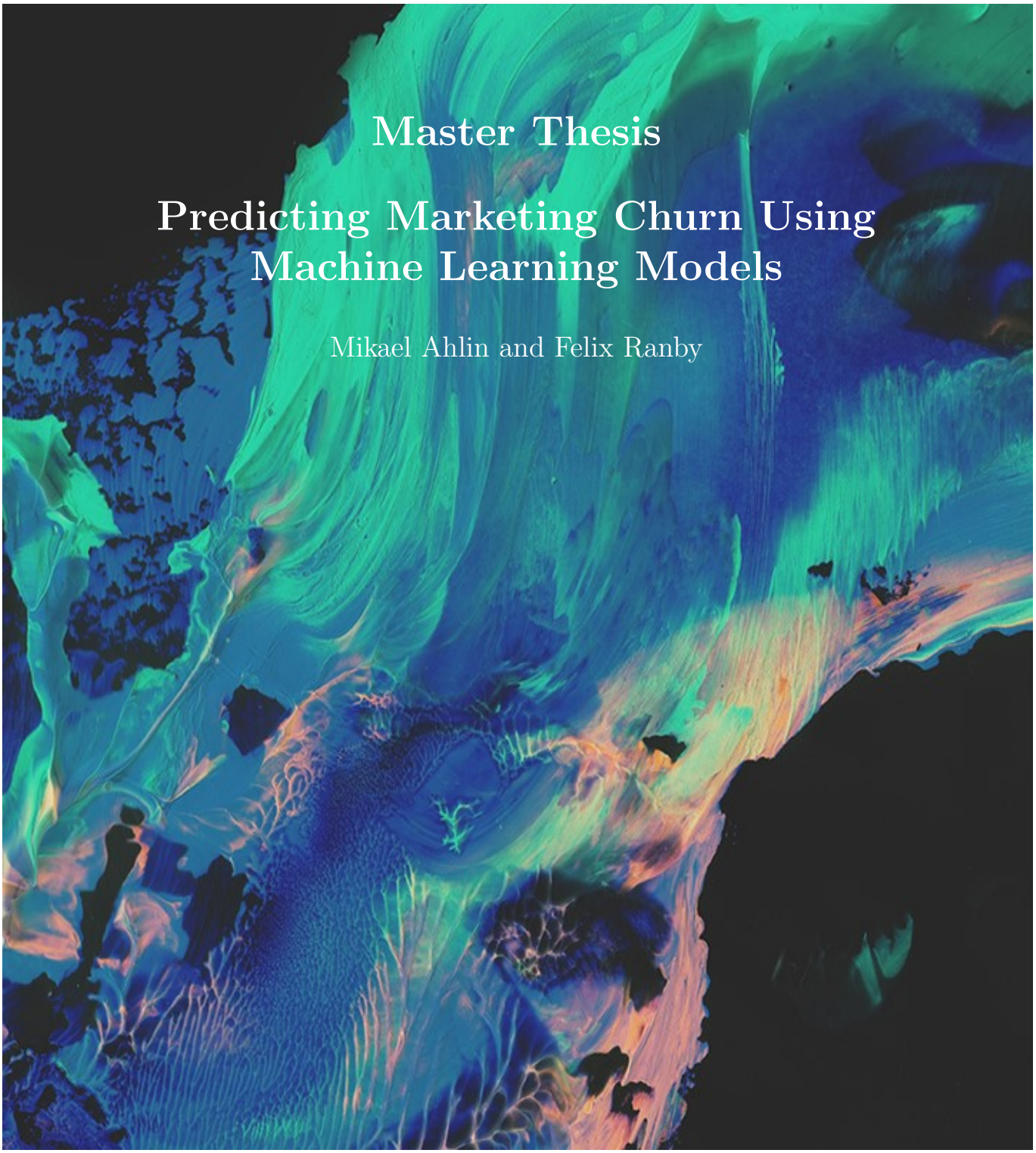

UMEÅ UNIVERSITET

Master Thesis

Predicting Marketing Churn Using Machine Learning Models

Mikael Ahlin and Felix Ranby

Master Thesis, 30 credits
Department of Mathematics and Mathematical Statistics
Spring Term 2019

PREDICTING MARKETING CHURN USING
MACHINE LEARNING MODELS
Submitted in fulfilment of the requirements for the degree Master of Science in Industrial Engineering and Management
Department of Mathematics and Mathematical Statistics
Umeå University
SE - 907 87 Umeå, Sweden

Supervisors:
Peter Anton, Umeå University
David Norell, Telia Company

Examiner:
Natalya Pya, Umeå University

Abstract

For any organisation that engages in marketing actions there is a need to understand how people react to communication messages that are sent. Since the introduction of General Data Protection Regulation, the requirements for personal data usage have increased and people are able to effect the way their personal information is used by companies. For instance people have the possibility to unsubscribe from communication that is sent, this is called Opt-Out and can be viewed as churning from communication channels. When a customer Opt-Out the organisation loses the opportunity to send personalised marketing to that individual which in turn result in lost revenue.

The aim with this thesis is to investigate the Opt-Out phenomena and build a model that is able to predict the risk of losing a customer from the communication channels. The risk of losing a customer is measured as the estimated probability that a specific individual will Opt-Out in the near future. To predict future Opt-Outs the project uses machine learning algorithms on aggregated communication and customer data. Of the algorithms that were tested the best and most stable performance was achieved by an Extreme Gradient Boosting algorithm that used simulated variables. The performance of the model is best described by an AUC score of 0.71 and a lift score of 2.21, with an adjusted threshold on data two months into the future from when the model was trained. With a model that uses simulated variables the computational cost goes up. However, the increase in performance is significant and it can be concluded that the choice to include information about specific communications is considered relevant for the outcome of the predictions. A boosted method such as the Extreme Gradient Boosting algorithm generates stable results which lead to a longer time between model retraining sessions.

Sammanfattning

För alla organisationer som utövar direktmarknadsförning finns ett behov av att förstå hur människor reagerar på kommunikationsmeddelanden som skickas. Sedan introduktionen av dataskyddsförordningen har högre krav ställts på hur företag använder sig utav persondata och till vilken utsträckning personer ska kunna påverka hur deras personliga information används. Exempelvis ska det alltid finnas en möjlighet att avregistrera sig från direktmarknadsföringsutskick, detta kallas att Opta-ut. När en kund Optar-ut förlorar organisationen möjligheten till att skicka personaliserade erbjudanden till den individen och det resulterar i förlorade intäkter.

Målet med det här examensarbetet är att undersöka Opt-ut fenomenet och bygga en model som kan prediktera risken av framtida Opt-uts. Risken för Opt-ut mäts som den estimerade sannolikheten för att kunden kommer Opta-ut inom den närmaste tiden. Projektet använder sig utav maskininlärningsalgoritmer på aggregerad kommunikationsdata och kunddata. Av alla algoritmer som testades var Extrem Gradient Förstärkningsalgoritmen med simulerade variabler den bästa och mest robusta. Resultatet av modellen kan bäst beskrivas med en AUC på 0.71 och en lift på 2.21 på data två månader framåt i tiden från då modellen tränades. När man använder en modell med simulerade variabler går beräkningskostnaden upp, men med en signifikant ökning i prestanda. När information om specifika kommunikationer inkluderas i prediktionerna kan slutsatsen dras att det är signifikant för prestandan. En förstärknings method som Extrem Gradient Förstärkningalgoritmen genererar stabila resultat vilket kommer innebära längre tid mellan gångerna man behöver träna om modellen.

Acknowledgements

We would like to thank our supervisors David Norell and Morgan Elfvin at Telia. Your guidance and support in the area of data science has been of great value for us and the outcome of the thesis project. You made us feel welcome at Telia and assisted us in creating a vision for the project and how to best create value for the business.

Thank you to our collugues at Telia Company. Firstly our manager Johan Selling, who supplied us with the resources needed to complete the project. Secondly Atie Daglawi, without your support the complex PySpark library would have been less intuitive. Thirdly, the CRM-team for your insights within the area of customer relations and marketing. Your continuous positive attitude and support helped the project forward. Fourtly, the advanced analytics and visualisation team for your interest in the project and support during our time at TeliaCompany.

Finally, we would like to thank our supervisor at Umeå University Peter Anton. Your support has been important for the academic parts of the project, reviewing the the report and attending the weekly meetings discussing relevant theory and approaches.

Contents

1 Introduction

1.1 Disposition of the thesis

The thesis is divided into several parts and presented first is a short background of the host company and the project with its restrictions. Second, the data structure, method of procedure and modelling is presented, where the model selection process is also described. Third, results from a few models are presented in relation to a selected baseline. Important metrics of performance are showed. Lastly a discussion about results, methods, relevant concepts and future action is presented.

1.2 Background

As part of the work that Telia Company, hence forward called The Company, does in regards to customer insight, there is a need to understand why people unsubscribe from their communication channels, this is called Opt-Out. At the start of the project there was no risk assessment to estimate probabilities of which customers are likely to Opt-Out, so there was a demand for risk assesment. When customers Opt-Out they unsubscribe from the communication channels that The Company is able to reach them on. Since the introduction of the General Data Protection Regulation (GDPR) in 2018, companies that have customers in the EU region are obligated to inform customers that they store information about them and what this information is to be used for (Dataskyddsförordningen, 2018). Hence people are more aware of their digital footprint and have the possibility to control the amount of digital marketing they received.

With information about personal characteristics and behaviour, The Company is able to formulate and market personalised offers to their customers. If customers no longer wish to receive information and offers from The Company, they have the option to Opt-Out from the communication channels. The Company will not erase information about the customers since it will need to charge customers for their services and distribute information regarding their service that The Company is obligated to inform about, for example a subscription fee change. Since The Company wishes to sell additional services that are relevant to its customers it wants to retain as many people as possible in its marketing channels. This in turn means that the Opt-Out rate is desired to be as low as possible.

1.3 Company description

The Company is an international telecom company with business mainly in the Nordic region. Its core area is networks and it offers services within telecoms and broadband. The company was founded in 1853 and is listed on Nasdaq

Stockholm and Nasdaq Helsinki. It has approximately 20 000 employees and a revenue of 80 000 million SEK. The customer base consists of 23,1 million subscribers where Sweden is the largest market with 7,5 million subscriptions according to Telia Company AB (2017).

1.3.1 Advanced Analytics & Visualisation Team

The Advanced Analytics and Visualisation Team performs both extensive and ad hoc analytics projects all over The Company's organisation. They handle the more advanced analytics projects that require greater knowledge in areas such as mathematical statistics and data science. The team has supervised the thesis workers in their daily work and provided insights related to the technical aspects of the thesis project.

1.3.2 Customer Relationship Management Team

The Customer Relationship Management team, hence forward called the CRM team, is a team at The Company which is responsible of the customer relations in a higher perspective compared to a customer service team. The CRM team creates and analyses customer campaigns that are distributed through digital communication channels such as SMS, E-mail and Social media marketing. An important area for the CRM team is customer retention, where preventing customer churn and Opt-Out is a key part. The CRM team provides the thesis workers with insights in working processes related to digital marketing and suggestions of features to investigate.

1.4 Aim

The thesis project consists of two main goals presented by The Company.

1. Create machine learning models that set scores to predict future customer Opt-Out in a finite time frame. The final model is planned to be implemented by The Company to asses risk related to Opt-Out on a weekly basis.

2. Find the underlying factors that drive customer Opt-Out, mainly through data visualisation and feature analysis.

1.5 Project Scope

1.5.1 Confidentiality

The thesis project is written under confidentiality. The thesis workers have signed a confidentiality agreement where they have undertaken not to reveal information of confidential or secret nature related to The Company's business or customer data. All information and results presented in the thesis report have been approved by The Company before being published.

