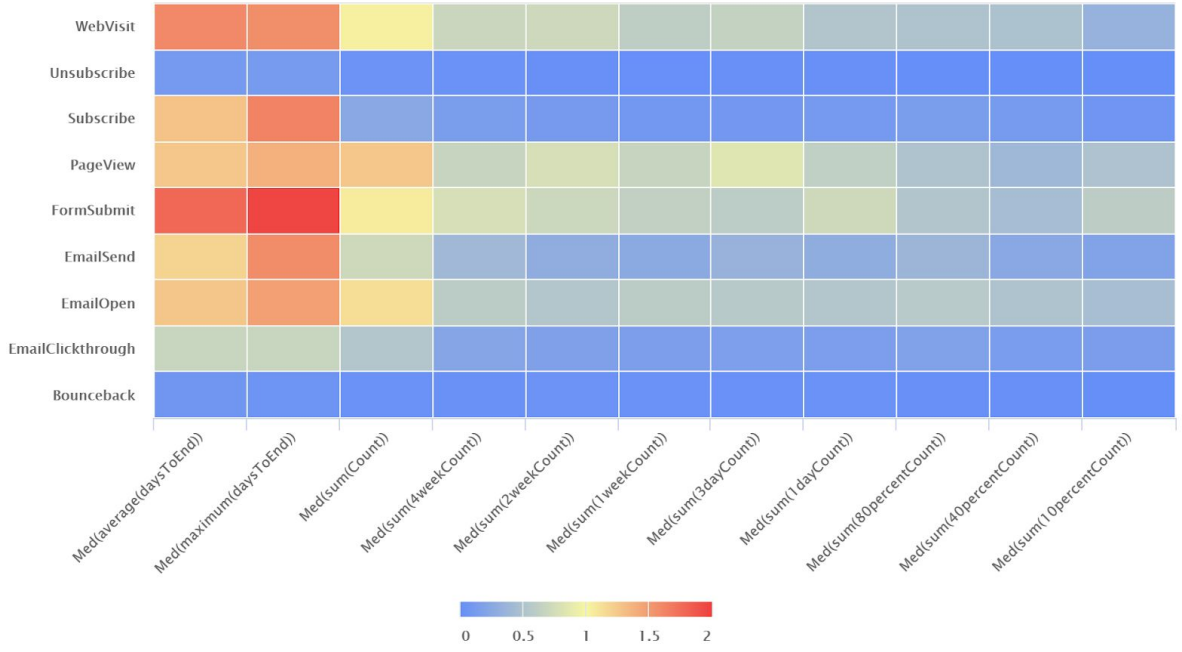


Figure 5.1 Random forest attribute importance heatmap

Figure 5.2 depicts the average and median amount of activities across five confidence value groups. The blue bars are the average activities and the green line is the median amount of activities. Each column contains an equal sum of leads, allowing us to visualize the behavior of leads in each group. There also seems to exist quite a few outliers in all columns

except the lowest. This can be observed by looking at the difference between the median and average of each column.

