Predicting Churn using Hybrid Supervised-Unsupervised Models

Meeke Rijnen (S1856723)

First Supervisor: Dr. W. Kowalczyk

Second Supervisor: Dr. E. Dusseldorp

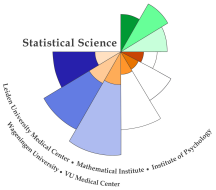
External Supervisors: B. Hazen and C. Vletter (NEWCRAFT)

MASTER THESIS

Defended on June 29th 2018

Specialization: Data Science

Universiteit Leiden



STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL SCIENCES

Abstract

Telecom providers suffer from a loss of valuable customers to competitors. This is known as churn. The first step to retain customers is to predict which customers are most likely to churn. Next, predicted churners can be targeted to encourage them to stay. It is therefore crucial to build a churn prediction model that is as accurate as possible. Such models are usually built by applying a supervised learning algorithm to historical data. In this study, a more sophisticated approach is investigated, where historical data is first clustered using unsupervised learning and then for each homogeneous group a model is built with the help of supervised learning.

Customer data, contractual data and online behavior data from a Dutch telecom provider are collected. Homogeneous groups of customers are identified based on the customer and contractual data using t-Distributed Stochastic Neighbor embedding (t-SNE), Gaussian Mixture Model (GMM) and Latent Class Analysis (LCA). Additionally, a partitioning of data that is suggested by domain experts (i.e. segmentation) is considered. The supervised learning models used are Logistic Regression (LR), Random Forest (RF), XGBoost and a heterogeneous ensemble of the aforementioned models. The performance of the various combinations are measured with the help of the Area Under the Curve (AUC). All combinations of techniques are compared to the benchmark approach that does not utilize any results from an unsupervised learning technique.

The results revealed that for the flexible models (i.e. RF, XGBoost and the ensemble) there is no added value of using a hybrid approach as the highest AUC is for the benchmark approach. However, for the less flexible models (i.e. LR), the largest AUC is for the hybrid approach. This suggests that a LR fitted for each homogeneous group is able to model the complex relations in the data set better than a LR for the whole data set.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Wojtek Kowalczyk for his support, knowledge and critical questions. Elise Dusseldorp has also been of great help. Thank you for your statistical insights. Moreover, it has been challenging, but I would like to thank you both for helping me to integrate the Computer Science, Statistics and Business objectives in one Master Thesis.

Thank you Newcraft for the ability to work on such an interesting topic for my Master Thesis. Special thanks goes to Bart Hazen who always was available for answering my questions. Thank you for retrieving a new data set, over and over. Cornelis Vletter, Thomas Hemels and Job Tiel, thank you for your guidance along this process.

I would like to thank Marieke Vinkenoog. You have the ability to explain difficult concepts in such an easy manner. I am grateful that I got the opportunity to work with you. Moreover, Ina Quataert thank you for our walks from the Snellius building to the train station. It was a great way of processing the information received during the day.

I would like to use the space that is left to thank, for me, the most important people. Thank you Bram Durenkamp for always motivating me. Without you I would never have finished this Master. Thank you for your believe in me and your down-to-earth mentality. Moreover, I would like to thank Ad Rijnen, Mia Rijnen and Noortje Rijnen, thank you for always supporting me.

Chapter 1

Introduction

A company's customer base is their most valuable asset as customer acquisition comes at a greater cost than customer retention [1]. Hence companies have shifted their focus from acquiring new customers to making sure that existing ones stay. However, companies suffer from a loss of valuable customers to competitors. This is known as churn [2]. To effectively retain existing customers, an appropriate churn management strategy is vital for any business.

Among all industries which suffer from churn, the telecommunications industry can be considered at the top of the list. The telecommunications sector makes communication possible on a global scale, through (mobile) phone or the Internet. The spectacular growth of the purchasing of the mobile phone from 2000 onwards has been driven innovation and encouraged new telecom providers to enter the market. As customers have multiple providers to choose from and switching costs are low customer loyalty is minimal. Moreover, the rise of online shopping makes product comparison transparent. This is especially a threat in a market where products are homogeneous and where prices and services are interchangeable. It is estimated that the average churn rate for telecom providers is 2.2% per month [3]. Moreover, the costs of acquiring new clients has increased from $300 to $600 [4]. As a result, the focus of telecommunication companies has shifted from building a large customer base to retaining customers.

To retain customers, an appropriate churn management strategy is vital. The first step to manage churn is to predict which customers are most likely to churn. The availability of data makes churn prediction possible. Examples of data sources are call centers, online browsing behavior and Customer Relationship Management systems. Next, predicted churners can be targeted to encourage them to stay. This setup enables the firm to focus on customers who are at risk to churn. In this study, the focus is on the first step within the churn management strategy: predict which customers are most likely to churn.

To effectively predict which customers are most likely to churn, data mining techniques can be applied. Data mining techniques come in two main forms: supervised and unsupervised learning. In the unsupervised learning problem, only the features X are observed and there is no measurement of the outcome y. The task is to describe how the data are organized

or clustered. In the supervised learning problem, the features X and the outcome y for a set of objects are observed. The goal is to predict the outcome y for new unseen objects as accurately as possible [5].

Hybrid data mining models, where unsupervised and supervised learning techniques are combined, are proven to improve the predictive performance [6]. In this work, churn prediction will therefore be performed in two learning stages. In the first stage, homogeneous groups of customers will be identified using unsupervised learning. In the second stage, using supervised learning, a classifier will be developed for each cluster to predict which customers are most likely to churn. The expectation is that the hybrid approach will improve the predictive performance by combining unsupervised and supervised learning.

1-1 Problem statement

The highly competitive and dynamic telecommunication market results in customer churn. It is unclear what the indicators of churn behavior are and which customers are at high risk of churn. The general research goal to achieve in this thesis is:

Main goal: Evaluate the added value of combining unsupervised and supervised learning with respect to the accuracy of predicting churn.

To achieve the main goal, it is decomposed into three specific subgoals:

Subgoal 1: Identify groups of customers who share the same characteristics using different unsupervised learning techniques.

A distinction is made between finding homogeneous groups of customers using clustering and segmentation. Three clustering methods are used: t-Distributed Stochastic Neighbor embedding (t-SNE) [7], Gaussian Mixture Model (GMM) [8] and Latent Class Analysis (LCA) [9]. Additionally, hand-crafted data segmentation is considered, where domain experts provide definitions of data segments.

Subgoal 2: Within each homogeneous group, predict which customers are most likely to churn using different supervised learning techniques.

To predict the probability of churn the following methods are used: Logistic Regression (LR) [5], Random Forest (RF) [5], XGBoost [10] and a heterogeneous ensemble containing the previously mentioned methods.

Subgoal 3: Evaluate different combinations of unsupervised and supervised learning techniques with respect to the accuracy of predicting churn.

Several combinations of unsupervised and supervised learning techniques are examined which are listed in Table 1-1. This study predicts which customers are likely to churn for the mobile customers that are allowed to renew their contract (i.e. renewable mobile customers).

		Unsupervised Learning			
		Data set		Hybrid approach	
		All mobile customers	**Renewable mobile customers**	**For each homogeneous group**	**Homogeneous group label as feature**
Supervised Learning	**LR**	GMM	t-SNE-GMM, GMM, LCA, Segmentation	t-SNE-GMM, GMM, LCA, Segmentation	LCA
	RF	GMM	t-SNE-GMM, GMM, LCA, Segmentation	t-SNE-GMM, GMM, LCA, Segmentation	LCA
	XGBoost	GMM	t-SNE-GMM, GMM, LCA, Segmentation	t-SNE-GMM, GMM, LCA, Segmentation	LCA
	Ensemble	GMM	t-SNE-GMM, GMM, LCA, Segmentation	t-SNE-GMM, GMM, LCA, Segmentation	LCA

Table 1-1: Overview of all combinations examined in this study.

Perhaps, renewable mobile customers are a subgroup of the full mobile customer data set of the telecom provider. To investigate if this is the case, the unsupervised learning task is applied on the full mobile customer base as well as on the renewable mobile customer data set.

Secondly, two methods of hybridization for utilizing the results of the unsupervised learning techniques are examined. The first method uses the labels that represent the identity of the homogeneous groups as input for the prediction of churn. The second method separates customers into homogeneous groups and builds a predictive model for each group.

To identify which approach most accurately predicts the probability of churn, all combinations of techniques are evaluated using the Area Under the Curve (AUC).

1-2 Research outline

This thesis is organized as follows: the first chapter is used as an introduction into the research goals of this study. Chapter 2 contains a literature overview of relevant studies on churn prediction. Chapter 3 presents the analysis strategy used in this study. Chapter 4 specifies the analysis methods in detail. Chapter 5 provides insights in the data collection, selection, and preprocessing steps required in order to prepare the data for the analysis. Subsequently, exploratory analysis is performed. Chapter 6 presents the results. Lastly, a discussion of the results, concluding remarks and directions for future work are provided in chapter 7.

Chapter 2

Theoretical foundation and background

In this chapter relevant literature is reviewed to set the theoretical foundation of the current study. In section 2-1 common approaches for churn prediction are described. In section 2-2 the contribution of the current study to the literature is described.

2-1 Common approaches to predict churn

Data mining

A common approach to predict churn is to use data mining models [11]. Data mining refers to the process of extracting useful information from data [12]. A standard methodology for data mining is Cross-Industry Standard Process For Data Mining (CRISP-DM) [13]. The CRISP-DM methodology breaks the data mining process into six major phases as illustrated in figure 2-1. The sequence of the phases is not strict and the process is iterative as a data mining solution needs continually optimization.

The first stage from CRISP-DM specifies the objectives and correlates them with data mining applications. In the second stage, the data are gathered, checked and the first insights are collected. In the third stage, the raw data are transformed into a suitable format such that data mining algorithms can be applied. It includes data cleaning, transformation and reduction. In the fourth phase, suitable data mining algorithms are selected and applied. The goal of the fifth stage is to thoroughly evaluate the model to be certain that the model achieves the business objectives set in stage 1. The final step is the deployment stage, where the knowledge gained is organized and presented in a way that the customer can use it. In this study, the focus is mainly on the first 5 phases of the CRISP-DM methodology.