1.5.2 Limitations

Due to the restricted time frame and the complexity of the project some limitations are set. These limitations could in the future be amended to achieve additional insights and knowledge about customer retention, although it is not in the scope of the project. The following limitations are set:

- The thesis project only focuses on transaction data for SMS & E-Mail communication as they are the only communication channels were customer can Opt-Out from marketing.
- The thesis project only focuses on marketing for business to consumers (B2C) excluding (B2B) as B2C communication has less customised communications. This makes the communication less complex to analyse.
- The thesis project limits the amount of personal data that is used. Only information about age, type of services and time related data such as for how long they have been a customer will be used.
- The set time frame for the data used is 2017-04-01 to 2019-03-31.

2 Method

In this section a description of the method that was implemented in the project is given. An overview of the data and how the data was processed before classification. After that model setup and performance are presented followed by the procedure for identification of important Opt-Out variables.

2.1 Data Description

In this section a description of the data used in the project is presented. The Company provided customer communication data for B2C transactions through different channels over the period of September 2012 to December 2018. Moreover, the database includes tables connected to the communication data through different keys. For example, such tables includes customer, campaign and Opt-Out information. An overview of the database structure is shown in the star scheme in Figure 1.

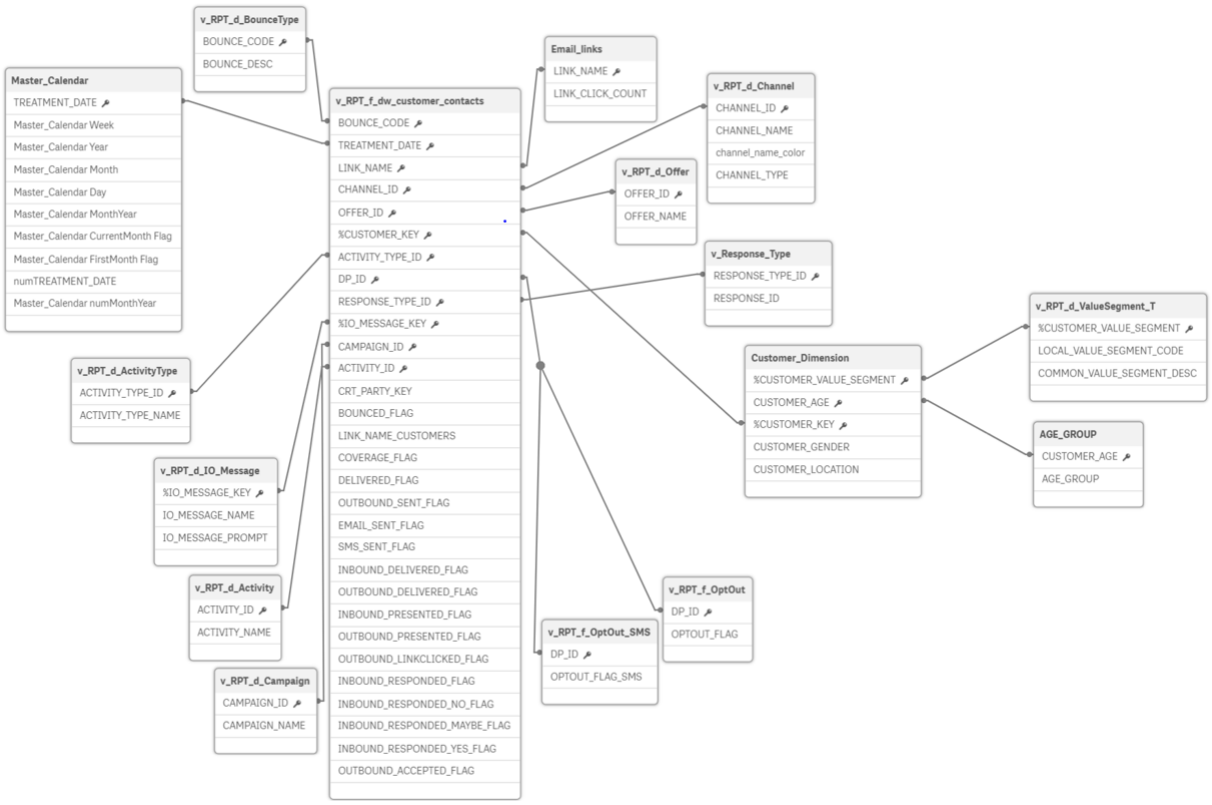

Figure 1: Star-scheme of the data set.

As seen in Figure 1 the data that the project is built upon is divided into several parts and the four main parts are described below in further detail.

2.1.1 Communication Data

The communication data covers all data related to communication that is done towards a customer. Each row in this data set is a specific message related to information about the message. For example campaign id, treatment date and anonymised personal id of a specific customer. An example of this data set is visualised in Table 1 and is called matrix **U**. It is the centre piece in Figure 1. From the matrix **U** two other matrices are constructed, **X** and **Z**. There are also 41 other columns in this data set - not all relevant to the analysis.

Table 1: Example of the communication data matrix **U**.

Treatment Log ID	**DP ID**	**Treatment Date**	...	**Customer Key**
45681	887792	2018-01-03	...	Key 40
45682	887792	2018-02-08	...	Key 40
45683	423478	2018-03-10	...	Key 1
⋮	⋮	⋮	⋱	⋮
n	235478	2018-05-23	...	Key 87

The matrix **U** contains the explanation data on a treatment level, meaning that each row in the model data set represents a single communication to a customer. Each treatment in **U** is connected to a dialogue key, called DP_ID.

An issue with the communication data is that when a communication contains several offers, each offer will be logged as a specific row in the data. To avoid overlap in the model data matrix, **U**, duplicate rows that contain the same dialogue key on the same day are removed, see Figure 2. This is important since Opt-Outs are connected through the dialogue key and when Opt-Out information is joined on the model data, the actual Opt-Out is only connected to one communication. To avoid that the Opt-Out is joined on multiple instances of the dialogue key, the join is performed on the maximum date of that specific dialogue.

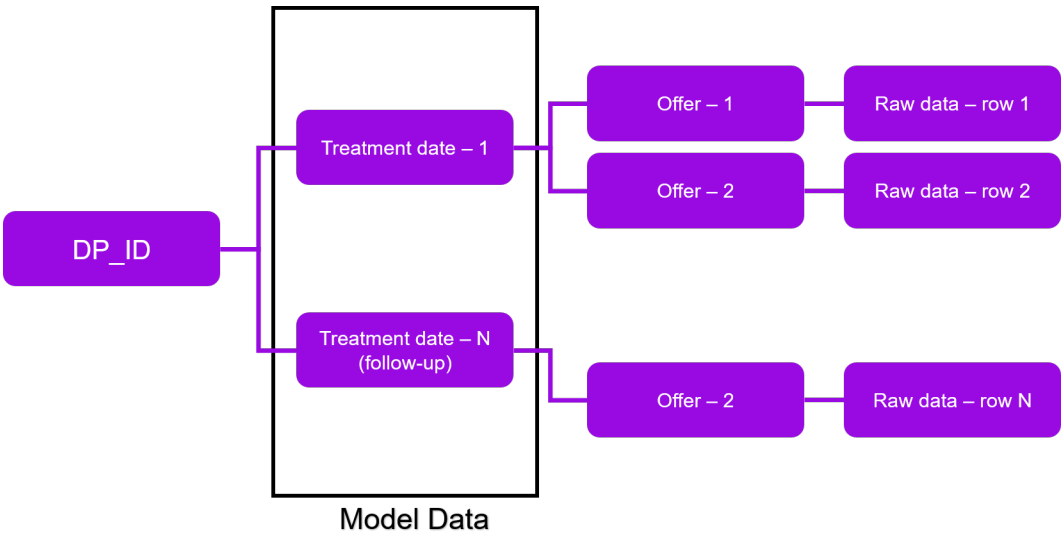

Figure 2: Flow-chart of DP_ID in relation to how offers and dialogues are presented in the raw data. The black-box around treatment dates represents how DP_IDs are presented in the model data matrix. If an Opt-Out has occurred only the last treatment in the dialogue will have one as the response.

2.1.2 Personal Data

The personal data that the project has access to covers, but is not limited to, the age of the customer, what type of services they have, the region they live in and the anonymised personal id of each specific customer. The personal data is visualised as the Customer Dimension data set to the right in Figure 1 and an example of the data set is seen in Table 4. Every customer has related information that The Company stores in order to keep track of all their customers and the products and services they have at The Company. The personal data for each customer are generally updated every month, meaning that the communication training sets including more than one month are required to be merged both on customer key and a monthly timestamp to avoid data leakage.

2.1.3 Campaign Data

The campaign data set contains information of the campaigns and is connected to the dialogue data which is used to search for important information contained within each dialogue that is sent to the customer. The information in the campaign set is campaign name and keys. The campaign data set is visualised in the star scheme to the left of the centre in Figure 1. There were no structure of campaign groups or clusters present, so these groups were created manually with the help of the CRM team. This was done in order to investigate the influence of campaign types such as newsletters or price updates.

2.1.4 Opt-Out Data

This data set contains information about the customers that have Opted-Out from The Company's marketing campaigns. The information in the data set consists of data and times when the customer executed the Opt-Out, a dialogue key which can be connected to a specific dialogue and a customer key that can be connected to a specific individual. The Opt-Out data set is visualised to the right in Figure 1 and connected with the communication data through DP-ID. The number of Opt-Outs are significantly small in comparison to the number of communications sent. An Opt-Out is a rare occurring event which result in a heavily imbalanced response.

2.2 Model Data Matrices Overview

The goal of the project is to model the response (Opt-Out) based on the explanatory data which consists of communication data, personal data and campaign data, all described in section 2.1. The response vector for Opt-Out is defined as $\mathbf{Y}$ with binary classification values of 0 and 1, where 1 represents Opt-Out and 0 represents no Opt-Out. The scores of Opt-Out that are used in the risk assessments are calculated from the probabilities of observation belonging to class 1 based on the different models used. Table 2 shows an example of how the response vector $\mathbf{Y}$ is constructed.

Table 2: Example of the response vector $\mathbf{Y}$.

	Y
DP_ID	Opt-Out
1	0
2	1
⋮	⋮
n	0

From the communication data matrix $\mathbf{U}$ the matrix $\mathbf{X}$ is constructed by reducing the matrix to a subset of features, which were decided upon when evaluating important features in the communication data set. They are called simulated features since they are observation specific and unknown up until the point where the communication is sent. So the features are a future outcome for new observations but known in the case of training and testing data. The simulated variables in the training data set consists of the original combinations of the observations. For the live scoring these combinations will be simulated for each customer. An example of the matrix $\mathbf{X}$ is visualised in Table 3.

Table 3: Example of the simulated data matrix **X**.

DP ID	Channel ID	Weekday	PeriodofMonth	Customer Key
887792	1	3	1	Key 40
887792	1	5	1	Key 40
423478	4	2	3	Key 1
⋮	⋮	⋮	⋮	⋮
235478	4	2	3	Key 87

The demographics and subscription information for each customer is described in the data matrix **W**. As these change over time the table is built on snap shots for each month for each customer. Since the treatment data in **X** spans over several months back it is not applicable to always use the latest customer information, as customer demographics change over time. Instead we use a monthly snap shots from **W** to match the treatment in **X** with the latest information at the time of treatment occurrence. This means that **W** is merged with **X** on both customer key and treatment period to obtain desired results. An example of matrix **W** are shown in Table 4.

Table 4: Example of the personal data matrix **W**.

Customer Key	Customer Age	...	Total Number of Services
Key 1	20	...	1
Key 2	40	...	4
⋮	⋮	⋱	⋮
Key n	67	...	0

To investigate the impact of former events for the probability of Opt-Out, historic frequency variables and rates are constructed from matrix **U** and forms the data matrix **Z**. These variables measure the customers treatment interactions over bounded time windows for each treatment, meaning that the historic frequency variables are calculated for all rows in **X** independently. These time windows also contain variables that show how specific customers have reacted to previous communication during the specified time period. A visualisation of the matrix **Z** with historic frequency features and rates is shown in Table 6 followed by a visualisation of example features in table 5.