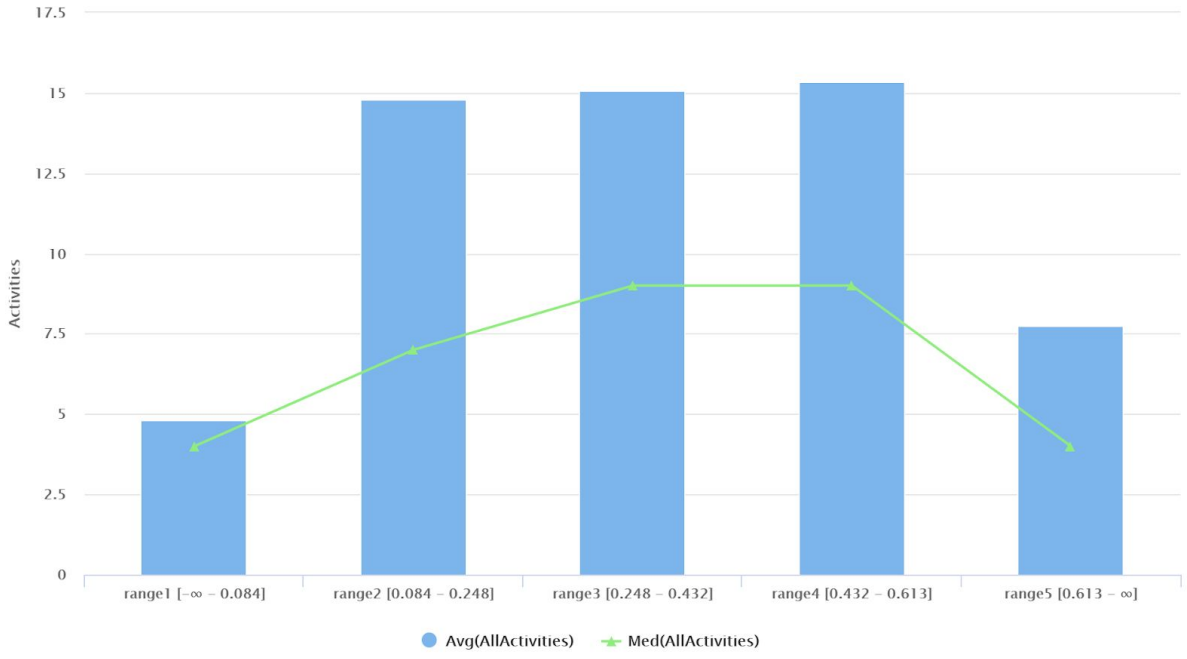


Figure 5.2 Average and median activity amount per purchase probability group

In Figure 5.3, the median amount of different types of activities are shown. The different types of activities are stacked on top of each other and distinguishable through their colors. For example, the highest confidence rating group does not have a median of three browsing sessions. That group has one browsing session, one form submission and one subscription, which is a median total of three activities. Note that the slight difference from Figure 5.2's median line (green) is due to the way that the median value is counted separately for each activity in Figure 5.3.

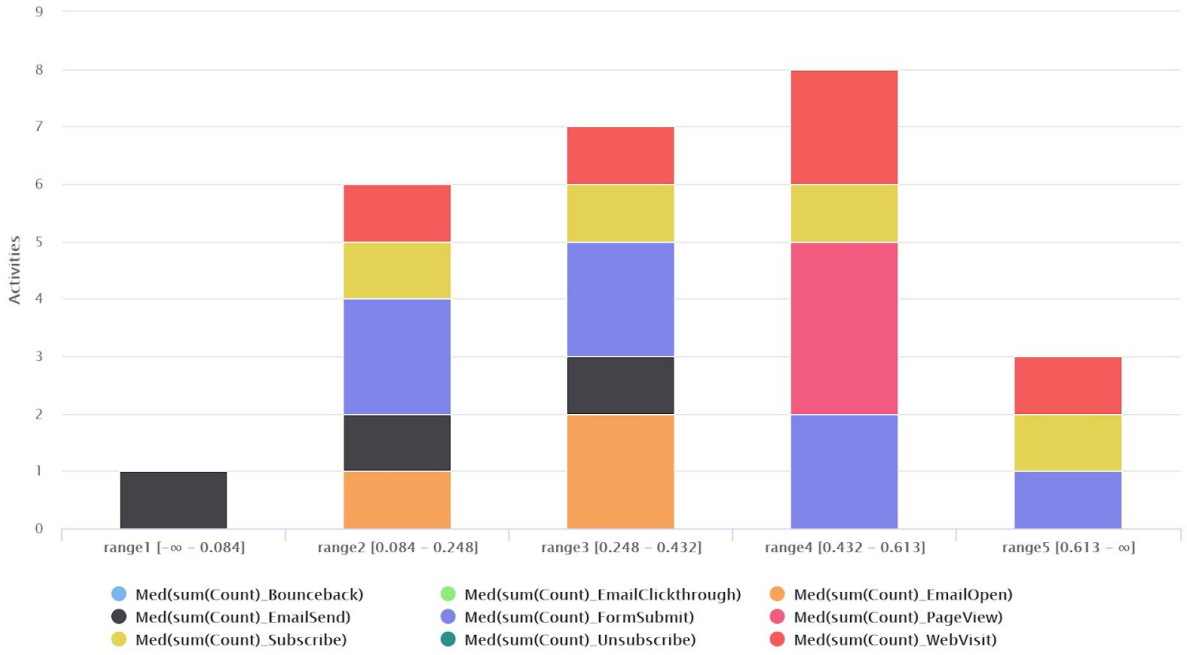


Figure 5.3 Median activity amount per purchase probability group

In Figure 5.4, the length of the customer's activity span, or the amount of days between the first activity and the end date, is visualized in a similar way as in Figure 5.2. The trend seems to be that more time between the first and last activity indicates a lower purchase probability. The median value is close to 0 for several of the columns, suggesting that there are outliers in this chart as well. If the activity time span is between 0 and 1, it means that the lead only had activities on one day, suggesting that they may have simply visited the website, sent a form and subscribed simultaneously.