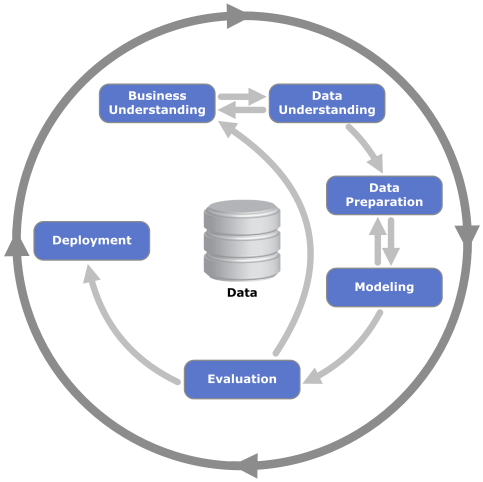


Figure 2-1: Process diagram showing the relationship between the different phases of CRISP-DM.

Classification

Classification is the data mining task that assigns objects/items to a target class [14]. The churn prediction problem in this study is therefore a classification task where customers are assigned to predefined classes [15].

Individual classifiers

Within the individual classifier models, a decision tree is the most popular model to predict churn [16] [17]. Other commonly used models are Logistic Regression [18] [19], SVM [20] [21] [22] and Neural Networks [23]. There are multiple studies that compare the performance of various individual classification techniques [24] [25] [26]. However, there is no agreement on which classification method performs best for customer churn prediction. Whilst some methods perform better than others, there is significant variability across the problems and metrics. Even the best models sometimes perform poorly, and models with poor average performance occasionally perform exceptionally well depending on the task it handles [27].

Ensemble classifiers

The second type of classification techniques is an ensemble learner [28]. A growing body of literature has examined the use of ensemble learners for classification [29]. In an ensemble classifier, several classification models are combined into one aggregated classifier. The predictions are combined into one aggregated outcome using a fusion rule [30]. Several studies within the churn literature have demonstrated that ensembles of classifiers demonstrate superior performance over single classification models [31] [28].

There are two techniques that can be applied within ensemble classification: homogeneous and heterogeneous assembling [32]. Homogeneous ensembles build individual classifiers on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. Example methods of homogeneous ensembles are boosting and Random Forest (RF).

Boosting algorithms iteratively train weak classifiers with respect to a distribution and com-

bines them to a final strong classifier. Boosting have gained increased popularity and attention due to its simplicity and high predictive performance [10]. Boosted algorithms have also been proven to increase the predictive power within the churn prediction problem [33].

RF is an ensemble model proven to be successful for the prediction of churn [34]. Each tree in the ensemble is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. When splitting a node during the construction of the tree, the split that is picked is the best split among a random subset of the features. As a result, the bias slightly increases while due to averaging the variance decreases.

Heterogeneous ensembles combines the predictions of different individual classifiers by averaging, taking the majority vote or stacking. To make sure an heterogeneous ensemble is successful it is important to (1) teach each individual classifier accurately and (2) ensure diversity among the base models [35].

Extending the data set

A different approach to increase the predictive performance of a model is by adding observations or features. Braun and Schweidel [36] suggest that to increase the predictive performance of a churn model, extra explanatory variables should be incorporated. They specifically suggest touch points with the customer, such as online behavior, will have considerable practical value. Online behavior characteristics are common features in predicting purchasing behavior [37] [38]. However, there is no study found that use online behavior for the prediction of churn.

Unsupervised learning

One of the most important steps for better performance of a classifier is to pre-proccess the data correctly [39]. An advanced pre-processing step is to first identify homogeneous groups of customers [6]. Within a classification task, the observations are first assigned to homogeneous groups and a classifier is built for each group. Applying this hybrid framework to a cluster prediction problem has shown an increase in predictive performance [40] [17] [41]. There are two main approaches available to find homogeneous groups of customers: clustering and segmentation.

Clustering is an unsupervised learning technique that explores data sets in order to discover the natural structure and unknown but valuable behavioral patterns of customers' hidden in it [42]. There are various approaches available that can be used for clustering. These approaches can be split into hard and fuzzy cluster assignment. Hard clustering techniques assign a class label while fuzzy clustering assigns for each observation a probability for each of the clusters [43]. Within hybrid churn prediction models both clustering techniques are applied [41] [17].

The precursor of clustering is market segmentation. Based on general, domain and brand-specific variables, segmentation divides observations in homogeneous subgroups [44]. Market segmentation has been proven to be a solid foundation for determining marketing strategies [45].

2-2 Contribution to the literature

This study contributes to the churn literature in several ways. First, it divides customers into homogeneous groups using both clustering algorithm and expert-made data segmentation and evaluates both approaches with respect to the accuracy of predicting churn. Research on the comparison of clustering and market segmentation in a hybrid framework has not been found.

Secondly, both single classifiers, homogeneous ensemble classifiers and heterogeneous ensemble classifiers are applied to the prediction of churn. Although research on hybrid churn models investigated the predictive performance of at least one of the classifiers, no research on hybrid churn models is found that compared the techniques.

Lastly, whereas literature often uses the same data source for the unsupervised and supervised learning stage, this study evaluates the use of different sources of data for each stage. Customer characteristics are used for defining homogeneous groups of customers while online behavior is used for the prediction of churn.

To conclude, this study implements several combinations of hybrid techniques to predict churn and compares them to the benchmark model when no unsupervised learning is used. As a result, it provides insight into the classification ability of different combinations of hybrid approaches on a real life data set in order to evaluate the best performing approach for this particular case.

Chapter 3

Analysis strategy

In this study multiple modeling approaches to predict churn are proposed. This chapter outlines the general framework for each of the approaches. The framework is illustrated in Figure 3-1.

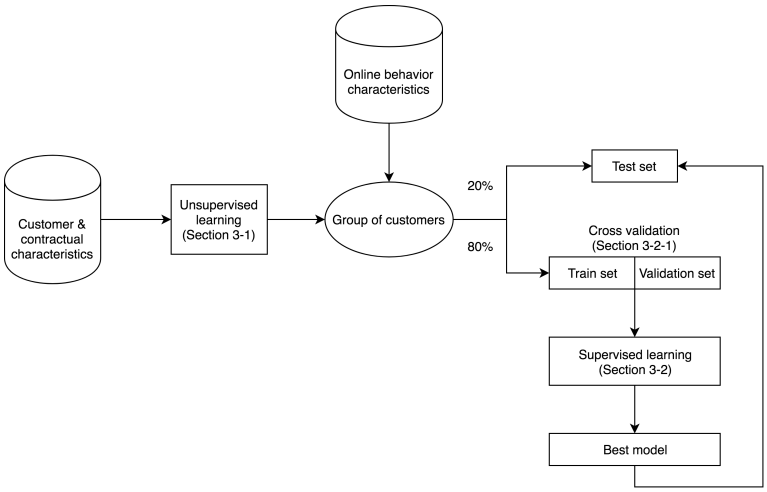


Figure 3-1: Framework of the analysis.

3-1 Unsupervised learning

The first subgoal is to obtain homogeneous groups of customers that share the same characteristics using clustering and segmentation.

One of the biggest challenge when applying clustering algorithms is to determine the number of clusters to be generated. Although, there are a couple of statistical techniques available (e.g. the elbow method or BIC criteria), in this study domain experts suggest the number of groups based on their expert knowledge of the field (i.e. 7 homogeneous groups).

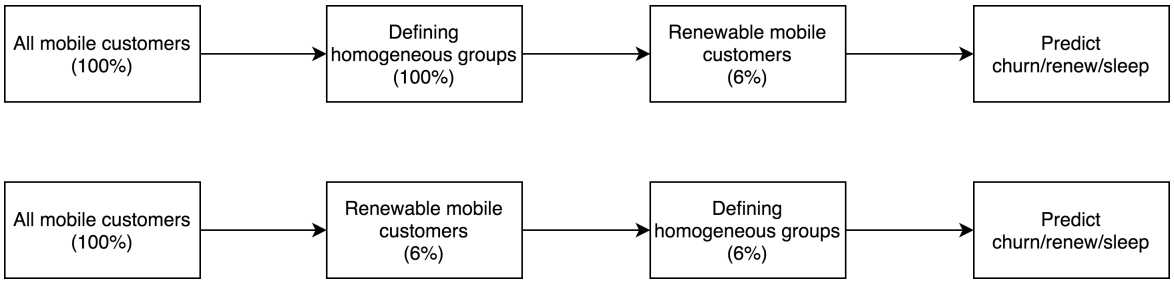


Figure 3-2: Clustering is performed on the full mobile customer data set and on the renewable mobile customer data set.

When a new customer is entering the database, it is important to specify how his cluster or segment label will be determined. In this study, fuzzy clustering is applied. Fuzzy clustering assigns for each observation a probability for each of the clusters based on a fixed set of parameters. Thus each data point can have membership to multiple clusters. After the constructing of the clustering model, the probability of any future observation belonging to each of the cluster components can be estimated using the model parameters obtained with Expectation-Maximization Algorithm (EM). More information on EM will be provided in Chapter 4. For labeling new customers with the segmentation approach, the defined rules by the marketeers can be used to determine the segment label.

The objective of this study is to predict which renewable mobile customers are going to terminate their contract. Perhaps, renewable mobile customers are a subgroup of the full (i.e. renewable and non-renewable) mobile customer data set. To identify if clustering algorithms learn the same structure from both the renewable and the full mobile customer data set, the unsupervised learning stage is performed on both data sets. This procedure is illustrated in Figure 3-2. The obtained clusters from both approaches are compared by measuring the similarity between the two different clusterings of the same set of customers with the Adjusted Rand Index [46]. The Adjusted Rand Index determines how each customer is assigned by each clustering method. Given a set S of n elements and two clusterings of these elements, namely $X = X_1, X_2, ..., X_r$ and $Y = Y_1, Y_2, ..., Y_s$ the overlap between X and Y can be summarized in a contingency table $[n_{ij}]$ where each entry n_{ij} denotes the number of objects in common between X_i and Y_j:

$X \backslash Y$	Y_1	Y_2	...	Y_s	Sums
X_1	n_{11}	n_{12}	...	n_{1s}	a_1
X_2	n_{21}	n_{22}	...	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Sums	b_1	b_2	...	b_s	

The Adjusted Rand Index can be calculated by

$$\text{Adjusted Rand Index} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{3-1}$$

where b_j and a_j correspond to the values from the contingency table.