Table 5: Visualisation of the data matrix $\mathbf{Z}$*, each column represents different time intervals with the same historic features and rates. For instance* Z_7 *displays features and rates 7 days back from the specific fictive historic scoring date.*

	Historic Features & Rates				
Customer Key	Z_7	Z_{30}	Z_{90}	Z_{180}	Z_{360}
Key 1					
Key 2					
⋮	⋮	⋮	⋮	⋮	⋮
Key n					

Table 6: Example of the historical feature matrix $\mathbf{Z}$*.*

Customer Key	**Newsletter rate 360 Days**	...	**Sales rate 360 Days**
Key 1	0.3	...	0.08
Key 2	0.7	...	0.06
⋮	⋮	⋱	⋮
Key n	0.5	...	0

By modelling the response data vector $\mathbf{Y}$ as a function of the three explanation data matrices $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{Z}$, a function $\mathbf{f}$ is obtained. This function represents the machine learning methods that estimates the probabilities of Opt-Out, see equation 1. The methods used for obtaining the probability estimates are defined in Appendix A.4 and A.5.

$$\hat{\mathbf{Y}} = \mathbf{f}(\mathbf{X}, \mathbf{W}, \mathbf{Z}) \tag{1}$$

2.3 Classification Setup

The implemented model will score customers on a weekly basis by estimating the probability of Opt-Out. This allows the user to screen customer Opt-Out scores and filter target groups before sending a treatment. As the data processing and modelling are computationally expensive and there are some delays when writing data to the system it is not possible to score customers each day. Therefore all observations in the data set are scored on a date a week before the treatment is sent, known as the scoring date. This is initially done on a weekly basis. This is only done in the training phase and when the model is implemented the scoring date will be the current date.

The setup for training the classification model can be divided into five parts.

1. Data collection and processing
2. Train, Validation and Test split
3. Undersampling the training set

4. Train on training set and Tune the model with validation set

5. Testing the best model on test set

2.3.1 Data setup

To successfully estimate probabilities of Opt-Out the data is divided into several parts with a few different purposes. There are two main areas of partitioning, which is train-test split and sampling on imbalanced classes. So the data will be divided into a training set, validation set and test set. Noticeable is that the partitioning is done after merging the data matrices **X**, **W** and **Z**, meaning that no historic information is lost for any observation.

Most machine learning methods use the property of splitting data into a training and validation set to evaluate performance. For instance, it is common to randomly split a data set into 80% training and 20% validation data with corresponding response in a predetermined time frame (Hastie et al., 2009, p. 247-248). However, since the goal of the thesis project is to predict the future risk of Opt-Out this type of split will not be sufficient. Instead the training and validation data are divided with respect to time. The Opt-Out problem is, as Ballings and den Poel (2012) states for churn prediction, a three dimensional one. Where the Time window is the third dimension in addition to Data and Algorithm. For the model performance to be sufficient one need to consider the optimal time period setup with respect to the event history. There are several different configuration for the time window and from different test it was decided that the training data should consist of observations from a twelve month period. For each observation in that twelve month period frequency variables are calculated for up to 360 days back, as described in Table 6. The validation data consider the immediately following month after the training period for performance evaluation. The period length of the validation data is adjustable to a minimal length of one week as observations are independent of future scoring dates. The test data consist of all periods following the validation period. The setup can be viewed in Figure 3. With this setup it is possible to live score the customers on a weekly basis and use the estimated probabilities in the business.

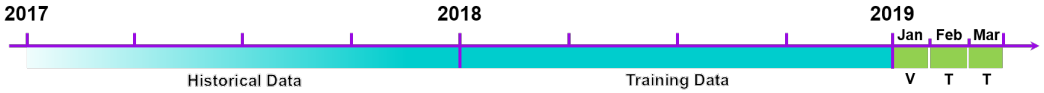

Figure 3: A visualisation of the data setup for training the model. The historical data part is shown since the first observation in the training data need a 360 day historical window for calculating features. The V and T represent validation data and test data.

As the response data vector **Y** is heavily weighted towards one class there is a need to re-sample the training data to create a more stable model. If the training data is not re-sampled there is a risk of overfitting (Appendix A.2) and the models will probably not predict any Opt-Outs at all, which would be inferior in relation to the thesis project goals. Therefore, it is of high importance to re-sample the data to obtain desired results. The two approaches when it comes to imbalanced learning are over- and under-sampling, these are defined in Appendix A.6. Under-sampling of the majority class was used to stabilise the training data by the `imblearn` library in Python. The usage of under-sampling resulted in a data set with desired balance combined with an acceptable computational cost. Both over-sampling and a combination of the two methods was tested, but under-sampling proved to be the best fit for the project due to the amount of data and performance requirements.

2.3.2 Used Classifiers

Three different classifiers were used to estimate the risk of Opt-Out. First a single Decision Tree, defined in Appendix A.4.1, was fitted as a baseline model and secondly a Random Forest classifier, defined in Appendix A.4, followed by the third classifier which was an Extreme Gradient Boosting model, defined in Appendix A.5 and from now on called XGBoost model.

The three models were trained with and without the presence of matrix **X** and then compared between the two setups. The reason behind this is that the variables in **X** are observation specific and can be viewed as simulated variables since the feature value is only set once the communication is sent to customer, making scoring hard. These features are called simulated variables since they are possible future outcomes, but are in reality unknown until the actual content is created. The models were trained with and without these features to show the impact of the simulated variables. Not only the independent impact but also the interaction effect with historic features in **Z** and customer demographics in **W** were considered. How this was solved in respect of scoring is explained in detail in Section 2.3.4.

The models were then evaluated by looking at the ROC-AUC and Lift score described in A.1.3 and A.1.2 respectively. Model metrics from the confusion matrix described in A.1.1 was also used to evaluate performance. The estimated probabilities for the training periods was visualised through distribution plots to see separation of estimated probabilities between the two classes.

2.3.3 Tuning

The Random Forest and XGBoost model were cross validated using K-Fold Cross Validation with 5 folds with a random stratified approach as described in Appendix A.3. Cross validation is used to analyse the robustness of the trained model. When performing the 5-fold cross validation the model was

validated on the in-sample random fold as well as the out-of-sample validation period. From these two validations we could compare in-sample scores and out-of-sample scores. The goal was to maximise the AUC but at the same time avoid overfitting. A similar AUC score in-sample and out-of-sample will prevent the model to overfit. In line with Figure 3 the models were validated with the period of January 2019 as out-of-sample validation data.

2.3.4 Scoring and model retraining

As scoring customers on a weekly basis was one of the main goals of the thesis the set up of scoring is important. A scoring pipeline was created that processes data and scores chosen customers on the best model trained. However this caused some difficulties as the data setup for scoring had to be constructed differently than the one for training. For instance if the Company wants to estimate the probability of Opt-Out before sending a specific treatment to a customer, the features in $\mathbf{X}$ will not be present. As observations in this matrix are observation specific and not specified until communication is sent, meaning that it is not possible to live score. To solve this when scoring, all combinations of a subset of features in $\mathbf{X}$ are scored as separate observations. The models was also trained without the features in $\mathbf{X}$ in order to compare the performance with or without the observation specific features. The model will be retrained on data from more recent periods once the model starts lacking performance in prediction of the test data. The frequency of scoring and retraining are adjustable which leads to flexibility in usage of the model.

The scoring is done by the following key steps.

1. Collect and process data ranging from the current date and 12 months back
2. Load the trained model
3. Score on all possible combinations

When the model starts to perform less accurate, retrain the model by following the steps for training the model as described in Section 2.3.

2.4 Important variables for Opt-Out

To investigate what drives customer Opt-Out the main method used is the feature importance calculated by the ensemble methods. For the Random Forest model and the XGBoost model the feature importance is based on the Gini impurity and entropy. Though the XGBoost model also weights the split in line with its effect on the final prediction.

2.5 System Tools

Through out the thesis project, **Python** has been the central programming language used. Data was initially imported through CSV-files with anonymised customer data that the supervisor at The Company had extracted. Later on the data was extracted with **SQL** queries connected to The Company's databases. At first the data was processed using the **Pandas** library in Python which worked well for the extracted data. However, when access to The Company's databases was approved the amount of data used exceeded the performance of the used computers, causing memory errors. To handle the large amount of data the code was rewritten to work with the **PySpark** library which work against a shared cluster instead of a local machine. PySpark solved the issue with memory error and made the data processing and preparation significantly more effective. However, the complexity of the library resulted in many challenging problems through out the project.

The PySpark libraries outperformed Pandas when it came to data processing and feature engineering, however it was not feasible to perform modelling with Spark. The fact that the thesis project involved some configurations of the algorithms which were not available in Spark machine learning libraries combined with the high complexity of the present configurations, libraries with Pandas as base were used for modelling instead.

As for modelling and tuning **Scikit-Learn** was the primary library used to create the random forest model. Scikit-Learn is highly compatible with Python libraries such as Pandas and is the most used modelling library in Python resulting in a user friendly experience. The Extreme Gradient Boosting model was modelled with the library **xgboost** and tuned with the help of Scikit-Learn. XGBoost has many computational advantages compared to Scikit-Learn as it uses parallel computing, optimised cache settings and out of core computing to optimise memory allocation. Tools used for data visualisation were mainly **Matplotlib**, **Seaborn** and **Scikit-plot** as these libraries are compatible with the previous mentioned libraries for modelling.

3 Results

The results in the report are based on data from the start of January 2018 to March 2019. The time frame is fixed for all models to streamline the comparison and interpretation of the different approaches. The training set consists of re-sampled data from the period of January 2018 to December 2018. The validation test data set are the following month after the training period (January 2019), while the test data are the second (February 2019) and third month (March 2019) after the training period. A visualisation of the data structure is shown in Figure 3.

The results for the three models with simulated variables are displayed for both testing periods February and March, hence referred to as the **complete model** ($\mathbf{Y} = \mathbf{f}(\mathbf{W}, \mathbf{X}, \mathbf{Z})$). As mentioned in Section 2.3.4 all models are also trained without the simulated features in matrix $\mathbf{X}$, this setup is referred to as the **reduced model** ($\mathbf{Y} = \mathbf{f}(\mathbf{W}, \mathbf{Z})$) is presented after the complete models. As the observations in $\mathbf{X}$ are simulated features, they are unknown when scoring.

The performance of classification depends on the threshold chosen. As different models might estimate the distribution of probabilities in different magnitudes the threshold is adjusted in order to find 50 percent of the Opt-Outs for the Random Forest and XGBoost model. The threshold is held at 0.5 for the Decision Tree model, as its estimated probabilities are binary. The confusion matrices defined Appendix A.1.1 are normalised by dividing the number of classified observations in a category with the number of observations in each true class separately. This is done to enhance interpretation of the imbalanced response. The models are also evaluated by looking at the lift score, defined in Appendix A.1.2.

3.1 Decision Tree

The decision tree models estimated probabilities are binary as it splits until class purity, meaning that the estimates will either be 0 or 1. Hence plotting a distribution of probabilities is irrelevant, but the share between classes is more interesting. Since the estimated probabilities are binary the classification is independent on the threshold set.

3.1.1 Complete Model

The baseline model is a decision tree without specified parameters. This means that there are no specified tree depth with splits based on the Gini criterion, see Appendix A.4. The ROC-curve and AUC Score described in Appendix A.1.3 for the two training periods are shown in Figure 4.