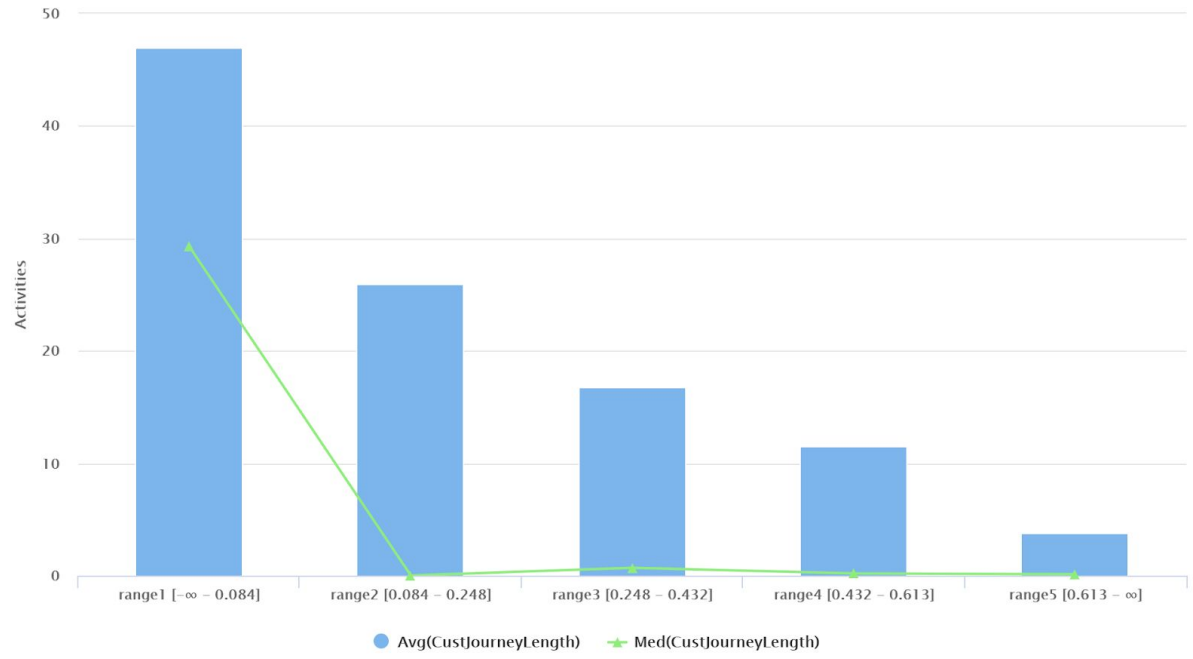


Figure 5.4 Average and median customer activity time span per purchase probability group

By looking at the data as a whole, it seems the leads with the lowest lead scores have a very high amount of time between the first and last activity, suggesting that a wider customer activity time span has a decremental effect on the purchase probability. In addition, these leads have very few activities, the most common one being receiving emails.

In contrast to the behavior exhibited by leads with the lowest purchase probability, the leads that have an average to low purchase probability seem to have a shorter average activity time span. These leads usually have at least some browsing sessions, a subscription, form submission and have opened some emails.

The leads in with a high lead score differ from the other ones because they seem to show their interest by browsing the webpages. These leads do not necessarily have any email activities, but they have a large amount of website browsing sessions and form submissions along with a shorter activity time span.

Leads with the highest lead score seem to have only visited the webpage, submitted a form and subscribed. These leads also have a very short average and median activity time span.

The median activity time span is much lower for all groups except for leads with the lowest lead score, suggesting that there is a type of lead that only visits the webpage once, submits a form and does nothing else. The fact that these types of leads are also present in higher purchase probability value groups suggests that they have made purchases, meaning that they could be leads that were contacted outside of the data and made a purchase decision prior to submitting the form. On the other hand, these leads could simply be people that made a form submission, received an offer they were happy with and made a purchase without further activities.

5.2 Empirical study discussion

Some aspects of the structure could be improved upon. For example, the first and second step should be more of an iterative process instead of strictly moving on to the next step after finishing the first. In addition, understanding the business objective and the data would be easier if there was a more active communication channel between the client and the data scientist. Data preparation and model building were also closely linked together due to the need to test different aggregation methods and how they affect the resulting model.

The program used for this empirical study, RapidMiner, proved to be quite useful for performing quick visualization, data transformations and machine learning models. It can be highly recommended to use RapidMiner for data science projects, even though the more advanced data transformations can be quite difficult to implement without prior experience.

When optimizing the models, AUC was used as the primary performance criterion. AUC is not sensitive to class imbalance, so it made sense to emphasize that over other metrics. Sensitivity and specificity are convenient metrics that describe how the model has treated the different classes, but they should not be used by themselves as the primary performance criterion in lead scoring. Due to the high class imbalance, using accuracy as the primary performance metric was out of the question.

Out of all models, random forest was the one with the highest overall score. This was not unexpected, as it seems to be the general opinion of many machine learning communities that random forest and neural networks are the best-performing algorithms for most machine learning problems. However, the low performance of the neural network model might be due to limited model parameter adjustments. Since the model takes a long time to train, adjusting the parameters for a perfect fit would have been an extremely lengthy endeavor. Logistic regression was useful in creating quick models that also made the attribute coefficients available.

The machine learning model that was created using the random forest algorithm does not respond to real time. This means that the scores do not change as time goes on, only when the lead performs another activity. An obvious drawback of this is that the model could include very old leads with very high lead scores, but a CRM system using this should still have a way to or mark the leads that have already been contacted.