The Adjusted Rand Index has a value close to 0.0 for random labeling and exactly 1.0 when the clusterings are identical.

3-2 Supervised learning

The online behavior data set carries a class label. Firstly, customers who retain their contract (renewers). Secondly, customers who do not take any action regarding their contract (sleepers). Lastly, customers who terminate their contract (churners). As a result, this study deals with a multi class classification task.

3-2-1 Cross-validation

To test if the supervised learning model is able to generalize well to new data, the data set is split into a train and a test set. The train set is a set of examples used for learning. The test set is an independent data set used to assess the performance of the learned classifier. In this study, the training set consists of a random hold-out sample of 80% of the total data set. The test set consists of the other 20%.

To make sure that each class is represented in the train and test set, stratified sampling is used. With stratified sampling a sampling fraction of each outcome class that is proportional to that of the total population is used. Thus, each set contains approximately the same percentage of samples of each target class as the complete set. If, in the complete set, one of the classes has more observations than the other, the proportion of the outcome classes is imbalanced. To deal with this problem an appropriate class imbalance strategy is needed as discussed in Section 3-3.

Splitting the data set into train and test sets makes it possible to test if an algorithm generalizes well to unseen data. However, it does not offer an explanation for why a model generalizes poorly. If predictions vary based on the specific data used to train them this is called overfitting. When overfitting occurs, the model would just repeat the labels of the samples in the training set which results in a near perfect score but would fail to predict anything useful on unseen data.

To limit a model from overfitting, the 80% training set is split in a train and validation set. Given a choice of hyper-parameter values, the train set is used to train the model. The validation set is used to evaluate the performance of the model for the different combinations of hyper-parameter values. Thus the validation is used to decide on the set of values for the hyper-parameters. The 20% test set is used to compare models in an unbiased way.

When partitioning the data into a train, validation and test set, the number of samples used for learning the model is reduced. Therefore, cross-validation is used. In $k-$fold cross-validation the training data is partition into k subsets, called folds. The algorithm is trained on $k-1$ folds while the remaining k folds are used as the validation set. This process iterated k times until every subset has been used as the validation set. In this study, 5 fold cross-validation is used.

3-2-2 Hyper-parameter optimization

To determine the hyper-parameter values that should be evaluated during the training and validation process randomized search cross validation is used [47]. Randomized search cross validation samples combinations of parameter values from a random distribution for a fixed number of iterations. In contrast to grid search cross validation, not all parameter values are tried out, but a fixed number of parameter settings are sampled from specified distributions. Randomized search is used as the results are quite similar to grid search, while the run time for randomized search is drastically lower [47].

Randomized search cross validation takes a scoring parameter that controls on which metric the hyper-parameters should be evaluated. In this study, the metric used is the macro F1 score. The macro-average method is especially useful when the interest is in the minority class as it gives equal weight to every class. For more information on how to calculate the F1 metric see Section 3-4. A macro average F1 score will compute the F1 metric independently for each class and then takes the average over all classes. Hence, macro F1 score treats all classes equally.

3-3 Class imbalance

Class imbalance occurs when one of the classes has more observations than the other. Algorithms will focus more on the classification of the major class while ignoring or misclassifying the minority samples [48]. Substantially better performance can therefore be obtained by using an appropriate class imbalance strategy [49]. Especially as the interest of this study is in the minority class (i.e. churn) an appropriate class imbalance strategy is needed. There are several strategies available such as incorporating the misclassification costs (see section 3-4-2) or undersampling the majority class or oversampling the minority class.

The disadvantage of undersampling is the loss of valuable information while the disadvantage of oversampling is that it reuses data and thus the classifier might learn a rule covering only a few, replicated, samples. Oversampling might therefore lead to overfitting. In addition, there are some more advanced approaches available (e.g. the SMOTE technique) which creates synthetic minority class samples [50]. However, these more sophisticated sampling techniques do not give any clear advantage [51]. Van den Poel et al., 2009 [52] argues that undersampling is favoured more than oversampling. They showed that undersampling can lead to an improved prediction accuracy in a churn prediction problem. Therefore, this study will use undersampling to make the classes balanced.

3-4 Performance

There are several performance measures available to evaluate classification algorithms. The choice of an evaluation criterion depends on the objectives of the research. The main objective in this study is to evaluate the added value of combining unsupervised and supervised learning with respect to the accuracy of predicting churn. To achieve this, the combinations of techniques need to be compared using a suitable performance metric.

		Predicted		
		Churn	*Renew*	*Sleep*
Actual	*Churn*	TP_{churn}	$FN_{churn,renew}$	$FP_{churn,sleep}$
	Renew	$FP_{churn,renew}$	$TN_{renew,renew}$	$FP_{renew,sleep}$
	Sleep	$FP_{churn,sleep}$	$Fn_{sleep,renew}$	$TN_{sleep,sleep}$

		Predicted	
		Churn	*Not churn*
Actual	*Churn*	TP_{churn}	$FN_{churn,renew}$ + $FP_{churn,sleep}$
	Not churn	$FP_{churn,renew}$ +$FP_{churn,sleep}$	$TN_{renew,renew}$ +$FP_{renew,sleep}$ +$Fn_{sleep,renew}$ +$TN_{sleep,sleep}$

Figure 3-3: Left table illustrates a confusion matrix in a multi class setting. Right table illustrates for the churn class how the multi class confusion matrix can be transformed to a 2x2 confusion matrix.

3-4-1 Performance measures

The starting point of performance measures is the confusion matrix. A confusion matrix consists of four metrics. The four metrics are the number of correctly classified observations (true positives, TP), the number of correctly classified observations that do not belong to the class (true negatives, TN) and observations that either were incorrectly assigned to the class (false positives, FP) or that were not recognized as class examples (false negatives, FN). An example of a confusion matrix in a multi class setting can be found in the left table of Figure 3-3. To calculate the final metrics for the class of interest, the metrics of the other classes can be summed as being illustrated for the churn class in the right table of Figure 3-3.

Based on the confusion matrix, several performance metrics can be calculated. First of all, the percentage of correct classifications (i.e. accuracy) can be calculated:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{3-2}$$

Accuracy is not preferred in this study due to the class imbalance. A model that classifies almost all observations as a non-churner will have a high accuracy while it is not accurate in the prediction of churn.

Secondly, the proportion of predicted churners that actual churns (i.e. precision) can be calculated:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3-3}$$

Moreover, the proportion of actual churners that are predicted to churn (i.e. True Positive Rate or recall) can be calculated:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{3-4}$$

In addition, the False Positive Rate can be calculated which measures the fraction of non churners that are misclassified as churners. It can be calculated as follows:

$$\text{False Positive Rate (FPR)} = \frac{FP}{FN + TN} \tag{3-5}$$

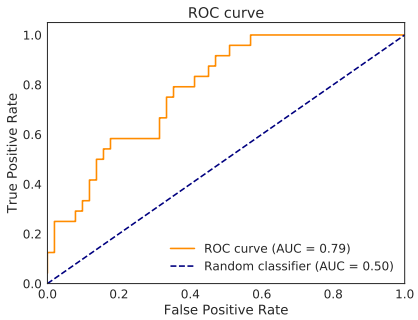


Figure 3-4: An example of a ROC curve. The area under the orange line is the AUC.

Lastly, the F1 score is the harmonic mean of precision and FPR and can be calculated as follows:

$$\text{F1 score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3-6}$$

To understand the implications of the above mentioned performance metrics it is important to introduce the threshold concept. The threshold is the decision rule which determines when an observation should be classified as a churner or not. All cases above a certain threshold are classified as churners; all cases having a lower churn probability are classified as non-churners. The above mentioned performance metrics are dependent on the chosen threshold as only one threshold is considered.

However, various thresholds result in different TPR and FPR. Deciding on where to set the threshold is therefore important as it requires weighing the cost of a FP versus a FN case. In this study, the cost of mis-predicting churn (i.e. FP) is higher than that of mis-predicting non-churn (i.e. FN). This is due to the fact that contacting a FP case might result that the customer changes its mind and decides to churn.

The training set in this study is balanced by using a class imbalance strategy as discussed in Section 3-3. However, the actual class distribution (i.e. the independent test set) is imbalanced. For classifying imbalanced data, using the default decision threshold will lead to high accuracy in predicting the majority class and low accuracy in predicting the minority class. A change of the decision threshold can therefore increase or decrease the predictive performance.

The Receiver Operating Curve (ROC) accounts for the overall performance of a classification technique by considering all possible thresholds. The ROC can thus be used to assess the performance of a classifier independently of any threshold. An example of a ROC can be found in Figure 3-4. The ROC is created by plotting the FPR against the TPR at various probability threshold settings. As this study is a multi class classification problem the ROC analysis is performed using pairwise comparison (i.e. one class. vs all other classes).

In order to be able to evaluate and compare the different combinations of unsupervised and supervised learning techniques, not the ROC but the summary statistic Area Under the Curve (AUC) is used. The AUC is equal to the probability that a classifier will rank a randomly chose positive instance higher than a randomly chosen negative one. Thus if a churn model indicated an AUC of 0.60, this means that if one randomly picks an actual

churner and a non-churner from the dataset, then 60% of the times the churner will have a higher churn probability output by the classifier than the non-churner. A random classifier has an AUC of 0.5 (illustrated with the blue line in Figure 3-4). A perfect classifier has an AUC of 1. The orange line in Figure 3-4 is therefore a better classifier than the blue line.