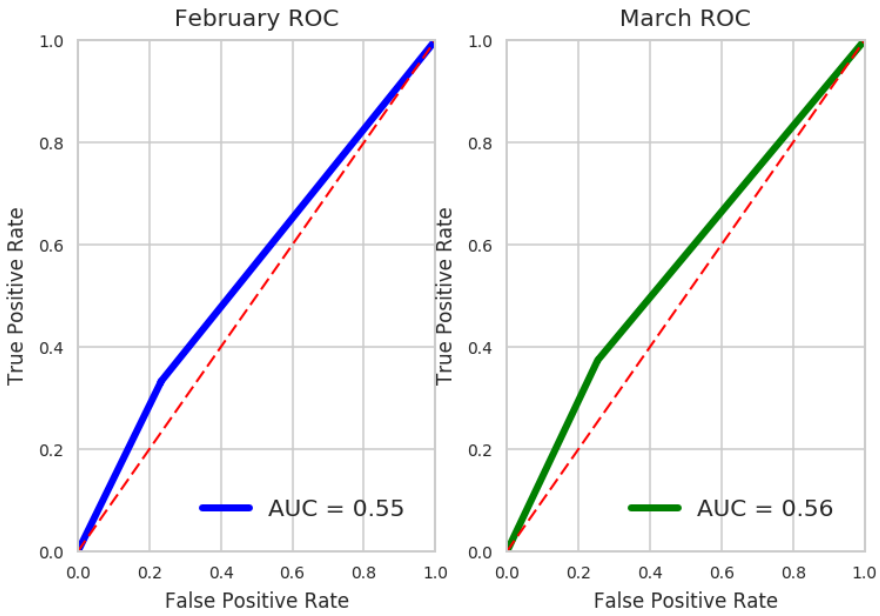

Figure 4: Receiving Operating Characteristics (ROC) for the complete decision tree model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The normalised confusion matrix and lift score for the complete decision tree model is presented in Table 7.

Table 7: Normalised confusion matrices and the lift score for the complete decision tree model.

		Predicted Class			
		February		March	
		0	1	0	1
True Class	0	0.77	0.23	0.76	0.24
	1	0.65	0.35	0.65	0.35
Lift Score		1.49		1.45	

3.1.2 Reduced Model

The reduced decision tree model has the same set up as the complete model but with the simulated features in **X** removed. The ROC-curves for the reduced model in periods February and March are shown in Figure 5.

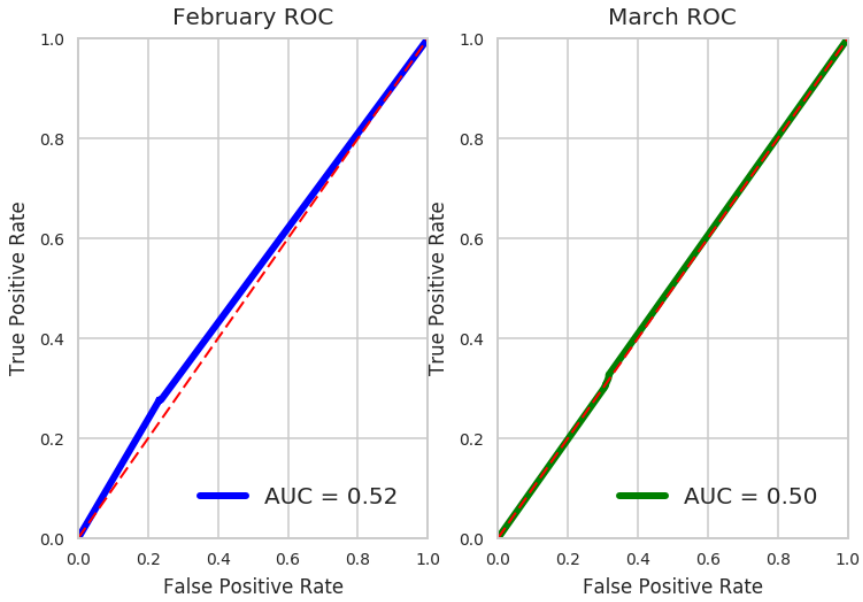

Figure 5: Receiving Operating Characteristics (ROC) for the reduced decision tree model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The normalised confusion matrix and lift score for the reduced decision tree model is presented in Table 7.

Table 8: Normalised confusion matrices and the lift score for the reduced decision tree model.

		Predicted Class				
		February			March	
		0	1		0	1
True Class	0	0.77	0.23		0.69	0.31
	1	0.72	0.28		0.70	0.30
Lift Score		1.19			1.00	

3.2 Random Forest

The tuning of the Random Forest model resulted in a parameter setup with 250 trees. Further, the max depth was set to 10 and the minimum samples required for splitting were also chosen to be 10. An important setting to achieve desired result was for the class balance parameters to match the balance of the training data set. The threshold chosen for classification is dynamic between test periods, but to make comparison both models thresholds will be set at a limit where approximately 50 % of the Opt-Outs are classified correctly.

3.2.1 Complete Model

The distribution of estimated probabilities for complete random forest model in testing periods February and March are shown in Figure 6. The distribution differs between the two training periods where March in general estimates lower risk of Opt-Out than February. It is possible to see separation between classes as the model estimate in some sense estimate lower probabilities for the no Opt-Outs and the actual Opt-Outs has higher probabilities. Note to say is that there are overlapping between classes, meaning that the separation are not perfect.

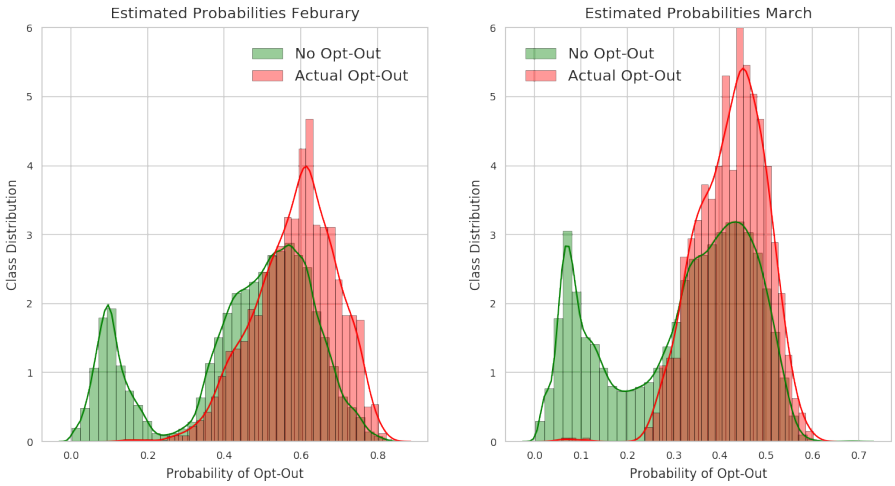

Figure 6: Estimated Probabilities for the complete random forest model. The kernel density functions for the class distribution are represented by the solid lines.

The ROC-curves for the two training periods are shown in Figure 7. As seen in the ROC-curves, February has more curvature and higher AUC score than March, meaning February it separates classes slightly better. This could imply that the model lose performance as the time from retraining increases.

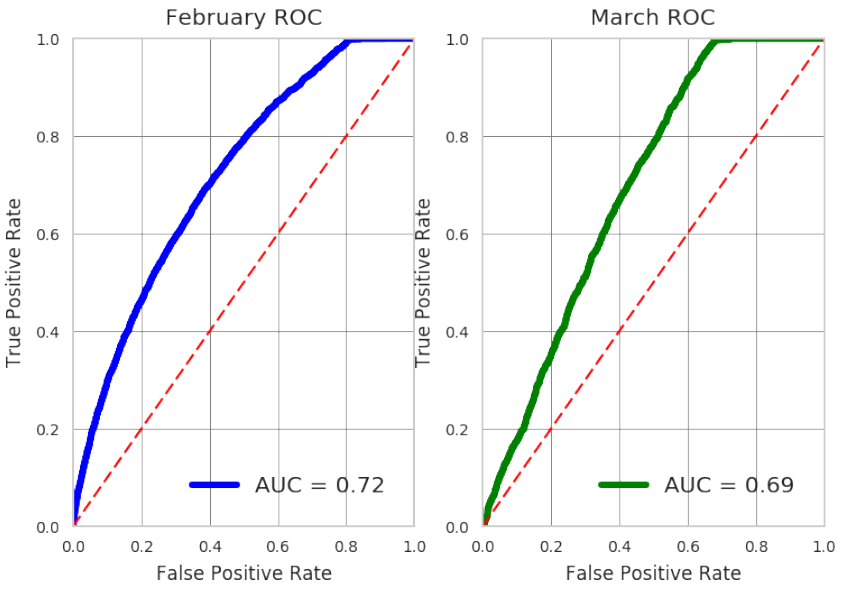

Figure 7: Receiving Operating Characteristics (ROC) for the complete random forest model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The classification of Opt-Out is dependent on the threshold selected for the estimated probabilities. To find a desired amount of Opt-Out (approximately 50 %) two different threshold were chosen for the two testing periods. The confusion matrix of the classifier in the two testing periods combined with lift score and chosen threshold are shown in Table 9.

Table 9: Normalised confusion matrices, threshold and lift score for the complete Random Forest model.

		Predicted Class			
		February		March	
		0	1	0	1
True Class	0	0.78	0.22	0.73	0.27
	1	0.50	0.50	0.50	0.50
Threshold		0.60		0.43	
Lift Score		2.25		1.83	

3.2.2 Reduced Model

The reduced random forest model has the same parameter tuning as the complete model but without the simulated variables in matrix **X** in the feature space. The distribution of estimated Opt-Out probabilities for the reduced Random Forest Model are shown in Figure 8. In comparison with the distribution of the complete model (Figure 6) the distributions overlap more, meaning that the reduced model performs worse in class separation.

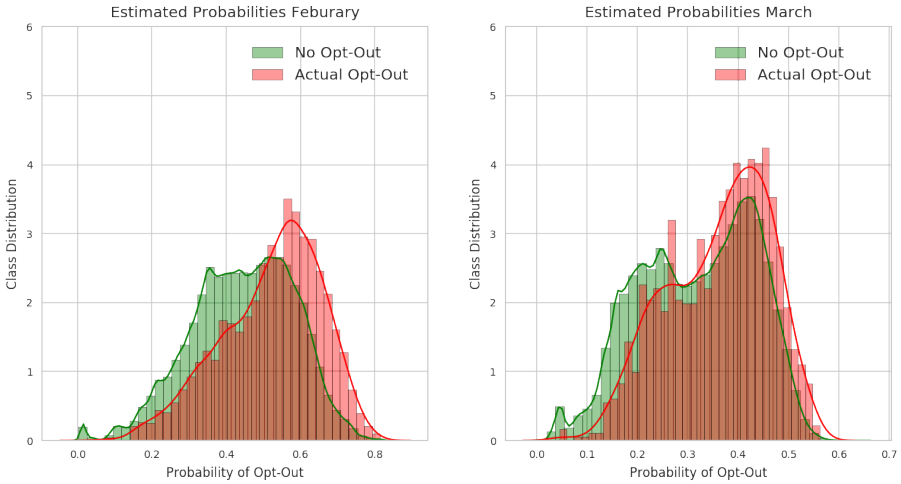

Figure 8: Estimated Probabilities for the reduced random forest model. The kernel density functions for the class distribution are represented by the solid lines.

As seen in Figure 9 there is less curvature in the ROC curve for the reduced model. This also means that the AUC scores are lower resulting in worse class separation.

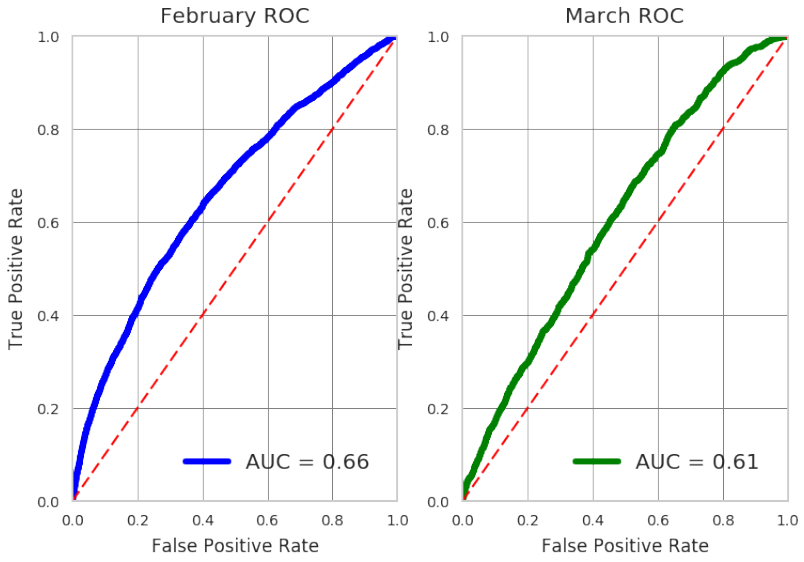

Figure 9: Receiving Operating Characteristics (ROC) for the reduced random forest model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The confusion matrix, threshold and lift score for the reduced random forest model are shown in Table 10.