6. Conclusion

The objective of this thesis has been to understand how machine learning can be used to perform lead scoring. The thesis has, to some extent, been able to provide some answers to the following research questions:

- *Which machine learning algorithm gives the best performance in lead scoring?*
- *What business insights are to be derived from the lead scoring results?*

6.1 Research questions

The answer to the first research question is the random forest algorithm. It had the best overall performance out of all the different models and the algorithm did not take nearly as long to build the model as, for example, the neural network did. However, there is still room for improving the models through extensive parameter optimization. Therefore, it is impossible to say whether, for example, the neural network model would have performed better than the random forest model had its parameters been extensively optimized. Since there are countless algorithms and other data manipulation procedures that are not included in this thesis, it is impossible to say that the random forest algorithm is for certain the best among them.

There were no comparisons with lead scoring using machine learning and manual lead scoring, so it is not possible to directly say which one is better. However, it should be noted that the study by Aberdeen (Michiels, 2008) notes that the inclusion of implicit attributes often serves to increase the performance of lead scoring models. The study also notes that a more complex model is often better than a simple one. Both claims indicate that supervised machine learning would be suited for lead scoring (Michiels, 2008).

The second research question was primarily answered in chapter 5.1, during the data visualization. The primary insight from that chapter is that it is of great importance to understand how the business processes are reflected in the data. For example, at first glance it seems as if the first few moments that a lead has visited the site and submitted a form are the most crucial in securing a purchase. However, it is entirely possible that the lead has obtained information elsewhere and made his or her purchase decision before even visiting the site the first time. Additional activity data is required to verify this; however, I believe that the model has treated the time aspect in the correct way, assigning the leads with a longer activity time span a lower lead score.

6.2 Future research

Some areas of possible future research have been brought to light during the progress of the thesis. Companies could possibly be interested in adding customer lifetime value to lead scoring, resulting in a monetary value which may seem more tangible than a simple purchase probability. For example, one could just multiply customer lifetime value with the purchase probability. Another example would be to use regression instead of classification to estimate the customer lifetime value of leads.

In addition, identifying different lead types would be beneficial for companies. That way, they could treat the different types of leads with different types of marketing material, for example through nurturing campaigns. This could probably be done using unsupervised learning, since it is unknown how many different types of leads there are.

Furthermore, different ways to aggregate or handle time-series data should be researched. There should be a way to aggregate time-series data with minimal information loss, however I did not find any such literature. However, if one were to use a machine learning algorithm that is compatible with time-series data, the problem of aggregating the data with minimal information loss does not exist.

7. Svensk sammanfattning

Under senaste åren har alltmer data samlats in av företag som försöker använda dem för att lösa affärsproblem. Ett av dessa problem är att bestämma vilka kunder försäljningsteamet bör prioritera. Företag vill också veta varför kunder engagerar dem så att de kan fatta effektivare marknadsföringsbeslut. Lead scoring är ett sätt att prioritera kunder enligt sannolikheten att de köper något. Ett manuellt lead scoring-system kunde till exempel ge en kontakt 5 poäng om kontakten besökte företagets webbsida och kanske 25 poäng om kontakten svarat på ett e-mail. Idén är att kontakterna med högre lead score har en större sannolikhet att köpa en produkt, och därmed borde försäljningsteamet ägna dem mera tid. Problemet med manuell lead scoring är att poängen som delas ut för diverse aktiviteter inte är grundade på statistik. Ett annat problem är att den utdelade mängden av poäng inte beaktar kontaktens andra egenskaper, till exempel kön eller tidigare aktiviteter. Manuell lead scoring klarar inte av att behandla alla data som krävs för att skapa en fungerande och opartisk lead scoring-modell.

Däremot kunde man använda prediktiv analys för att skapa en modell som har kapacitet att bearbeta alla data som behövs. Prediktiv analys består av en samling statistiska och matematiska metoder som kan användas för att förutspå framtiden. Maskininlärning är en del av prediktiv analys. Övervakade inlärningsalgoritmer kan användas för att skapa modeller som förutspår vilka kontakter som kommer att köpa en produkt i framtiden. Detta görs genom att mata in historiska data i inlärningsalgoritmen som sedan lär sig urskilja beteendet av kontakter som tidigare köpte och kontakter som inte köpte en produkt. När modellen är skapad kan man undersöka hur viktiga vissa aktiviteter eller händelser är för att kunden ska köpa produkten.

Syftet med pro gradu-avhandlingen är att utforska hur maskininlärning kan användas för lead scoring. Avhandlingen innehåller en empirisk studie som stöds av en litteraturöversikt som behandlar maskininlärning och lead scoring. Den empiriska studien är ett lead scoring-projekt för en av ID BBN:s klienter. Forskningsfrågorna är följande:

- *Vilken maskininlärningsalgoritm presterar bäst i lead scoring?*

- *Hurdana affärsinsikter kan man hitta i lead scoring-resultaten?*

7.1 Metod och empirisk studie

Syftet med studien är att slutföra ett lead scoring-projekt för en av ID BBN:s klienter och tillika få svar på forskningsfrågorna. I den empiriska studien används en modifierad version av Schmueli och Koppius (2011) kvantitativa metod för att utföra forskning som använder sig av prediktiv analys. Den modifierade versionen av metoden förenklar den ursprungliga för att göra studien mer begriplig. Metoden består av sex steg, varav de fem första stegen är placerade under kapitlet om den empiriska studien och det sjätte under diskussionskapitlet.