3-4-2 Informative decisions

To transform the predictive model into an actionable model that can be used by marketeers, a different performance metric, including the costs and savings, should be used.

The probability threshold can be used as the start of a cost-sensitive approach. The choice of the probability threshold is based on the business context. If the telecom provider wants to target a large amount of customers then a low threshold should be set. As a result, more people will be predicted to churn. However, this increases the likelihood that customers who are not at risk will pass the threshold and be predicted to churn. In contrasts, if the company wants to be more efficient in spending, a higher threshold should be set.

A queue rate - precision - recall graph can be used to visualize the operational and strategic decision of how a model can be used to make informative decisions:

1. Queue Rate: The queue is the number of renewable mobile customers that have shown online behavior. This metric describes the percentage of customers that must be reviewed. It depends on the cost of treating an individual customer and the overall capacity.
2. Precision: What fraction of customers predicted to churn are truly churners?
3. Recall: What fraction of customers that actually churned are predicted to churn?

3-4-3 Benchmark

The proposed models will be compared to a benchmark model. It is common to use a random classifier to be the benchmark in a classification tasks. Random classifiers generate predictions by respecting the class distribution. The idea behind using a random classifier is that any other classifier that does not result in better predictions than randomly drawing possible outcomes, is useless.

However, in this study a hybrid approach is examined. Therefore, the benchmark is a one stage approach: the classifiers that do not utilize any results from an unsupervised learning technique. It is thus a classifier applied on top of the complete data set (i.e. customer and online behavior data set). As a result, the added value of the hybrid approach of combining unsupervised and supervised learning can be estimated.

Chapter 4

Analysis methods

4-1 Unsupervised learning techniques

4-1-1 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets [7]. t-SNE consists of two stages. During the first stage, a probability distribution is constructed such that similar objects have a high probability of being selected. In the second phase, t-SNE defines a probability distribution over all points in a low dimensional map. A detailed description of t-SNE can be found in Appendix A.

One of the pitfalls of t-SNE is the time complexity. The algorithm computes pairwise conditional probabilities and minimizes the sum of the differences of the probabilities in higher and lower dimensions. To reduce computational time, t-SNE is applied using Multicore t-SNE in Python [53].

Prior to applying t-SNE, it is critical to convert the categorical variables to dummies. Moreover, variables are standardized to make sure that all variables are treated with equal importance.

The perplexity parameter is defined using visual examination of different values of perplexity. To ensure convergence is achieved the model is implemented using 4000 iterations. Moreover, the dimension of the embedded space is set to 2 (i.e. x, y) so that the results can be visualized in a two-dimensional space.

Important to note is that it is not possible to see the relative sizes of clusters when visualizing the results of the t-SNE algorithm. The t-SNE algorithm expands dense clusters and contracts sparse ones. As a result, distances do not have a meaning in the t-SNE plot [54]. Hence, distance based unsupervised learning on top of t-SNE are not recommended. Therefore, in this study, Gaussian Mixture Model (GMM) is applied on top of t-SNE. GMM is a model based technique which also has been proven to increase clustering power [55] when used in combination with t-SNE.

4-1-2 Gaussian Mixture Clustering

A GMM is a probabilistic model that assumes all data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [56]. GMM allows for uncertainty in cluster membership as it assigns a probability for each cluster. Therefore, GMM is particularly useful when true clusters overlap and when the data is spread out. A detailed description of GMM can be found in Appendix B.

GMM is implemented using sklearn package Mixture Gaussian Mixture in Python. The GMM is based on the Gaussian density, which is a continuous density. The measurement level of the variables is therefore converted to be continuous. Moreover, variables measured at different scales should contribute equally to the model. Therefore, variables are standardized so that they are centered around 0 with a standard deviation of 1. Important to note is that for dummy variables standardization is not preferred as the mean of the dummy variables does not have interpretable properties in this study. Therefore, the interpretation of the clusters is on the non standardized data. GMM is implemented using 1000 iterations with a convergence threshold of 0.001. Moreover, each component has its own general covariance matrix (i.e. covariance type full). Based on the number of segments defined by domain experts, 7 components are fitted.

4-1-3 Latent Class Analysis

Latent Class Analysis (LCA) is a technique which can be used to identify and characterize clusters of similar cases when dealing with multivariate categorical data. LCA seeks to stratify the cross-classification table of observed (i.e. manifest) variables by an unobserved (i.e. latent) unordered categorical variable that eliminates all confounding between the manifest variables [9]. A detailed description of LCA can be found in Appendix C.

LCA is implemented with the PoLCA package in R. LCA requires polytomous variables and therefore the continuous variables are categorized [57]. This is performed by looking at the distribution of the categorical variables and defining appropriate bins which can be recoded into categories. For example, age is converted into N age categories (e.g. <18 year, 18-25 year, etc.). Moreover, the categorical variables are recoded to an increment from 1 to the maximum number of outcome categories for each variable (e.g. the N age categories are recoded to increment from 1 to N). The tolerance value for judging when convergence has been reached is set to be 0.0001. The maximum number of iterations through which the estimated algorithm will cycle is 1000. Moreover, multiple repetitions are used to automate the search for the global rather than just a local maximum of the log-likelihood function. The number of assumed latent classes in the model is 7.

4-1-4 Segmentation

Customer segmentation is the division of potential customers in a given market into discrete groups. The division can be based on customers having similar needs or characteristics. The process of segmentation begins with observing customer actions and continues with learning about the demographic and psychographic characteristics of these customers. The implementation of this process is outside the scope of this study. Therefore, domain experts are asked to identify the main characteristics of the customers groups the telecom provider is serving.

4-2 Supervised learning techniques

4-2-1 Logistic Regression

Logistic Regression (LR) is a popular model to classify observations. It is based on the logistic function [58]. The logistic function is an S-shaped curve that can take any real input x and map it into a value between 0 and 1. As a result, it can be interpret as a probability which can be used to assign classes to observations. A detailed description of LCA can be found in Appendix D.

The output of a LR is informative as it expresses not only the relevance of a predictor but also its direction of association. However, by its definition, it assumes a linear relation between the dependent and independent variables which is often not the case. A LR is therefore considered as a not very flexible classification algorithm that might oversimplify the real problem, resulting in a high bias.

Moreover, LR requires that there is little or no multicollinearity among the independent variables. This means that the independent variables should not be highly correlated with each other. In the presence of multicollinearity ridge regression can be applied. Ridge regression introduces a bit of bias to reduce the variance, pushing coefficients to zero. In the case of multicollinearity least squares is not uniquely defined but ridge regression is [5]. Therefore, along the LR approach, a ridge penalty is applied.

In this study, LR is implemented using the sklearn Logistic Regression package in Python. Each multiclass logistic model is fitted using a ridge penalty and stochastic average gradient. The regularization strength C is determined using the hyper-parameter procedure described in Section 3-2-2. Moreover, the class weights are balanced meaning that the y values are used to adjust the weights inverse proportional to the class frequencies in the input data. The predicted probabilities are transformed to the class label which the highest probability.

4-2-2 Random Forest

Random Forest (RF) is based on the concept of a decision tree. Decision trees are a high variance model as they have the risk of being tuned to the training set. The idea in RF is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much [59]. RF is a substantial modification of the bagging technique that builds a large collection of de-correlated trees and then averages them. A detailed description of RF can be found in Appendix E.

RF is implemented using the sklearn Random Forest Classifier package in Python. The function to measure the quality of a split, the maximum depth of the tree, the number of trees in the forest and the minimum number of samples required to split an internal node are determined using the hyper-parameter procedure described in Section 3-2-2. The model is implemented with at least 50 samples in the nodes and the number of features for optimal split is the square root of the number of the total number of features. Moreover, to balance the class weight the y values are used to adjust the weights inverse proportional to class frequencies in the input data.

Important to note is that RF is trained using bootstrap aggregation, where each new tree is fit from a bootstrap sample of the training observations. This gives the possibility to calculate

the average error for each training observation using the predictions from the trees that do not contain the training observation in their respective bootstrap sample. This is called the out-of-bag error. It allows RF to be fit and validated whilst being trained. However in this study the validation procedure as described in Chapter 3 will be used for RF.

The strength of a RF is that they are quick to train and minimal input preparation is needed as it can deal with categorical and continuous features. Moreover, as a RF reduces the variance of a decision tree it is a rather robust algorithm.

4-2-3 Gradient boosted tree

Gradient boosted trees, like RF, is an ensemble model. The base learners are decision trees. However, in contrast to RF, it does not use bagging as the ensemble method but boosting. Boosting is an ensemble technique that converts weak learners to strong learners by adding models on top of each other and correcting the errors made by the previous model.

This study uses one of the available implementation of the gradient boosted decision trees: XGBoost [10]. A detailed description of XGBoost can be found in Appendix F.

XGBoost are powerful due to its accuracy and feasibility of the algorithm and can be implemented using the sklearn XGBClassifier in Python. The following hyper-parameters are defined using the hyper-parameter procedure described in Section 3-2-2: minimum loss reduction required to make a further partition on a leaf node of the tree, maximum tree depth for base learners, subsample ratio of the training instance, subsample ratio of columns when constructing each tree, boosting learning rate and the minimum loss reduction required to make a further partition on a leaf node of the tree. Moreover, the minimum sum of instance weight(hessian) needed in a child is set to be 50. Since this study applies multiclass classification, the objective is set to softmax. Lastly, the positive and negative weights are balanced by the sum of the negative cases divided by the sum of the positive cases.

4-2-4 Heterogeneous ensemble classifier

Heterogeneous ensemble classifier combines multiple individual classification models and use the average predicted probabilities to predict the final class label. In this study, the individual models are LR, RF and XGBoost. A voting ensemble for the classification is created using the sklearn VotingClassifier class in Python. First the instances of the three standalone models are created. Next, the Voting Classifier is used to wrap the models and average the predictions of the sub-models when asked to make predictions for new data.