Table 10: Normalised confusion matrices, threshold and lift score for the reduced Random Forest model.

		Predicted Class			
		February		March	
		0	1	0	1
True Class	0	0.74	0.26	0.70	0.30
	1	0.50	0.50	0.50	0.50
Threshold		0.55		0.38	
Lift Score		1.90		1.36	

3.2.3 Important features for predicting Opt-Out

The feature importance of the random forest model is calculated with the Gini impurity criterion. In Figure 10 one can see that the complete random forest model is highly dependent on the type of channel the communication is sent through. Whilst the reduced random forest model has a more even distribution of important variables. One can also observe the standard deviation on the feature importance of each feature among all trees. The importance could have been large in one random tree and lower in another. It can also be seen that the standard deviation is large for the reduced random forest model.

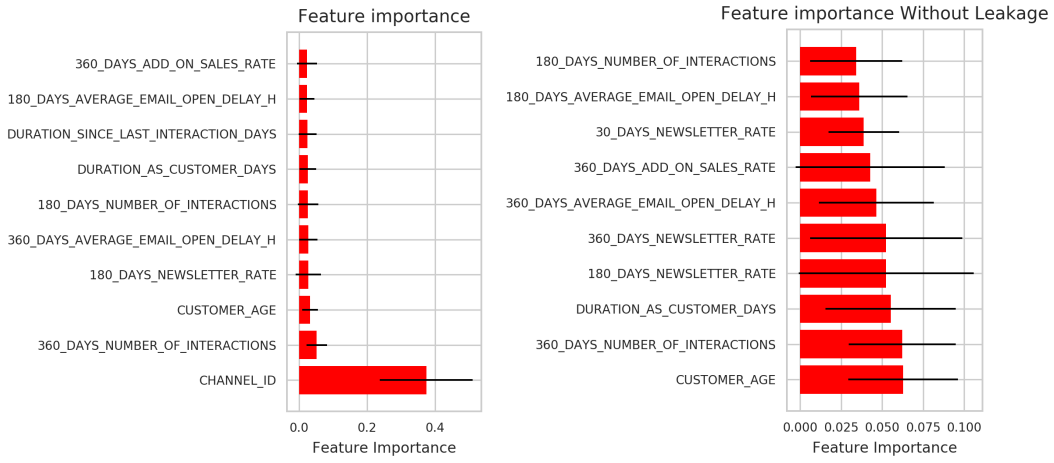

Figure 10: The feature importance of the complete random forest model and the feature importance of the reduced random forest model. The reduced model does not contain any simulated variables (i.e. leakage variables) The black line represent the standard deviation.

3.3 Extreme Gradient Boosting

The Extreme Gradient Boosting model, from now on called XGBoost, was setup with 250 trees after tuning with 5-fold cross validation. The max depth was set to 5 with a feature and observation specified sub-sampling of 0.6, meaning that 60 % of the features and 60 % of the observations will be included when growing each tree. The XGBoost model was regularised with regularisation parameter $\gamma = 0.3$ and a set learning rate of $\eta = 0.1$. The model was balanced based on the response balance in the training data.

3.3.1 Complete Model

The estimated probabilities of the complete XGBoost model are shown in Figure 11. In comparison to the Random Forest model the XGBoost model estimates low probabilities for a large part of the communication that are not actual Opt-Out. By looking at the bar to the far left of the distribution in Figure 11 one can see that a large part of the distribution for Class 0 have estimated probabilities close to zero. The class distribution still overlap but at the same time the XGBoost models probability distributions has less variation between the two months. This could mean that the model is more stable when estimating probabilities than the Random Forest model.

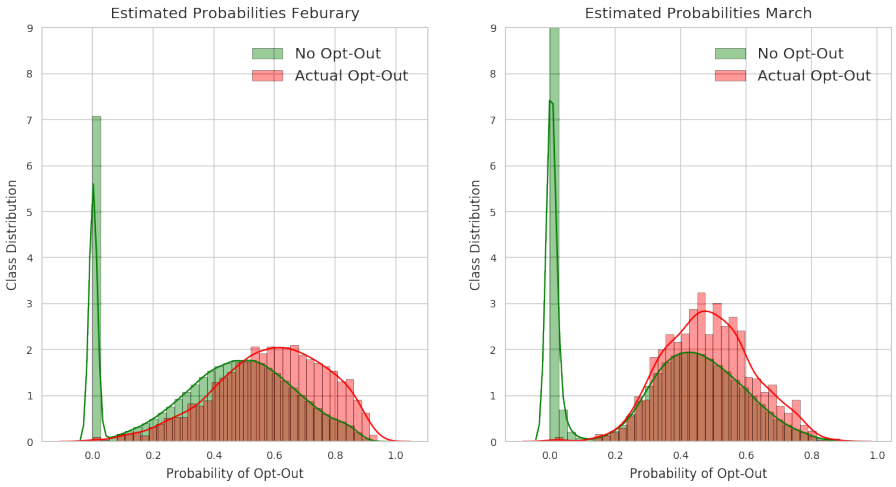

Figure 11: Estimated Probabilities for the complete XGBoost model. The kernel density functions for the class distribution are represented by the solid lines.

The ROC-curve for the complete XGBoost model is shown in Figure 12. There are less difference in curvature and AUC between testing period for the XGBoost model.

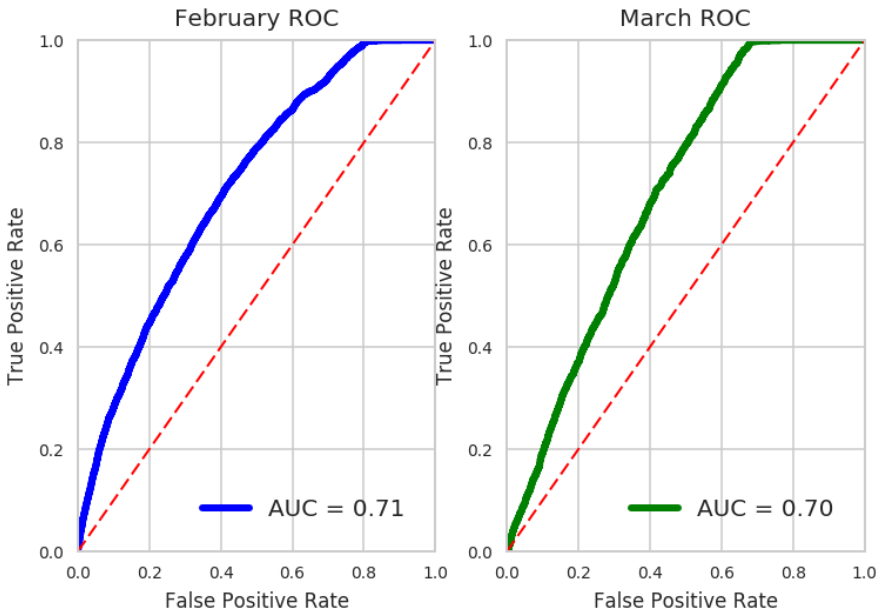

Figure 12: Receiving Operating Characteristics (ROC) for the complete XGBoost model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The complete XGBoost model classifies estimated probabilities by setting a threshold that finds 50% of the Opt-Outs for each testing period. The confusion matrix, threshold and lift score are presented in Table 11.

Table 11: Normalised confusion matrices, threshold and lift score for the complete XGBoost model.

		Predicted Class			
		February		March	
		0	1	0	1
True Class	0	0.77	0.23	0.70	0.30
	1	0.50	0.50	0.50	0.50
Threshold		0.6		0.48	
Lift Score		2.21		1.73	

3.3.2 Reduced Model

The reduced XGBoost model has the same parameter setup as the complete XGBoost model but with a decreased feature space since the simulated features in $\mathbf{X}$ are removed. The estimated probabilities for the reduced XGBoost model are displayed in Figure 13. As seen in the Figure the reduced model does not estimate probabilities for class 0 with the same certainty as the complete model. There is still signs of class separation but as before overlapping are still present.

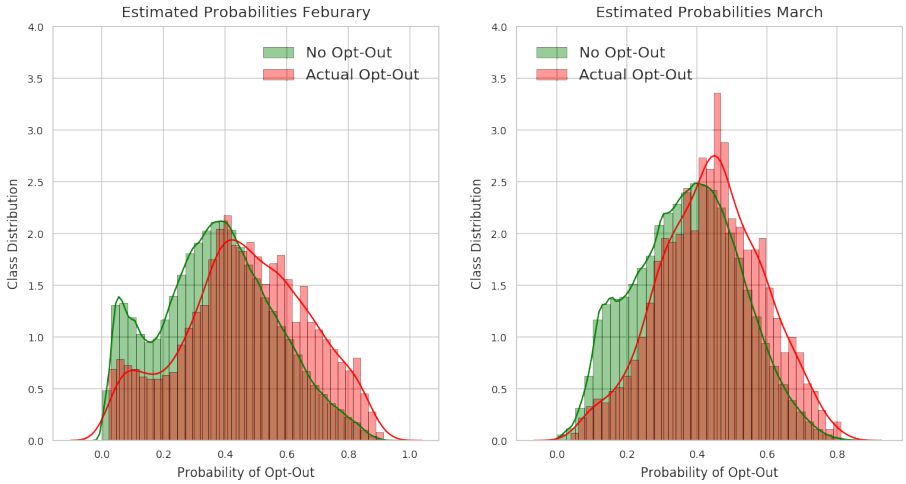

Figure 13: Estimated Probabilities for the reduced XGBoost model. The kernel density functions for the class distribution are represented by the solid lines.

The ROC-curves for the two training periods are shown in Figure 14.

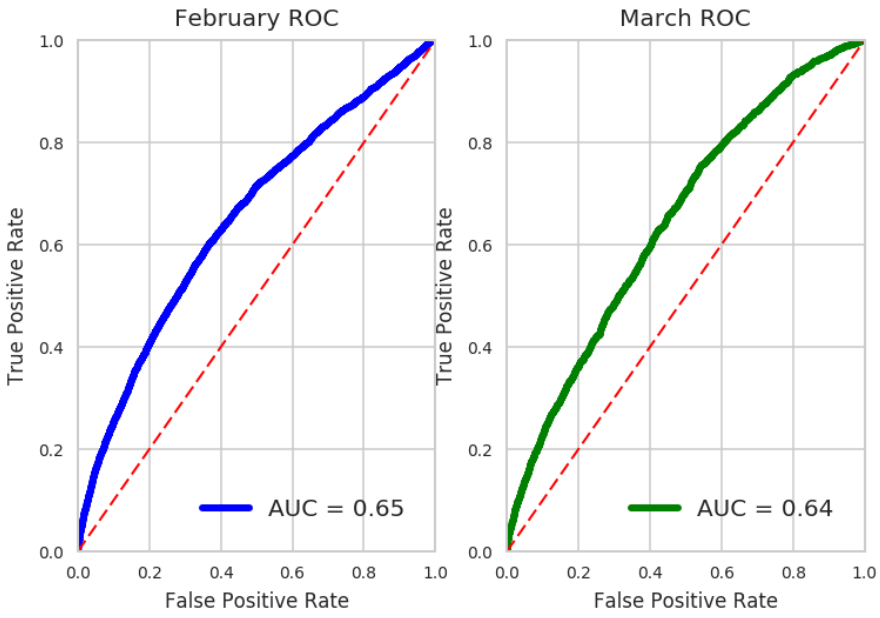

Figure 14: Receiving Operating Characteristics (ROC) for the reduced XGBoost model. The Area Under the Curve (AUC) are shown in the box to the bottom right in each plot.