Studiens första steg är att förstå klientens affärsproblem. Klienten vill optimera sitt försäljningsteams tidsanvändning genom att fokusera på de kontakter som har den största köpsannolikheten. Produkten är ett stort B2C-köp och endast kontakter från Finland inkluderas i studien.

Det andra steget är att förstå klientens data och välja de variabler som ska tas med i resten av studien. Data hanteras i RapidMiner och laddas ner från klientens Eloqua-server. Två olika typer av data används i studien. Kontaktdata innehåller variabler som kontakternas stad, e-post-domän och om kontakten gjort ett köp. Eftersom kontaktdata är fyllt med väldigt många olika variabler, väljs endast de väsentliga och resten exkluderas. Aktivitetsdata innehåller alla kontakters aktiviteter, som till exempel besök till företagets webbsida och om kontakten öppnat ett e-post. Aktivitetsdata innehåller variabler som beskriver en tidspunkt, en aktivitetstyp och vilken kontakt som gjorde aktiviteten. Problemet med aktivitetsdata är att de är formaterade på ett sätt som inte är kompatibelt med de flesta övervakade inlärningsalgoritmer. En kontakt kan ha många aktiviteter, vilket innebär flera än en rad aktivitetsdata. Därför måste aktiviteterna aggregeras till en rad per kontakt.

I det tredje steget filtreras kontakterna så att enbart B2C-kunder från Finland är inkluderade i data. Sedan förbereds aktivitetsdata så att de är kompatibla med algoritmerna. Aktiviteter som skedde efter att en kontakt hade köpt en produkt tas inte med, eftersom

algoritmen ska lära sig vilka som är köpare innan köpet sker. Utmaningen i att aggregera aktivitetsdata är att det måste göras så att en minimal mängd information går till spillo. Därför skapas variabler som till exempel antal dagar mellan första och sista aktiviteten samt mängden aktiviteter per aktivitetstyp. Utöver variablerna måste olika sätt att aggregera data testas. Eftersom köparnas aktiviteter begränsas enligt deras köpdatum så måste icke-köparnas aktiviteter också begränsas på något sätt. Annars skulle algoritmen genast förstå att de som har korta tidsramar är köpare medan de som har långa tidsramar är icke-köpare.

I det fjärde steget skapas modellerna med fyra olika maskininlärningsalgoritmer: decision tree, random forest, logistisk regression och neurala nätverk. Fem olika sätt att aggregera aktivitetsdata testas också, vilket innebär att 20 olika modeller skapas.

Det femte steget handlar om att evaluera de olika sätten att aggregera data, jämföra modellerna och välja den bästa. Det bästa aggregeringssättet var det som hade den sämsta prestandan men som inte hade någon bias. De andra aggregeringssätten hade troligtvis hög bias eftersom köpare och icke-köpare behandlats på olika sätt. Modellen med den högsta prestandan var random forest.

I det sjätte steget undersöks modellen med hjälp av visuell analys och förbereds sedan inför integration med klientens affärsprocess. Eftersom avhandlingen inte behandlar integrationen flyttas den visuella analysen till diskussionsdelen av avhandlingen.

7.2 Diskussion och slutsats

Visuell analytik används för att undersöka på vilken basis modellen har poängsatt kontakterna. En värmekarta visar vilka aktiviteter och aggregerade variabler som var viktigast för random forest-algoritmen. Formulärinsändningar var de viktigaste aktiviteterna följt av webbsidebesök och öppning av e-post. Antalet dagar sedan formulärinsändningen var den viktigaste variabeln, och avslutandet av prenumerationer samt blockering av e-post var de onödigaste i avhandlingen.

Modellen undersöks vidare genom att skapa histogram av köpsannolikheten och variablerna. De som har den lägsta köpsannolikheten har en väldigt lång tid mellan första och sista aktiviteten och få aktiviteter. Kontakterna med mellanstor köpsannolikhet har lite kortare tid mellan första och sista aktiviteten och fler aktiviteter. De har ofta några e-post-aktiviteter samt en formulärinsändning, ett webbsidebesök och prenumenerar på klientens e-post. De med den högsta köpsannolikheten har ofta väldigt få dagar mellan första och sista aktiviteten samt inga e-post-aktiviteter. Däremot har de ett webbsidebesök, en formulärinsändning och de prenumenerar på klientens e-post. Längre tid mellan första och sista aktiviteten sänker köpsannolikheten och fler aktiviteter höjer den, förutom för de med den högsta köpsannolikheten som verkar ha besökt företagets webbsida en gång och skickat ett formulär.