Within the ensemble, for each of the individual models a weight is provided. In this study weights are determined by the best-worst weighted voting system as presented by [60]. Best-worst weighted voting sets the weight for the best member classifier to 1, set the weight for the worst member classifier to 0, and have the weights for other member classifiers linearly proportional to their training performance. The predicted class probabilities for each classifier are collected, multiplied by the classifier weight and averaged. The final class label is then derived from the class label with the highest average probability.

Chapter 5

Data

In this chapter the data set used for the experiments is described. The data collection and pre-processing steps are explained. Moreover, the results of the exploratory data analysis are provided.

5-1 Data collection

The database of the telecom provider consists of customers who have a mobile phone subscription, customers having a residential subscription (i.e. a internet, television and a fixed phone connection) and customers having both. In this study, only customers having at least a mobile subscription are included. Moreover, only customers who can extend their contract are included. Extending a contract is possible starting from 4 months before the end of a contract. Lastly, only customers who have shown online behavior on the website are included. To summarize, the data set used in this study consists of customers who at least have a mobile subscription, are allowed to renew or terminate their contract and have shown online behavior.

A data set containing customer information and a data set containing contractual information is stored each day. The data sets are aggregated on customer ID in order to obtain the customer information data set. During this process, irregularities were encountered. For example, the aggregation results in a customer having both a sim only and a handset subscription. To resolve this, the most frequent occurring subscription over all customers is imputed. Matching the data set with the data set from the prior day, the customers that are not subscribed can be identified. The information of these customers will be stored in a historical table. The customers in this historical table are used within this study to identify churners and renewers. Moreover, socio-demographic data was collected from the Statistics Netherlands database [61]. Ultimately, the features from this data set are available for each customer specifically instead of neighborhood-specific. However, this was not the case and therefore the more general Statistics Netherlands data set is used. The socio-demographic data set was aggregated on the customer information data set using the ZIP code.

An analytics tool is running on the website of the telecom provider which registers browsing behavior. The website aims at selling new subscriptions and providing information to existing customers. Browsing behavior and their respective cookie IDs is stored in the online behavior data set. Customers who logged into the website with a username and password are matched with the cookie ID available from the online behavior data. The obtained customer ID can be aggregated on the customer ID of the customer data to obtain the final data set used in this study. For a more detailed description on the collection of the online behavior data set see Section 5-3.

It is important to stress that although the data sets are retrieved with care, correctness cannot be guaranteed. Due to the complexity of the data collection, the results should be tested in real life. Moreover, a model that is optimal for this data set might not be applicable to the data gathered the following month. For example, a customer might have had contact with the telecom provider. This information is not included if the data set is not updated which results in unreliable predictions.

5-2 Customer data

5-2-1 Data cleaning

The collected customer data contains many features that are irrelevant for predicting churn and only the relevant features for churn prediction are extracted for the purpose of this study.

The selected features from the customer data set included are age, gender, 4-digit postal code and the kind of connection available to the customer (i.e. fiber connection or no fiber connection).

For the contractual data set the features included are the size of the SMS bundle, minutes bundle and data bundle. It is important to mention that in an early stage of this study the data set contained the bundle size of the new contract for renewers. After running several analyses it was found out that the bundle size was discriminating churners from renewers. This was because renewing a contract almost always results in a higher bundle. By including the bundle of the new contract, the bundle features are not predetermined variables anymore. To be able to make reliable predictions, the data set had to be generated again including the data and minutes bundle from the contract when the customer decided to renew.

In addition, a feature is included to identify if a customer is a residential customer (RSD) meaning that the customer has a internet, television, fixed phone and a mobile phone subscription. RSD customers benefit from a discount, free minutes, extra data, additional television channels, or free television recording. A feature is included to indicate the type of advantage. Moreover, the type of the current mobile subscription of the customer is included (i.e. handset or sim-only type of contract). Lastly, the connection type (i.e. copper or fiber) is also included.

There are more than hundred features available in the socio-demographic data set, but not all are expected to be informative for predicting churn. A hypothetically relevant feature is the education level of a customer. The churn likelihood may be diverse by different education levels as each education level has its own evaluation and expectation of an operator. Other features selected are income level, household size and urbanity. Education level is measured as

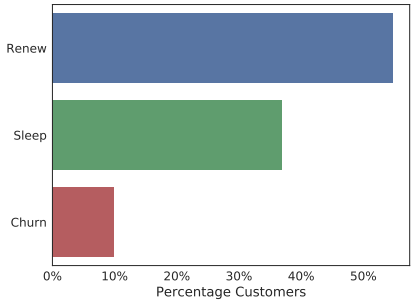


Figure 5-1: Percentages of customers per action type.

the percentage of people in a neighborhood that are having a low, middle or high education level. Income level is measured as the average income in a particular neighborhood. The household composition is the average household composition in the area.

Lastly, urbanity is measured as the concentration of people in a particular area where an area falls into one out of five classes. Based on the definition of Statistics Netherlands [62], the following classification is used:

1. Class 1: >= 2500 addresses per km^2
2. Class 2: 1500 - 2500 addresses per km^2
3. Class 3: 1000 - 1500 addresses per km^2
4. Class 4: 500 - 1000 addresses per km^2
5. Class 5: < 500 addresses per km^2

5-2-2 Feature engineering

The goal of feature engineering is to create informative features. For example, the birth date of a customer is not informative but age is. Moreover, a feature is included indicating the number of days before the end of the contract at the moment of the action. This feature is the difference between action date and contract end date. This feature is potentially informative for predicting churn as customers whose contract end date is near are more likely to either terminate or renew their contract.

5-2-3 Exploratory data analysis

The final features from the customer data set are listed in Table 5-1. As illustrated in Figure 5-1, the classes are not equally represented in the data set. Only 10% of the data set consists of the class of interest: churn. As a result, this study deals with an imbalanced classification task. In Section 3-3 the tactic used in this study to handle the class imbalance problem is discussed.

There are several features that show an interesting relation with the outcome variable action type. First of all, the distribution of the number of days before the end of the contract is different for each action type (see Figure 5-2). More than 95% of the customers make the

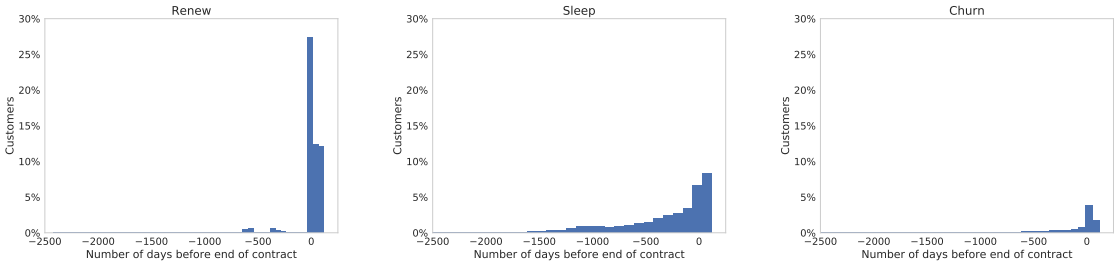


Figure 5-2: Distribution of days end of contract per action type.

decision to renew their contract before the actual contract end date (i.e. a positive value of days end of contract). This is in contrast to sleepers and churners where this is respectively 29% and 51%. To conclude, most customers make the decision to renew their contract before the end of their contract while they make the decision to churn when their contract already has expired. Perhaps, renewers are eager to change phones or bundles. However, to confirm this, more information is needed on for example the type of phone and the amount of minutes and data a customer uses. Unfortunately, this information is not available.

Secondly, as seen in Figure 5-3, renewers have a larger data bundle than churners and sleepers. To be more specific, 39% of the renewers have a data bundle more than 1000 MB while this is the case for only 20% of the churners. The data bundle is therefore likely to be informative for separating renewers from churners and sleepers. The minutes bundle is, except for the outliers, equal for each of the action types. Moreover, for the SMS feature there is not a difference between the action types as for this telecom provider (almost) all customers have an unlimited SMS bundle.

Figure 5-4 illustrates the relationship between several categorical features and the action types. Figure 5-4 demonstrates that the distribution of male and females is equal among the three action types. However, there seems to be a difference among the action types for the availability of a fiber connection. In relation to churners and sleepers, renewers demonstrate a larger rate of fiber connection. Moreover, in relation to churners and sleepers, the renew class

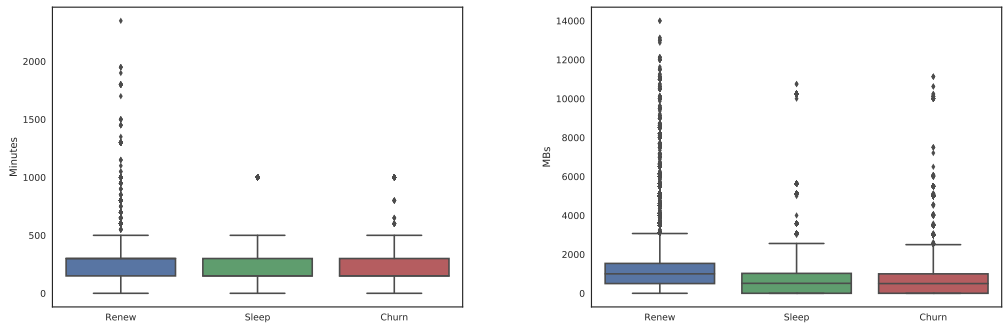


Figure 5-3: Boxplot of number of minutes and MB's per action type.