The classification of Opt-Out for the reduced XGBoost model is divided by an adjustable threshold to find 50 % of the Opt-outs. The normalised confusion matrix for the model is shown in Table 12.

Table 12: Normalised confusion matrices, threshold and lift score for the reduced XGBoost model.

		Predicted Class			
		February		March	
		0	1	0	1
True Class	0	0.72	0.28	0.69	0.31
	1	0.50	0.50	0.50	0.50
Threshold		0.50		0.44	
Lift Score		1.63		1.59	

3.3.3 Importance features for predicting Opt-Out

In Figure 15 one can see that the XGBoost model has customer specific variables as the top important variables. And not the type of channel the communication is sent through as the Random Forest model. There are two important simulated variables which demonstrate the impact they have on the predictions.

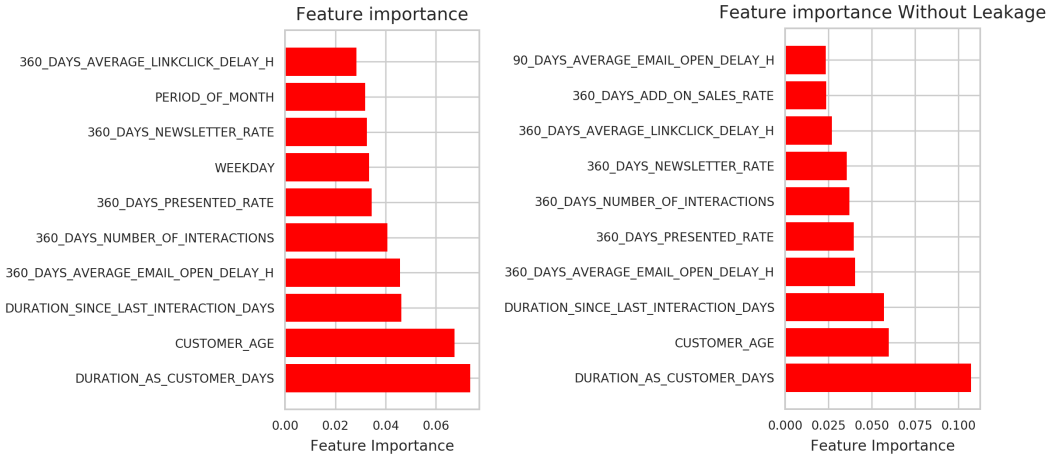

Figure 15: The feature importance of the complete XGBoost model and the feature importance of the reduced XGBoost model. The reduced model does not contain any simulated variables (i.e. leakage variables)

4 Discussion

Opt-Out can to some degree be separated with the XGBoost model but in general it is a complex subject and is most likely exposed to random events that are unknown. For example a customer might Opt-Out from several different channels at the same time, though not directly related to any content that were sent from the company that are assessing the risk of Opt-Out. Customers have committed to many firms in their everyday life and could expect some marketing actions from these companies. When a marketing action from any company finds itself bad at the customer this might lead to an Opt-Out from other companies as well. This makes the Opt-Out issue difficult to assess to a degree where the uncertainty is low among all observations. The Opt-Out is also likely related to how personalised the content of a specific message or email is to the customer in question. A likely reason for Opt-Out is when the content is far from what the customer is expecting to receive. One could argue that if a customer Opt-Out from your company channels it is still a result of the marketing actions that has been undertaken. If the content would have been better the customer would not have Opted-Out from your channels in the end, even if content from other firms are effecting the customer Opt-Out.

4.1 Data setup

The final data set that is used contains simulated variables, which means that for every probability scoring a fixed set of expected settings for the campaign is known. For example the weekday, or which period of the month the campaign is to be sent in. With the knowledge of this, there is a significant increase in the model performance. Not only by the independent significance of the features in $\mathbf{X}$, but also the interaction customer dimensions in $\mathbf{W}$ and historic features in $\mathbf{Z}$. The use of simulated variables is more computationally expensive and requires predictions on all possible combinations of settings. Though for a problem like the Opt-Out problem it is useful to take advantage of the knowledge about the campaign that is about to be created when assessing the risk of Opt-Out among the customers for which the campaign is created for. This is done through leveraging knowledge that is unknown beforehand.

The data that was used for this project consists mostly of historical features regarding the content and the type of content that have been sent to the customer during the last year. All customers have not had any communications sent to them during the last 12 months and thus less historical information is used. More historical features could have an impact on the model performance, it could capture some patterns that still are unused by the current model. An example historical features that is still unused by the model is information about retention actions which in general refers to specific favourable offers are created for the customer with the intention to retain the customer. Using historical variables that considers favourable offers and leverages this

to increase the performance of the model could benefit the class separation for Opt-Out. With information about retention this could effect the separation between groups in a way that where there are retention actions done the customer is less likely to Opt-Out. These features are not used by the current model since there is no clear id among the observations due to a wide variety of retention actions. Also if more customer specific information is used the model could improve even more. All information that could separate customers even more would generate better results. Information such as pay grade, general ability to commit to a transaction or other knowledgeable facts about the customer could have a significant impact on the model performance.

For all models that have been tested the data have been undersampled, to a rate of 20%, to get less unbalanced data. Without this sampling all tested models have underestimated the probabilities of Opt-Out to an extent such that the model did not estimate any Opt-Out at all. This is due to the fact that Opt-Out is a very rare event and among all observations the probability of Opt-Out is often small. To generate larger probabilities for Opt-Out one can make the portion of Opt-Out larger in the training data set or lower the probability threshold for when the model should classify a likely Opt-Out. In our case both methods are used to generate results that are comparable, intuitive and understandable.

4.2 Limitations

The limitations of the project could have had a large effect on the model performance outcome as the focus was on historical information and not on personal information. With more customer specific information the performance might increase. When dealing with an issue that is very customer specific such as the Opt-Out issue the relative gain of using more customer related variables would most likely benefit model performance. Other limitations that where considered, such as only considering B2C is a beneficial limitation since the difference in communication towards B2B customers is large and a model that would consider both might not perform as good and the variables would not be able to stay the same since there might be several marketing receivers at a the B2B customer. For B2B prediction a wholly separate model should be considered.

Arguments to make further limitations are also present, for instance focusing only on SMS-marketing or a specific customer group. These models might had performed better to predict Opt-Out and describe the underlying factors. However, since no similar models related to Opt-Out existed at The Company before, the project aimed to create a more general model that assess risk on a larger scale.

4.3 Model selection

Throughout this thesis project plenty of models have been considered and most of them have been rejected in favour of the tree based models presented in the result.

Since the problem worked on is of a classification nature an initial model that was considered was the logistic regression model for binary classification, which have been previously used for churn prediction in mobile markets (Hassouna et al., 2015). Our initial test with the logistic model showed that with the data setup of many categorical variables and few continuous variables the model performed very poorly. Furthermore a Generalised Additive Model (GAM) was considered (Wood, 2017). Though since the data contain plenty of categorical variables it was decided that the model would contain a substantial amount of factor variables and not continuous variables which is preferred for GAMs. With plenty of categorical variables one need to create dummy variables for each possible category of every categorical feature. This setup is needed for both the GAM model and the more complex Neural Network model which was considered in a later stage. The neural network was briefly tested and resulted in an overfitted model that did not classify any of the observations as Opt-Outs in the testing periods of February and March. The use of a neural network in a similar setting has been considered by (Adwan et al., 2014) and it was one of the reasons to why it was tested. The neural network is as mentioned a complex model where it is more difficult to interpret the significance of the variables. So with the combined considerations of hard variable interpretation and overfitting the data, the Neural Network model was rejected.

The project finally chose to look at tree ensemble methods as they are good at handling different sorts of data, reduces variance and in some magnitude decorrelates features. We chose to test a random forest model and the more advanced XGBoost model. The baseline decision tree was introduced to show the performance of a single weak learner to compare with the more complex ensemble algorithms. The XGBoost was chosen to show how an algorithm that learns for previous trees and regularises can improve the performance of prediction. When hypertuning the random forest one can observe that when increasing the maximum depth and the number of trees considered, the complexity of the model goes up and the risk of overfitting the unbalanced data grows. So by finding a good balance of complexity and regularisation by using a minimum number of samples in each node before splitting is useful for decreasing the overfitting and force the model to generalise to a greater extent. The XGBoost are similar in essence of depth and sample split, but it also adds additional elements of fitting in form gradient boosted trees and regularisation.

4.3.1 Final model

The final choice of model is the XGBoost model since it performed better than the other two models. The results from the XGBoost is comparable to that of the Random Forest model though the XGBoost model generates probabilities with less variance between test periods. One can say that the XGBoost model is more stable due to a more controlled random selection and that it does not require to be retrained as often, which results in lower computational costs. The difference in AUC score is smaller between the testing periods for XGBoost model which could mean that the XGBoost model separates classes better and does not lose performance as fast as the Random Forest model. To strengthen this further, testing should be made on new test periods to see if it follows a similar pattern of difference in AUC scores. Noticeable is that the XGBoost model without the features in $\mathbf{X}$ does not predict many probabilities near 0 for the No Opt-Out class, meaning that the simulated variables are of importance to enhance class separation.

The lift scores is comparable between the Random Forest and XGBoost models, but as this is dependent on the threshold chosen it is better to look at the AUC since it is independent of threshold. The advantage of presenting a lift score is the simplicity of the score, making it understandable for people without an background in advanced statistics.

4.3.2 Tuning

The aim with tuning the model is to minimise the difference in AUC score between training and validation periods as well as finding the best and final model. With a low difference the final model will be more robust over a longer period, thus less incentive for retraining. Though a performance gain relative to a random guess is also required for the model performance, at the cost of retraining the model when the performance decline.

When tuning the model several settings were tried and the time complexity for training the model is dependent on the amount of possible settings that one wishes to test. Meaning that a large spectrum of possible combinations will lead to a long time spent tuning the model. For example five levels on four parameters will result in 5^4 possible combinations to try which is 625 combinations. In this case it would take several days to tune. One approach to select the best hyperparameter tuning could be with the use of design of experiments and response surface methodology as described by (Lujan-Moreno et al., 2018). To set up a design of experiments would have taken a significant amount of time and the number of hyperparameters that were considered was relatively few so the choice was to use a more brute force approach for the hyperparameter tuning. The brute force approach - which means that one test all combinations of possible settings - considered 36 different combinations of hyperparameter settings and took approximately 10 hours to run.

When tuning the random forest model with few trees, a low maximum depth and high minimum sample split, the important variables change and features that consider a much shorter time frame of the historical features come up as more relevant for the model.

4.3.3 Results

All three models indicated that the presence of the simulated variables in $\mathbf{X}$ enhance the performance. The advantage of removing the simulated variables is that it lowers the complexity of the model and especially the computational costs since the pipeline setup has to score each customer for each combinations of simulated variables. For instance, assume that $\mathbf{X}$ contains 6 features with 4 different categories for each feature. Assuming 1,5 million customers to score, the number of predictions will be $1.5 * 10^6 * 6 * 4 = 36 * 10^6$, meaning that 36 million different observations need to be scored compared to 1,5 million observations for the reduced model. Perfect balance between computational cost and model performance is hard to obtain. This thesis project has tried to find an optimal weight between the two, by including the simulated variables but only using the features that have high impact on the final result to minimise the computational costs. The comparison between the complete and reduced models are important as it shows the overall impact of the features in $\mathbf{X}$ and enhance the search of an optimal setup for the feature space.