Det är fullständigt möjligt att orsaken till att algoritmen lär sig dessa beteendemönster är att kontakten bestämt sig att köpa produkten innan kontakten skickat in formuläret. Det är också möjligt att modellen lärt sig att de som skickar in formuläret under sitt första besök på företagets webbsida har den största köpsannolikheten och att de som dröjer och undersöker webbsidan ofta väljer att inte köpa. Mera data behövs för att undersöka detta.

Svaret på den första forskningsfrågan är random forest. Även om modellen hade den bästa prestandan är det inte säkert att den är den absolut bästa modellen för lead scoring. På grund av tidsbegränsningar var det inte möjligt att testa vilka inställningar som skapade den bästa modellen. Detta innebär att en annan modell kunde ha presterat bättre med andra inställningar. Dessutom finns det ett enormt antal algoritmer som inte testades i studien. Den andra forskningsfrågan besvarades med hjälp av visuell analys i diskussionskapitlet. Den huvudsakliga insikten är att det är viktigt att förstå hur affärsprocesserna reflekteras i data. En kontakt kan ha fattat köpbeslutet innan kontakten syns i data, och detta kan påverka maskininlärningsmodellens funktionalitet.

Några framtida forskningsämnen upptäcktes under avhandlingen. Koppling av customer lifetime value med lead score, eller köpsannolikhet, kunde vara ett lukrativt område eftersom företagen då kunde koppla ett estimerat monetärt värde till varje kontakt. Ett annat möjligt framtida forskningsämne kunde vara identifieringen av olika sorters kontaktgrupper baserad på deras köpsannolikhet. Dessa grupper kunde utsättas för olika sorters

marknadsföringsmaterial för att förbättra köpsannolikheten. Olika sätt att aggregera aktivitetsdata eller tidsseriedata för att minimera förlusten av information skulle också vara ett potentiellt framtida forskningsområde. Alternativt kunde man använda sig av en algoritm som inte kräver att tidsseriedata aggregeras före användning.

References

Batista, G. E., & Monard, M. C. (2002). *A Study of K-Nearest Neighbour as an Imputation Method*. HIS, 87(251-260), 48.

Baker, A. M., Hsu, F. C., & Gayzik, F. S. (2018). *A method to measure predictive ability of an injury risk curve using an observation-adjusted area under the receiver operating characteristic curve*. Journal of biomechanics, 72, 23-28.

Benhaddou, Y., & Leray, P. (2017, October). *Customer Relationship Management and Small Data—Application of Bayesian Network Elicitation Techniques for Building a Lead Scoring Model*. In Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on (pp. 251-255). IEEE.

Bohlin, E. 2017. *Sorting Through the Scoring Mess.* URL: https://www.siriusdecisions.com/blog/sorting-through-the-scoring-mess (Retrieved 24.09.2018)

Buckinx, W., & Poel, D. V. D. (2005). *Customer base analysis: Partial defection of behaviorally-loyal clients in a non-contractual FMCG retail setting.* European Journal of Operational Research, 164, 252–268.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). *A systematic study of the class imbalance problem in convolutional neural networks*. Neural Networks, 106, 249-259.

Carter, J. V., Pan, J., Rai, S. N., & Galandiuk, S. (2016). *ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves*. Surgery, 159(6), 1638-1645.

Chen, Y. L., Hsu, C. L., & Chou, S. C. (2003). *Constructing a multi-valued and multilabeled decision tree.* Expert Systems with Applications, 25, 199–209.

Chen, W., Xie, X., Wang, J., Pradhan, B., Hong, H., Bui, D. T., ... & Ma, J. (2017). *A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility*. Catena, 151, 147-160.

Dreiseitl, S., & Ohno-Machado, L. (2002). *Logistic regression and artificial neural network classification models: a methodology review*. Journal of biomedical informatics, 35 (5-6), 352-359.

Gokgoz, E., & Subasi, A. (2015). *Comparison of decision tree algorithms for EMG signal classification using DWT*. Biomedical Signal Processing and Control, 18, 138-144.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning (Vol. 2, No. 12*). New York, NY, USA:: Springer series in statistics.

Järvinen, P. (2004). *On research methods.* Opinpajan kirja.

Kim, Y., & Street, W. N. (2004). *An intelligent system for customer targeting: a data mining approach*. Decision Support Systems, 37(2), 215-228.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling (Vol. 26)*. New York: Springer.

Liu, Z. G., Pan, Q., Dezert, J., & Martin, A. (2016). *Adaptive imputation of missing values for incomplete pattern classification. Pattern Recognition*, 52, 85-95.

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). *The impact of class imbalance in classification performance metrics based on the binary confusion matrix.* Pattern Recognition, 91, 216-231.