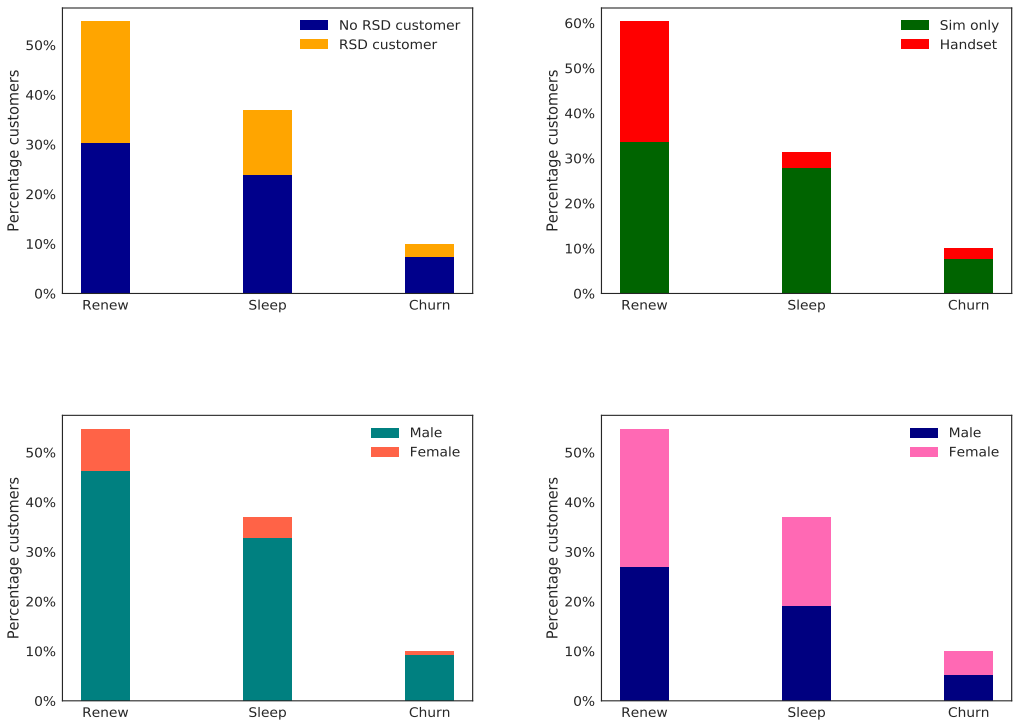


Figure 5-4: Percentages of customers per action type and class of the categorical variable. Note a RSD customer is a customer who has a internet, television, fixed phone and a mobile phone subscription

demonstrates a higher rate of hand set contracts. Especially the sleeper class demonstrates a large rate of sim only contracts. Lastly, in relation to the churners and sleepers, the renew class demonstrates a larger rate of residential customers.

5-3 Online behavior data

5-3-1 Data cleaning

Online behavior data consists of information on each customer click, such as the date and hour of the clicks, the URLs of the visited pages and a customer identifier. An example of the raw online behavior data set is shown in Table 5-2. The example illustrates a customer who views three pages in six clicks in two days.

First of all, irrelevant elements in the data set should be eliminated. The raw data set contains all requests sent to the server. However, a large part of these requests are not relevant for predicting churn (e.g., image requests). From the complete online behavior data set, 116 pages are selected to be relevant. From these 116 pages, twelve are identified as churn or renew pages as those pages imply a churn or renew request. These twelve pages are therefore called movement pages.

Variable name	Description
Customer characteristics	
Age	Age of the customer
Gender	Gender of the customer
Fiber	If a fiber connection is available
Education	Expected education level of the customer
Income	Expected income of the customer
Household	Expected household composition of the customer
Urbanity	Concentration of people based on address density
Action type	The type of action (i.e. sleep, renew, churn)
Contractual characteristics	
SMS	Size of SMS bundle
MB	Size of data bundle
Minutes	Size of minutes bundle
Type	Handset or sim only
Days end of contract	The number of days before the end of the contract at the moment of the action
Combination Benefits	The type of combination benefits (e.g. discount, recorder)
Residential customer	Whether the customer has combined mobile and non-mobile products
Connection type	What kind of connection type a customer uses (i.e. copper or fiber)

Table 5-1: Final list of customer data features.

Customer identification

To track the sequence of activities performed by the same customer, customer identification is needed. This stage maps each customer's actions to an identifier representing the customer behind the action. When not disabled, cookie data is available for each customer. Cookie data are designed to hold a modest amount of data specific to a particular client and can therefore be used for customer identification. In this study, the focus is on registered clients who already have a subscription from the telecom provider. Therefore, customer identification is accomplished by matching the cookie data of registered customers with their website login. Important to note is that in some cases it is impossible to identify a customer because they do not login. As a result, the data set used in this study contains less customers than that based on the selection rules would be expected.

Cookie	Date	Time	URL
12345	01-02-2018	09:00	'telefoons:mobiel-ret:shop:persoonlijk'
12345	01-02-2018	09:00	'simkaart:instellingen:instellingen:mobiel:mijn-provider'
12345	01-02-2018	09:00	'samenvoegen:mobiel:mijn-provider:persoonlijk'
12345	01-02-2018	15:00	'telefoons:mobiel-ret:shop:persoonlijk'
12345	01-02-2018	15:00	'samenvoegen:mobiel:mijn-provider:persoonlijk'
12345	05-02-2018	12:00	'telefoons:mobiel-ret:shop:persoonlijk'

Table 5-2: An example of raw customer access log data entries.

Sessionisation

Sessionisation performs a segmentation of customer activity records from each identified customer into sessions. Unfortunately, there is no formal identifier to mark session boundaries. However, there are several techniques available. A widely used method is to use time-out rules to distinguish sessions of the same customer. The standard time-oriented method defines a constant threshold, upon which if a customer surpasses that time it is considered to end that session [63]. A widely used threshold for starting a new session is 30 minutes. However, the available data set only logs data on a hourly basis. As a result it is impossible to define the exact order of visits of a customer within a hour. Moreover, customers do not visit the website of the telecom provider on multiple hours per day and therefore the online behavior is collected as the number of views per day.

The online behavior data is available from June 2017 onwards. For each customer an interval of 29 days is observed and analyzed. To give the telecom provider the time to communicate a churn reduction message with the customer, a 3 days communication period is incorporated. To define the observation period a different strategy for each of the outcome classes is used (see Figure 5-5). For customers who made a churn or renew request, the date of the request is assumed to be the end of the 29 days observation and the 3 days communication period. Thus, 32 days before the action date, is the start of the observation period for renewers and churners.

It is complicated to define the observation and communication period for sleepers as they do not perform an action. A sleeper is defined as a mobile customer who can renew its contract, visited at least one of the pages and who did not undertake a churn or renew request. Churners and renewers, perform the action (i.e. churn or renew) on average 18 days after they last visited a movement page. The action date of a sleeper is therefore defined to be 18 days after the last day they visited a movement page. The evaluation day and observation period is calculated like the evaluation and observation period for churners and renewers. Figure 5-5 illustrates the construction of the online behavior data stream for each of the action types as described in this section. Moreover, in Table 5-3 the final representation of the online behavior data is shown.

The above described approach serves several purposes. First of all, it makes the analysis time independent. Moreover, it is expected that the most informative days before the action are selected (i.e. 29 days before the action). This makes the input matrix less sparse and the analysis potential less prone to noise. However, the approach also has some drawbacks. First of all, the observation period is differently defined for sleepers as for the churn and renew class. As a result, it could be that it easy to separate sleepers from the other two classes.

customer id	Date	URL	Views
12345	01-02-2018	'telefoons:mobiel-ret:shop:persoonlijk'	2
12345	01-02-2018	'simkaart:instellingen:instellingen:mobiel:mijn-provider	1
12345	01-02-2018	'samenvoegen:mobiel:mijn-provider:persoonlijk'	2
12345	05-02-2018	'telefoons:mobiel-ret:shop:persoonlijk'	1

Table 5-3: An example of the final customer access log data representation.

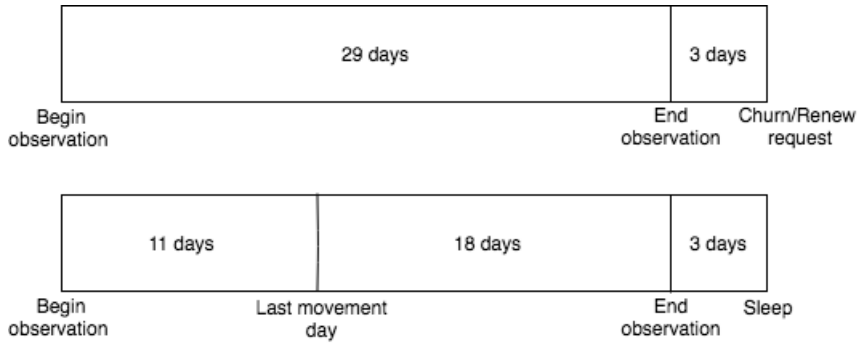


Figure 5-5: Structure of the online behavior data for each of the action types.

Moreover, incorporating the 3 days communication period and not taking the information that happens in those 3 days might result in a too optimistic model. For example, if a customer is predicted to be a sleeper but he contacts the telecom provider in the communication period, this information is not taken into account because this information is not available. On the other hand, the 3 days communication period is short and it is impossible to know when the 3 days communication period is. As a result, the model might not be able to accurately predict which customers are likely to churn.

5-3-2 Feature engineering

To improve the predictive power of the model, feature engineering is performed. Based on a study of Van den Poel [37], several online behavior based features are extracted and used for the classification. Van den Poel [37] found that online variables are important for classifying customers according to their online purchase behavior. These variables are potentially important for classifying churners as well. The included features can be divided into general and detailed online measures. All learned features are listed in Table 5-4.

General online behavior features measure the online behavior data at a general level. Features that are extracted in this category are the total number of days a customer visited at least one page and the average number of days a customer has visited the site. Moreover, based on the number of visits per day, the total number of visits and the average number of visits per day is calculated. In addition, the total number of different categories visited is calculated as well as the average number of different categories visited per day.