By looking at Table 11 which is the result of the final model, one might say that the result of miss-classifying 23% or 30% of the customer communications sent, to find 50 % of the Opt-Outs is a bad result of prediction. However, with the knowledge that customers only Opt-Out at a rate of around 1 in 1000 and the complexity to predict human behaviour the result is better. The lift scores shows that we predict almost double as good as random. For instance by looking at the example mentioned earlier we find nearly 50% of the Opt-Outs by miss-classifying 25 %. If this instead was done by a model predicting at random, the miss-classification would be at 50 % to find 50 % of the Opt-Outs. Figure 16 shows how much of the population (communication sent) we need to exclude to find a certain rate of Opt-Outs (class 1). For instance if we want to find 80 % of the Opt-Outs we need to exclude 50 % of the total communications sent. With this information the CRM team can adjust the threshold for estimated probabilities to treat a certain part of the customers and at the same time prevent Opt-Out at a desired level.

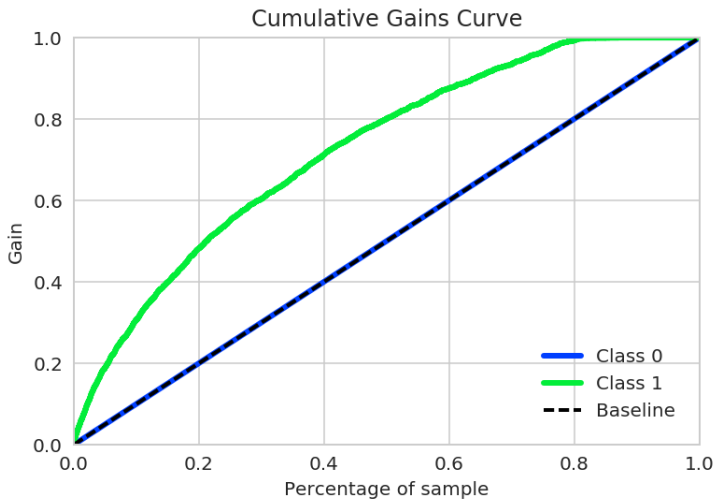

Figure 16: Cumulative Gain Chart for the complete XGBoost model

As the pattern of the features changes over time, so does the predictive power of the trained model. Hence a scheduled retraining of the model is preferable. The reason that retraining is not done at every scoring date is that it has high computational cost. The model should be retrained in a new training period once the model performance metrics drops below a specified level of performance.

4.4 Feature Importance

As one can observe in the figures in section 3.2.3 and section 3.3.3 some features are more important than others. The feature importance is based on the Gini criterion and entropy that evaluate the performance of a feature when the models makes splits. The important features are the ones that in the best possible way separate between Opt-Out and no Opt-Out. The information gain for using the most important feature at the top of the tree is much larger for the most important features. In the case of the complete random forest model Channel ID is the most significant variable. This is due to the fact that there are many more Opt-Outs observed for the email channel. Why more people Opt-Out from that channel could be due to the fact that customers receive more direct marketing through that channel and are more keen on keeping the channel clean from what specific customers consider as junk or non-relevant marketing. Channel ID does not have the same importance for the complete XGBoost model as it estimates higher importance of the customer dimension features. The large difference between feature importance between the models was not expected. Though since the random forest randomly select a subset of features for each tree and each time the model is trained a small difference could have been expected though not as large. Some of the important features that were expected to come up as important also showed large importance for

both models, such as the customer age. This was expected due to the fact that elderly people can in general be considered as less experienced in the usage of technical devices. The average email opening rate came up as a top important variable and this was not as expected as the total number of interactions, which also came up as important.

Due to confidentiality the univariate effect on the response is not presented in the report. As the ensemble methods can be viewed as black box models it is hard to present an intuitive interpretation of features. This can be solved with surrogate models such as logistic regression where the linear effect can be observed (Molnar, 2019). However, when tested the data did not give any indication of being linearly separable and therefore not applicable for the project. Another way to interpret the features is to use the so called Shapely values. The computational complexity of using the Shapely values is high and the method was not considered useful within the scope of this project. The Shapely method suggests a data set of maximum 1000 observations and 10 features (Lundberg and Lee, 2017), whereas our final data set consists of approximately 4 000 000 observations and 82 features.

5 Conclusion

The Company has many customers subscribed to different services and a large portion of their customers is subject to direct marketing actions. These actions aims to be as personalised as possible, though when customers no longer wish to receive any marketing actions they Opt-Out. At The Company there has been no in-depth analysis regarding the Opt-Out phenomena and initial insights have given a rough picture of what is the likely drivers for Opt-Out and some specific content were more interesting than others. With this initial knowledge the scope of the thesis project was to investigate underlying factors of Opt-Out and build a model that can be scheduled in production and able to score customers in a way that a probability of Opt-Out is returned.

The final data matrix was created in such a way that all training observations contained 360 days of historical data. It was also reduced to the degree that it represented only one communication message. It was created in this way since the actual Opt-Out is only connected to a specific dialogue with no respect to the actual offers contained within. Many different models were trained, tuned and evaluated on the processed communication data, personal data and dialogue data.

The final model that was decided upon and that will be used by The Company is the XGBoost model. The model performed better and showed to be more robust in testing periods. With an AUC score for the complete model of 0.71 two months ahead and 0.70 for three months ahead. And, for the reduced model AUC scores of 0.65 two months ahead and 0.64 for three months ahead. The most significant variables are related to customer specific information such as age and how long they have been a customer. Followed by communication related data such as time elapsed since they last received a communication or how long on average they are waiting before opening an email.

When considering different percentage of the total amount of customers the final model is able to perform at levels which on average among the testing periods, February and March, is 2.0 times random for the complete model and approximately 1.6 times random for the reduced model. The model should be retrained when the performance falls below a desired level. This has not been analysed during the project but it is recommended to look at once the models is set in production. It can be observed that the generalising ability is declining between in-sample training data and the validation period as well as the two testing periods. Hence, an initial recommendation is that the model should be scheduled for retraining every third month to maintain a good performance.

5.1 Future recommendations

This thesis project has served as an initial research where the aim was to investigate and build a predictive model that is able to estimate probabilities of Opt-Out. To be able to do this within the time frame that was set, some limitations were set beforehand. These limitations could to some extent be loosened, so that future versions of the model include a larger set of customer specific variables. These should then be investigated to see if they have any significant impact on the model performance. Other features that should be considered as an addition to the model are historical retention data. Features that reflect actions taken to retain customers, such as discounted months on their subscription or similar. Furthermore other time windows for when the historical features are calculated should be tested, which would extend the time frame that was set as a limitation, thought insights gained from the investigation could result in a shorter time window used. Before these investigations are undertaken it is recommended to first implement the more customer specific features since some shorter time windows have already been considered with worse performance.

To further improve the value that the model is able to create it one should include data that considers average gain per customer not Opting-Out. Meaning that if a customer stays in the communication channels how much will that customer on average generate in additional sales or other valuable metrics. If possible even divide customers into segments where they have different relative gain if kept in the communication channels. If one segment is considered less likely to generate money then different marketing actions might be necessary.

6 References

Telia Company AB. *Om företaget*, 2017. URL https://www.teliacompany.com/sv/om-foretaget/. accessed 2019-02-11.

Omar Adwan, Hossam Faris, Osama Harfoushi, Nazeeh Ghatasheh, and Khalid Jaradat. *Predicting Customer Churn in Telecom Industry using Multilayer Preceptron Neural Networks: Modeling and Analysis*. 2014.

Michel Ballings and Dirk Van den Poel. *Customer event history for churn prediction: how long is long enough?* 2012.

Tianqi Chen and Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System*. 2016.

Dataskyddsförordningen. *Dataskyddsförordningens syfte och tillämpningsområde*, 2018. URL [https://www.datainspektionen.se/lagar--regler/dataskyddsforordningen/dataskyddsforordningens-syfte-och-tillampningsomrade/]. accessed 2019-02-11.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *Additive Logistic Regression: A Statistical View of Boosting*. 2000.

Mohammed Hassouna, Ali Tarhini, Tariq Elyas, and Mohammad Saeed AbouTrab. *Customer Churn in Mobile Markets: A Comparison of Techniques*. 2015.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, New York, 2009.

Haibo He and Edwardo A. Garcia. *Learning from Imbalanced Data*. 2009.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *Introduction to Statistical Learning*. Springer, New York, 2013.

Gustavo A. Lujan-Moreno, Phillip R. Howard, Omar G. Rojas, and Dogulas C. Montgomery. *Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study*. 2018.

Scott M. Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017.

Christoph Molnar. *Interpretable Machine Learning*. 2019. URL [https://christophm.github.io/interpretable-ml-book/]. accessed 2019-05-04.

Simon Wood. *Generalised additive models. An introduction with R*. 2017.

A Complementary Theory

A.1 Model Metrics

A.1.1 Confusion Matrix

The theory related to the confusion matrix is based on *An Introduction to Statistical Learning* (James et al., 2013, p.148-149). A confusion matrix which can be applied for this thesis project is displayed in Table 13.

Table 13: Possible results when applying a classifier or diagnostic test to a observation .

		Predicted Class		
		- or 0	+ or 1	Total
True Class	- or 0	True Neg. (TN)	False Pos. (FP)	N
	+ or 1	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	N_t

When applying the confusion matrix on the result of the thesis project the `+ or 1` represents a customer Opt-Out while `- or 0` represents no Opt-Out. True negatives (TN) and positives (TP) represents the count of the observations that are correctly classified in the each class. False negatives (FN) and positives (FP) on the another hand represents the observations that are incorrectly classified. Table 14 lists popular performance measurements that are related to the structure of the confusion matrix.

Table 14: Popular measurements for classification and diagnostic testing

Name	Definition	Synonyms
True Neg. rate	TP/N	1−Type I error, specificity, selectivity
False Pos. rate	FP/N	Type I error, 1−specificity, fall-out
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
False Neg. rate	FN/P	Type II error, miss-rate, 1−sensitivity
Pos. Pred. value	TP/P*	Precision

A.1.2 Lift Score

Based on the metrics defined in Table 14 a Lift Score can be calculated, see equation 2. The score tries to describe the gain of a model by comparing the model predictions with randomly generated predictions based on the ratio between classes. A lift of 1 represents the performance of a random model and the range of the Lift Score is $[0, \infty)$. For example, if a random model predicts 10 % correctly between classes, then a lift of 3 means that the classification

model will predict 30 % correctly.

$$Lift\ Score = \frac{\frac{TP}{TP+FP}}{\frac{TP+FN}{TP+TN+FP+FN}} = \frac{\frac{TP}{P^*}}{\frac{P}{N_t}} \tag{2}$$

A.1.3 ROC-AUC

ROC (Receiver Operating Characteristics) - AUC (Area Under The Curve) is a performance measurement for classification methods which measures the separability of classes. The ROC curve is a popular graphic for simultaneously displaying the two types of errors for all possible thresholds. The axes of the ROC curve is represented by the true positive rate (TPR) and false positive rate (FPR) defined in Table 14. The overall performance of a classifier, summarised over all possible thresholds, is given by the AUC. The higher the AUC, the better the model predicts classes of 0s and 1s correctly, see Table 13. AUC lies in the range $[0.5, 1]$ where 0.5 represents a classifier that does not perform better than chance, while AUC=1 represents perfect separation between classes. An example of a ROC-curve is shown in Figure 17, where the shaded area below the upper ROC-curve represents the AUC-score of the model. The dashed line represents the ROC-curve for a classifier that does not perform better than chance, corresponding in $AUC = 0.5$ (James et al., 2013, p.147-148).

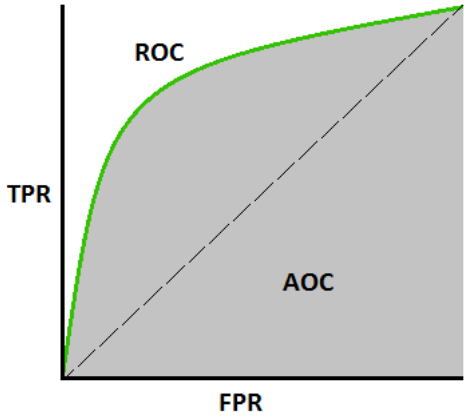

Figure 17: Example of a ROC-curve

A.1.4 Accuracy

From the confusion matrix it is also possible to calculate the accuracy of a model. The accuracy (ACC) is a popular performance measurement that calculates the overall classification performance of the model, it is defined in equation 3 (James et al., 2013, p. 398).