Marion, G. 2016. *Lead Scoring is Broken. Here's What to Do Instead.* URL: https://medium.com/marketing-on-autopilot/lead-scoring-is-broken-here-s-what-to-do-instead-194a0696b8a3 (Retrieved 24.09.2018)

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). *Big data: the management revolution*. Harvard business review, 90(10), 60-68.

Michiels, I. (2008). *Lead Prioritization and Scoring: The Path to Higher Conversion.* Aberdeen Group.

Ngai, E. W., Xiu, L., & Chau, D. C. (2009). *Application of data mining techniques in customer relationship management: A literature review and classification.* Expert systems with applications, 36(2), 2592-2602.

Prati, C., Batista, G. E., & Monard, M. C. (2009) *Data mining with imbalanced class distributions: concepts and methods*. In Indian International Conference Artificial Intelligence, pages 359–376.

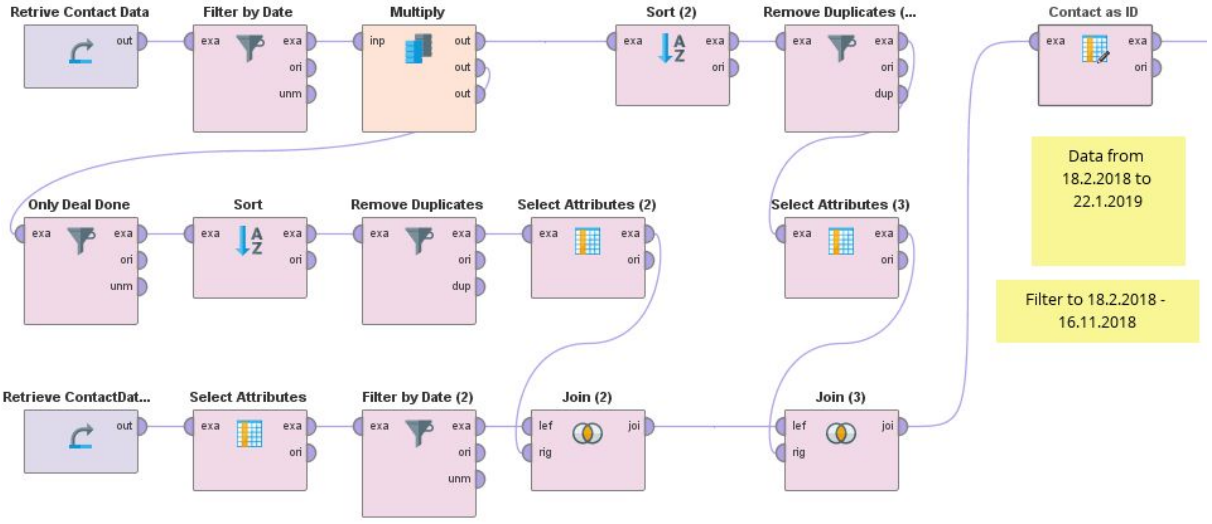
Rosenbröijer, C. J. (2014). *Customer Relationship Management and business analytics: a lead nurturing approach*. Proceedings of DYNAA, 5(1).

Shmueli, G., & Koppius, O. R. (2011). *Predictive analytics in information systems research.* Mis Quarterly, 553-572.

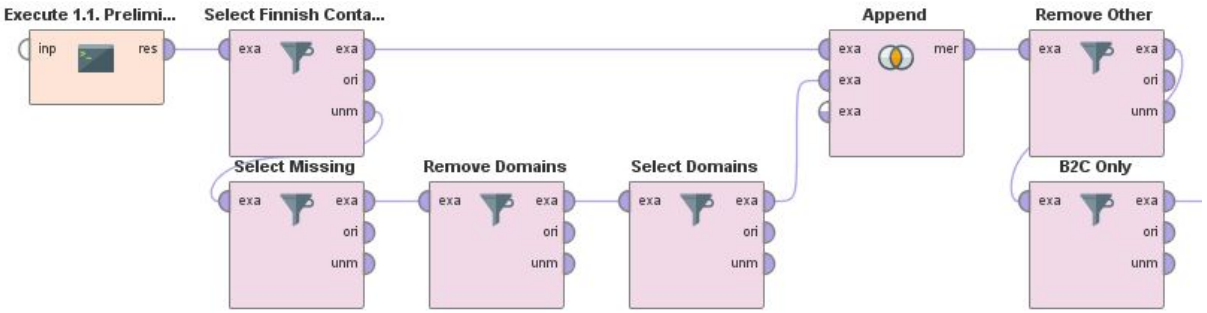
Tamaddoni, A., Stakhovych, S., & Ewing, M. (2016). *Comparing churn prediction techniques and assessing their performance: a contingent perspective.* Journal of service research, 19(2), 123-141.

Wang, L., Zeng, Y., & Chen, T. (2015). *Back propagation neural network with adaptive differential evolution algorithm for time series forecasting*. Expert Systems with Applications, 42(2), 855-863.