Detailed online behavior measures are features on a more detailed level. Several categories based on the content of the 116 pages are defined:

- Shop
- Account
- Search
- Forum
- Mobile contracts
- Contact
- Combination benefits
- Terminate contract
- Apps
- Webmail
- Homepage
- Internet - Television - Phone contracts

Consequently, for each of these categories the general online behavior features described above are calculated as well as two extra variables. The extra variables are, the total number of

Variable name	Description
General online behavior features	
TotalDays	Total number of days site visited
AverageDays	Average number of days site visited
TotalViews	Total number of views
AverageViews	Average number of views per day site visited
TotalDif	Total number of different pages visited
AverageDif	Average number of different pages visited per day
TotalPageLast	Total number of pages visited during last visit day
Detailed online behavior features	
TotalViewsCategory	Number of views concerning category
TotalDaysCategory	Number of days pages concerning category visited
PagePercentageCategory	The total number of pages viewed concerning category divided by the total number of views
LastPageCategory	Total number of views concerning category on last day of site visit
LastPercentageCategory	The total number of pages viewed concerning category on last day of site visit divided by the total number of views

Table 5-4: Final list of online behavior features.

pages viewed concerning the category divided by the total number of views and the total number of pages viewed during the last day visited concerning the category divided by the total number of views on that day.

5-3-3 Exploratory data analysis

The final features from the online behavior data set are listed in Table 5-4.

As seen in Figure 5-6, the majority of renewers visit the website on more days than the majority of sleepers and renews. As a result, the total days feature might separate renewers from the other two classes. This is also in line with the total views boxplot in Figure 5-6. Renewers view on average more pages than the other two classes. Moreover, sleepers visit the least amount of pages and are therefore least active. As expected, Figure 5-7 illustrates that renewers visit more pages regarding the shop of the telecom provider than the other two classes. Moreover, the fraction of the total views on the account page is larger for churners than for the other two classes.

From the plots can also be seen that there are outliers present in the data set. An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. As finding the rare instances (i.e outliers) can be more insightful than finding the common patterns [64] it is decided to include the outliers in this study.

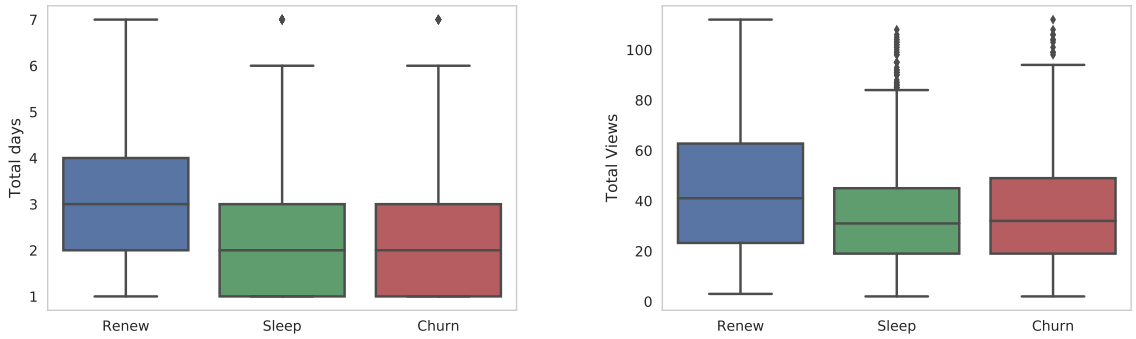


Figure 5-6: Boxplot for total days and total views per action type.

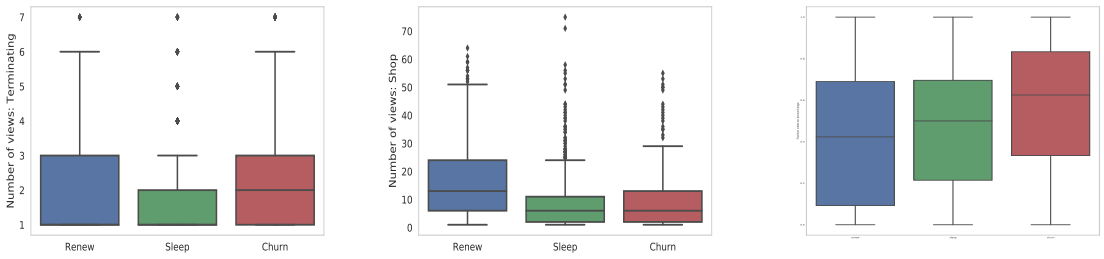


Figure 5-7: Boxplot of total views regarding terminate contract pages, total views regarding the shop and the fraction views on account pages per action type.

Chapter 6

Results

This chapter discusses the results from the methods described in Chapter 4 applied to the data set described in Chapter 5. Several combinations of the methods are implemented. In Table 6-1 each of the approaches are specified. Moreover, this table contains references to specific sections in which the results can be found.

	Unsupervised learning		**Supervised learning**		
Approach	**Methods**	**Data set**	**Methods**	**Data set**	**Section**
Benchmark	None	None	All classifiers	Customer and online data	Section 6-1
t-SNE-GMM	t-SNE and GMM	Renewable customers	All classifiers	Online data	Section 6-2
GMM-small	GMM	Renewable customers	All classifiers	Online data	Section 6-3
GMM-large	GMM	All mobile customers	All classifiers	Online data	Section 6-4
LCA	LCA	Renewable customers	All classifiers	Online data	Section 6-5
Segmentation	Segmentation	Renewable customers	All classifiers	Online data	Section 6-6
LCA clusters as feature	LCA	Renewable customers	All classifiers	Customer data + Online data + Cluster label	Section 6-7
Probability threshold	None	None	Random Forest	Customer and online data	Section 6-8

Table 6-1: Overview of applied approaches.

6-1 Approach: benchmark

As discussed in Section 6-1, the benchmark is a classification model that does not utilize any results from an unsupervised learning technique. Table 6-2 presents the Area Under the Curve (AUC) values for each benchmark classification model. XGBoost and the ensemble outperforms Logistic Regression (LR) and Random Forest (RF).

Performance metric	LR	RF	XGBoost	Ensemble
AUC churn	0.54	0.91	0.95	0.95
AUC renew	0.76	0.95	0.99	0.99
AUC sleep	0.78	0.96	0.99	0.99

Table 6-2: AUC of the baseline classifiers on the test set.

The AUC values reflect that the classification models are most effective at predicting the renew and sleep class and the least effective at predicting the churn class (see table 6-2). This is in line with the expectations due to the class imbalance problem. Especially, the performance of LR for the churn class is insufficient and only slightly better than random prediction. This suggest that the data encountered in this study can not appropriately be fitted by one LR. The results of the benchmark models XGBoost and the ensemble indicate that it will be challenging to increase the predictive performance as it is already very close to perfect. The top 10 most important features of the model constructed with XGBoost are illustrated in Figure 6-1. Interpreting and analyzing important features enables to gain profound knowledge in the reason why customers are predicted to churn, renew or sleep. The two most important features, by far, are the number of days before the end of the contract and the size of the data bundle.

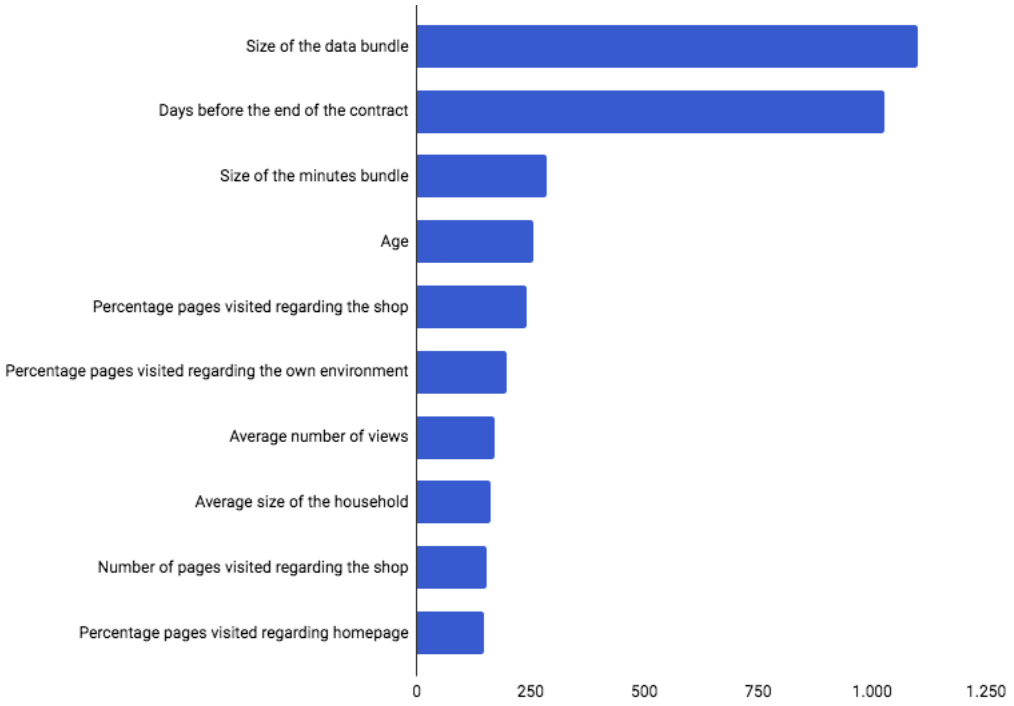


Figure 6-1: Feature importances for the XGBoost benchmark model.