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{N_t} \tag{3}$$

A.2 Overfitting

When creating machine learning models, the data are often divided into a training and a testing data set. The training data set is used to teach the model patterns based on the features and observations. The trained model is then validated by the test data. During the training phase it is possible that the model gets too complex and loses flexibility to fit new data, this is known as overfitting. In the case of overfitting the prediction error for the training data is small compared to the prediction error of the test data. In this case the model fitted will not yield good response estimates based on new observations (James et al., 2013, p.22).

A.3 K-Fold Cross Validation

One of the most common ways to estimate model performance on in-sample predictions is Cross Validation. The K-Fold Cross Validation uses the entire training sample but split the sample in K roughly equal parts. Each fold K will be used to validate the model trained on the remaining $K-1$ folds (Hastie et al., 2009, p. 241-249). In Figure 18 one can observe an example of a five fold cross validation. The fold could be done at random or not at random, with a stratified or not stratified method. Using a stratified method preserves the percentage of samples of the classes and a random fold will select observations at random from the entire set. The effect will generate a shuffled data set which is used as training set and another set used for validation on trained model.

Figure 18: Five Fold Cross Validation example.

A.4 Random Forest

The theory related to random forest is based on *Elements of Statistical Learning* (Hastie et al., 2009, p.309-314). Since random forest is built on the theoretical foundation of decision tree and bagging, the theories behind these algorithms are presented first.

A.4.1 Decision Tree

Decision trees can be used both for classification and regression problems. The main difference between the two is the applications related to the type of desired response. A classification tree generates a qualitative response that tries to predict where each observation belongs by assigning it to the most commonly occurring class of training observations in the region to which it belongs, while a regression tree gives a quantitative response based on the mean response of the training data. This project mainly focuses on classifications trees with its estimated probabilities.

The classification trees are grown by the use of recursive binary splitting for the data sets different features. When making the binary splits, classification trees use the classification error rate, which is defined as the fraction of the training observations in the node that do not belong to the most common class. The classification error rate (E) is defined in equation (4), where $\hat{p}_{mk}$ represents the proportion of training observations in the m:th region that are from the k:th class.

$$E = 1 - \underset{k}{max}(\hat{p}_{mk}) \tag{4}$$

As classification error is not sufficiently sensitive when growing trees, other measures are introduced. The Gini index is referred to as a measure of node purity. If the Gene index (G) is small, it means that a node contains predominantly observations from a single class. The Gini index is defined in equation (5) and measures the total variance for K classes.

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{5}$$

The Gini-index is often used to evaluate the quality of the split as it indicates when a class is underrepresented in nodes. An alternative to the Gini index is the Cross entropy which also aims to find the purity of the nodes. The Cross entropy (D) is defined in equation (6).

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk})) \tag{6}$$

A.4.2 Bagging

A problem with decision trees is that they suffer from high variance. This means that if we split the training data into two parts at random and fit a decision tree to both halves, the results that we get could be quite different. To reduce the variance of a statistical learning method, a Bootstrap aggregation method known as bagging is introduced.

Given a set of n independent observations $Z_1, \ldots, Z_n$ each with a variance σ^2, the variance of the mean $\bar{Z}$ will then be $\frac{\sigma^2}{n}$. Hence taking the mean of the set reduces the variance. We define $\hat{f}(x)$ as a decision tree built on the predictors in x. With this in mind we can calculate $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^B(x)$ using B separate training sets and use the average of them to minimise the variance. In general the access of multiple training sets are not available, so bagging uses Bootstrapping instead. By generating B different bootstrapped training data sets and then training the data sets to get $\hat{f}^{*b}(x)$. Lastly, taking the average of all predictions we can define bagging as in equation (7).

$$\hat{f}_{bag}(x) = \sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{7}$$

Bagging is popular to use together with decision trees as it handles the issue with high variance. By constructing B decision trees based on the B bootstrapped training sets it possible to make an overall prediction, the mean predicted value for an observation represents the predicted probability. For classification the model classifies the observation based on a chosen threshold. Often a threshold of 0.5 is used, meaning that the model classify an observation into the most commonly predicted class among the B predictors, this is known as majority voting.

It is possible to show that on average each bagged tree makes use of around two-thirds of the observations. The remaining observation that are not used to fit a given bagged tree are known as the out-of-bag (OOB) observations. We can predict the response for the i:th observation by looking at each of the trees where the observation was in the OOB. This will yield around B/3 predictions for the i:th observation. We can average the predictions by looking at the majority vote which leads to a single OOB prediction for the i:th observation. An OOB prediction can be obtained in this way for each of the n observations, from which the classification error can be computed. The resulting OOB error is a valid estimate of the test error for the bagged model, since the response for each observation is predicted using only the trees that were not fitted using that observation.

A.4.3 Random Forest Algorithm

As mentioned random forest is based on decision trees and bagging. The difference between random forest and bagged trees is that random forest decorrelates the trees. When the random forest algorithm grow decision trees, it selects a random sample of m predictors from the full set of p predictors when splitting the trees. Consequently, the split is only allowed to use m predictors. In general $m = \sqrt{p}$ and therefore the algorithm is not allowed to use the majority of predictors. By forcing the algorithm to only choose a subset of the predictors the random forest is able to handle highly correlated bagged trees.

A.5 Extreme Gradient Boosting

Extreme Gradient Boosting, more commonly known as XGBoosting is a scaleable end-to-end ensemble method. XGBoosting is built on the foundation of combining Gradient Boosting with Regularisation (Chen and Guestrin, 2016).

A.5.1 Regularisation of Ensemble Model

For a given data set D with n observations and m features, where $D = \{(x_i, y_i)\}$ a tree ensemble model uses K additive functions to predict the output y_i where F is the space of regression trees. ϕ represents the ensemble method that uses x_i to predict y_i, see equation (8).

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F. \tag{8}$$

Each f_k corresponds to an independent tree structure q with T number of leaves and leaf weights w. Unlike the decision trees defined in Section A.4.1 the regression trees in F contains a continuous score in each weight w and by summing these up for all T leaves we obtain the final prediction for each tree q. To train the functions in set F we minimise the regularised loss function for the response y_i in equation (9).

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \tag{9}$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda w^2.$$

$L(\phi)$ is the convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. The second term Ω penalises the complexity of the model with regularisation parameters γ and λ to avoid overfitting, see Appendix A.2.

A.5.2 Gradient Boosting

As the tree ensemble methods loss function in equation (9) cannot be optimised in the Euclidean space the model has to be trained in an additive matter. This means that for each tree/iteration k we compute a prediction $y_i^{(k)}$ for the i-th observation and choose the function that is optimal for the loss function at that iteration. This is done by including the function output f_k in the loss function in equation (9),

$$L^{(k)} = \sum_{i=1}^{n} (l(\hat{y}_i, y_i) + f_k(x_i)) + \Omega(f_k).$$

The loss function can then be optimised by approximating the second order Taylor expansion, more theory about this can be found in *Additive logistic*

regression: a statistical view of boosting (Friedman et al., 2000). The loss functions will then look as follows,

$$L^{(k)} \simeq \sum_{i=1}^{n} [l(\hat{y_i}^{(k-1)}, y_i) + g_i f_k(x_i)) + \frac{1}{2} h_i f_k^2(x_i)] + \Omega(f_k), \tag{10}$$

where $g_i = \partial_{y^{(k-1)}} l(\hat{y_i}^{(k-1)}, y_i)$ and $h_i = \partial^2_{y^{(k-1)}} l(\hat{y_i}^{(k-1)}, y_i)$ are the first and second order gradient statistics of the loss function. We defined $I = \{i|q(x_i) = j\}$ as the set of observations for leaf j. Then we can rewrite the loss function in equation (10) by removing the constant terms and expand Ω to get the loss function for each iteration k, see equation (11).

$$\tilde{L}^{(k)} = \sum_{i=1}^{n} [g_i f_k(x_i)) + \frac{1}{2} h_i f_k^2(x_i)] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w^2 \tag{11}$$

The optimal weight for a given tree structure $q(x)$ can be computed by

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda},$$

with the corresponding optimal value

$$\tilde{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{12}$$

Equation (12) can be used as a scoring function to evaluate the performance of each tree structure q. Normally it is impossible to enumerate all the possible tree structures q, therefore XGBoost uses a greedy algorithm instead. The algorithm starts from a single leaf and iteratively adds branches to the tree and evaluates the qualities of each split. Assume I_L and I_R that are the sets of observation after a split is performed. Letting $I = I_L \cup I_R$, we can defined the loss function for split evaluation as

$$L_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} + \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma. \tag{13}$$

The algorithm repeatedly chooses the splits for each node that minimises the loss function in equation (13).

A.5.3 Properties of Extreme Gradient Boosting

XGBoost uses two additional techniques beside regularisation to improve the performance of the model. The first technique is shrinkage of weights, which is done by scaling newly added weights with parameter η, aslo known as the learning rate. This reduces the influence of an individual tree and gives room

for future trees to improve the model. Feature sub-sampling is the other technique used to improve the model. It works as bagging does in the random forest algorithm by selecting sub-samples of features for each tree. This is done to decorrelate features, reduce bias and prevent overfitting of the ensemble model. Furthermore, the XGBoost algorithm has many computational advantages compared to other ensemble models. Such advantages are block structure for parallel learning, cache-aware settings and out-of-core computations. More information about the computational advantages can be found in *XGBoost: A Scalable Tree Boosting System* (Chen and Guestrin, 2016).

A.6 Imbalanced Learning

The learning and prediction of a machine learning algorithm can face problems when the response is heavily weighted towards one or more response categories. This is known as between-class imbalance and can cause problems with predictions as imbalanced models tend to have high accuracy for the majority class and low accuracy for the minority class. To deal with this problem methods for over/under-sampling are introduced. These methods re-sample the data by either generating, replacing and/or removing observations in desired categories. These methods can be applied to both binary and multi-class models. All the theory related to imbalanced learning is based on *Learning from Imbalanced Data* (He and Garcia, 2009).

Consider a given training data set S with m observations where $S = \{(x_i, y_i)\}$, $i = 1, \ldots, m$, and where $x_i \in X$ is an instance in the n-dimensional feature space $X = \{f_1, \ldots, f_n\}$. Furthermore, $y_i \in Y = \{1, \ldots, C\}$ represents the class labels related to the feature space X. For $C = 2$ which represents a two-class classification problem, we define two subsets $S_{min} \subset S$ and $S_{maj} \subset S$, where S_{min} represents the minority class and S_{maj} the majority class. Sets generated or removed from sampling are defined as E and the re-sampled data set are defined as $S_{sampled}$.

$$S_{sampled} = S_{min} + S_{maj} + E \tag{14}$$

A.6.1 Over-sampling

Over-sampling methods aim to generate new observation with or without replacement in order to balance the data set. There are various methods to perform over-sampling such as SMOTE and Adasyn (He and Garcia, 2009), however this project used the RandomOverSampling algorithm. Since over-sampling generates new observations there is a risk that models built on the re-sampled data are exposed with overfitting, see Appendix A.2.

RandomOverSampling adds a set E to the data set S, based on sampling of the minority set S_{min}. The larger class subset S_{maj} are kept intact to increase the size of S, as shown in equation (14).

A.6.2 Under-sampling

Under-sampling methods aim to create balance by removing observations from desired categories with or without replacement. The risk with under-sampling is quite obvious since data is removed/replaced, information might be lost. There are various types of algorithms for under-sampling such as KNN under-sampling and Near Miss 1-3, which both are based on distance measurements between observations (He and Garcia, 2009). However, in this thesis project only the RandomUnderSampler algorithm is applied.

RandomUnderSampling removes a set E from the majority class S_{maj} and decreases the size of the data set S by not generating new samples. In equation (14) the sampling term E will be negative.