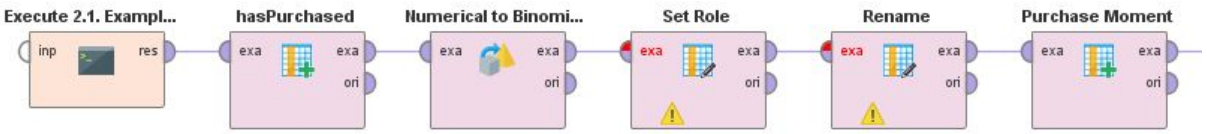
Appendix



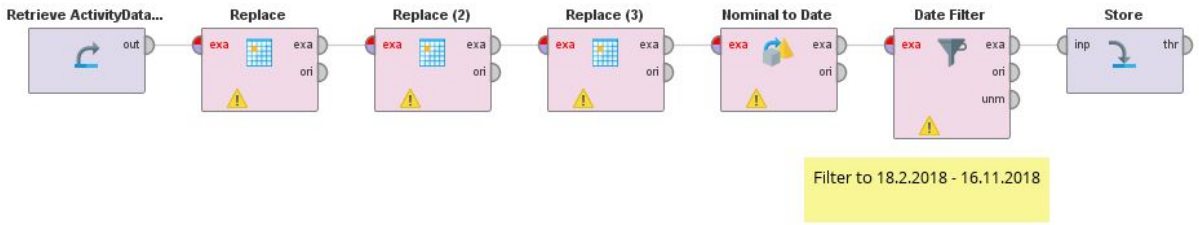
Appendix A. Preliminary variable selection & date filter



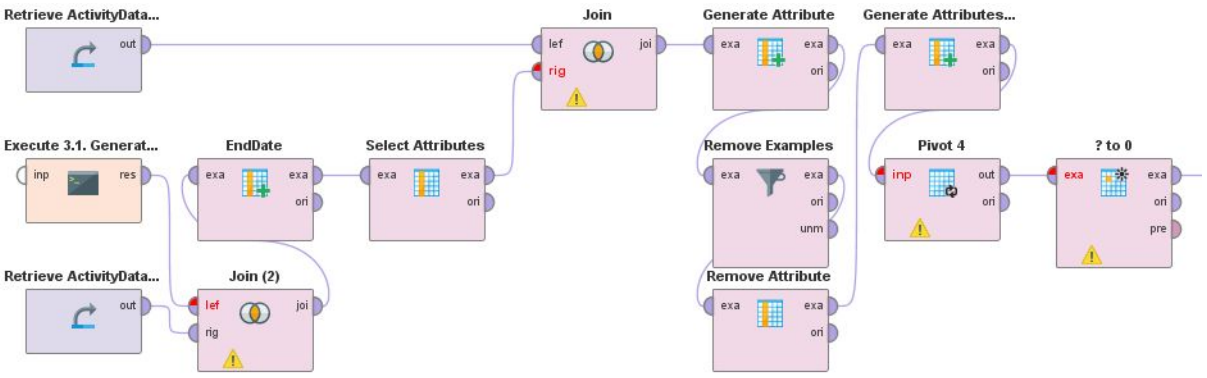
Appendix B. Data filtering



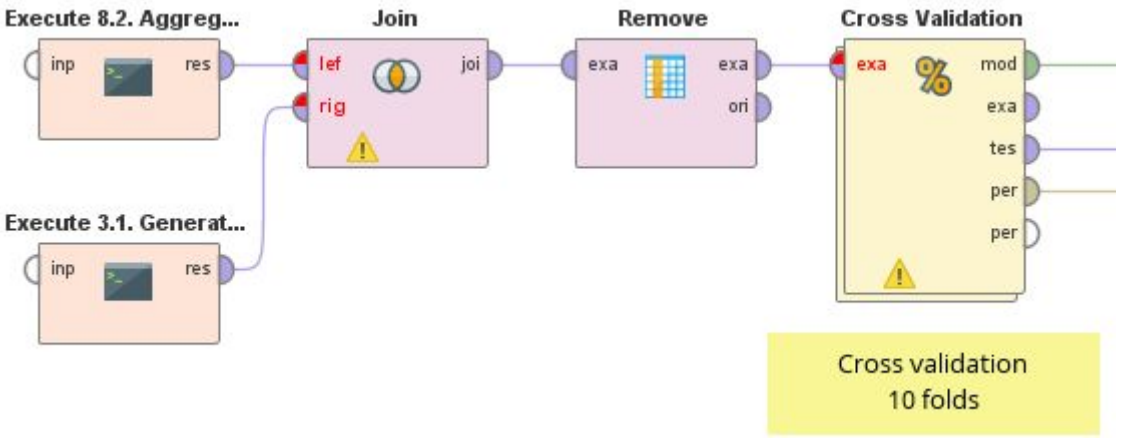
Appendix C. Purchase moment & label creation



Appendix D. Activity data transformation



Appendix E. Activity data transformation



Appendix F. Model creation example