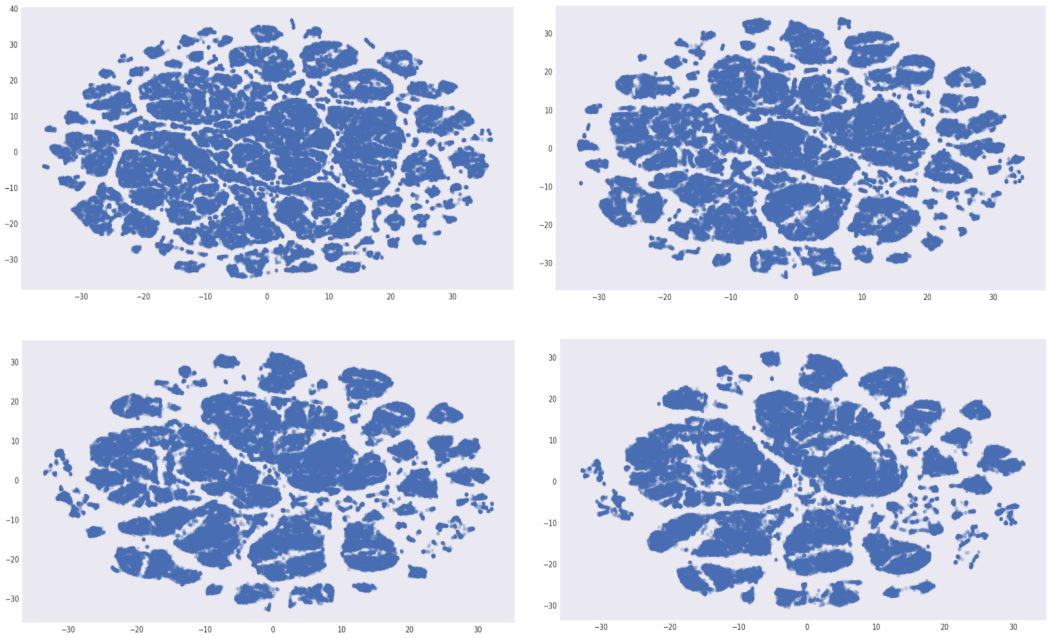


Figure 6-2: Results of t-SNE with different values of perplexity. From the top left figure to the bottom right figure perplexity values of 20, 40, 70 and 100 are used.

6-2 Approach: t-SNE-GMM

Unsupervised learning

t-SNE is applied on the renewable mobile customers data set using the settings as specified in Section 4-1-1. The final perplexity value is defined using visual examination of the plots in Figure 6-2. There is no clear separation for any of the perplexity values. However, the observations are more evenly spread for larger perplexity values. Therefore, a perplexity value of 100 is used within this study.

The x, y coordinates obtained with t-SNE are the inputs for the Gaussian Mixture Model (GMM). As specified in Section 4-1-1, GMM is trained on 7 components. The results are illustrated in Figure 6-3. There are more components visible than the 7 that are fitted. This explains why the clusters overlap the groups than when visual examination is used. As seen in Figure 6-4, in relation to the other clusters, clusters 2 and 4 demonstrate a higher rate of churn. Moreover, in relation to the other clusters, cluster 1 and 5 demonstrate a large rate of renewers.

In Figure 6-5 the distribution of three customer characteristics is illustrated for each of the clusters. Clusters 2 and 5 consist of customers with a small amount of minutes. Moreover, Figure 6-5 illustrates that cluster 1 mainly consists of customers with a non-residential contract. Lastly, the majority of customers that have a fiber connection are in cluster 5.

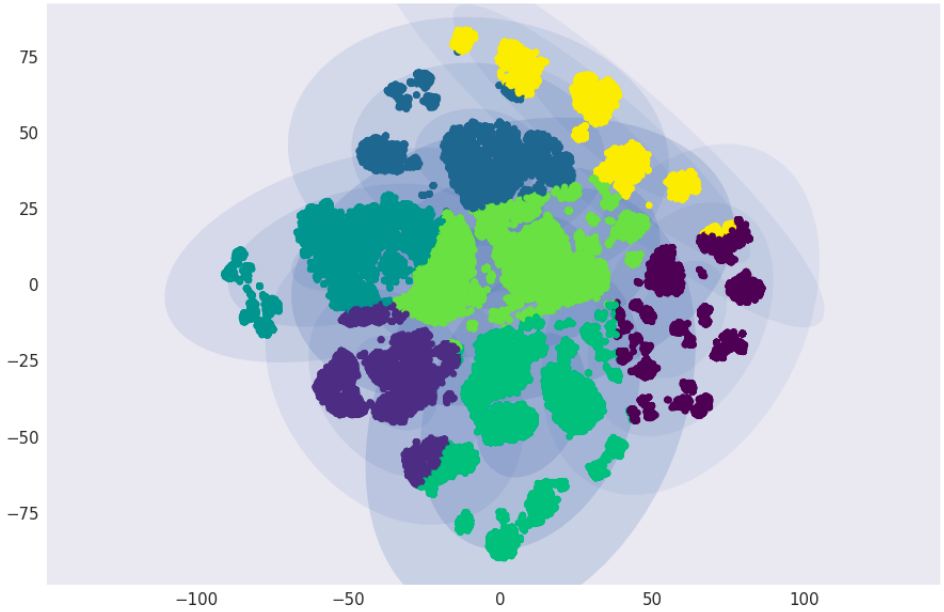


Figure 6-3: Results of 7 component GMM on the results of t-SNE.

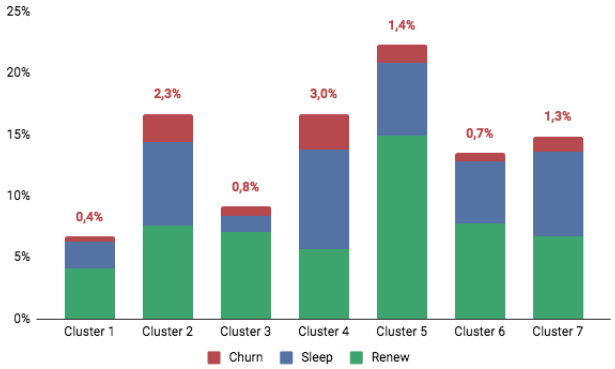


Figure 6-4: Distribution of each of the action types within the clusters obtained with t-SNE and GMM. The percentages indicate the percentage of churners per cluster.

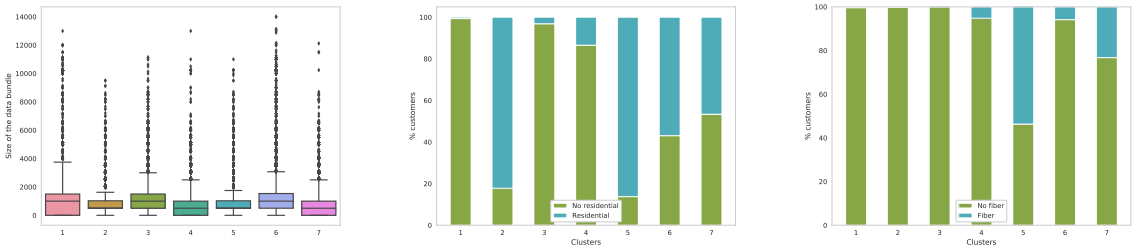


Figure 6-5: Distribution of the size of the data bundle, residential customer and the availability of fiber for each of the clusters obtained with the t-SNE and GMM method.

	% churn in test set	Cluster 1 5.86%	Cluster 2 13.67%	Cluster 3 8.70%	Cluster 4 17.69%	Cluster 5 6.39%	Cluster 6 5.10%	Cluster 7 8.44%	Weighted average	Benchmark
LR	AUC churn	0.59	0.57	0.57	0.54	0.56	0.68	0.57	0.57	0.54
	AUC renew	0.76	0.69	0.59	0.8	0.67	0.77	0.79	0.73	0.76
	AUC sleep	0.77	0.77	0.57	0.76	0.71	0.75	0.78	0.73	0.76
RF	AUC churn	0.69	0.76	0.78	0.78	0.78	0.75	0.76	0.77	0.91
	AUC renew	0.86	0.90	0.70	0.91	0.86	0.89	0.90	0.87	0.99
	AUC sleep	0.88	0.90	0.74	0.9	0.88	0.88	0.90	0.87	0.95
XGBoost	AUC churn	0.77	0.84	0.87	0.84	0.86	0.84	0.83	0.85	0.95
	AUC renew	0.92	0.94	0.81	0.94	0.9	0.94	0.95	0.92	0.95
	AUC sleep	0.92	0.92	0.85	0.91	0.92	0.93	0.93	0.91	0.99
Ensemble	AUC churn	0.76	0.83	0.84	0.84	0.82	0.81	0.83	0.83	0.95
	AUC renew	0.93	0.95	0.82	0.95	0.91	0.94	0.94	0.92	0.99
	AUC sleep	0.93	0.92	0.87	0.92	0.92	0.94	0.93	0.92	0.99

Table 6-3: AUC of the predictive models for each of the clusters obtained with t-SNE and GMM (on the test set).

Supervised learning

Which customers are likely to churn, renew, or sleep is predicted for each cluster. In Table 6-3 the AUC values for the t-SNE and GMM method are shown. For RF, XGBoost and the ensemble, the benchmark approach outperforms the t-SNE and GMM approach. For LR the AUC for the churn class is higher than that for the benchmark approach.

6-3 Approach: GMM-small

Unsupervised learning

The results of the GMM clustering on the renewable mobile customer data set are illustrated in Figure 6-6. Each cluster contains churners, sleepers and renewers. In relation to the other clusters, clusters 2 and 3 demonstrate a larger rate of churn and cluster 4 demonstrates a larger rate of renewers.

As illustrated in Figure 6-7, the majority of residential customers are in cluster 2, 3, 4 and 6. The size of the minutes bundle is homogeneous for the clusters. Cluster 4 illustrates that the majority of customers have a larger size of the data bundle than the customers in the other clusters. This is in line with the distribution of the action types as cluster 4 demonstrates a larger rate of renewers.

Supervised learning

In Table 6-4 the AUC values for the GMM small approach are shown. For RF, XGBoost and the ensemble, the benchmark approach outperforms the GMM small approach. For LR the AUC for the churn class is higher than for the benchmark approach.

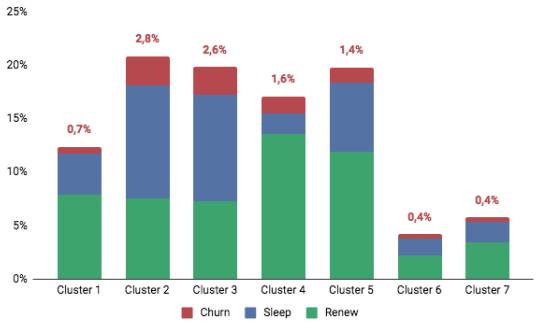


Figure 6-6: Distribution of each of the action types within the clusters obtained with the GMM small approach. The percentages indicate the percentage of churners per cluster.

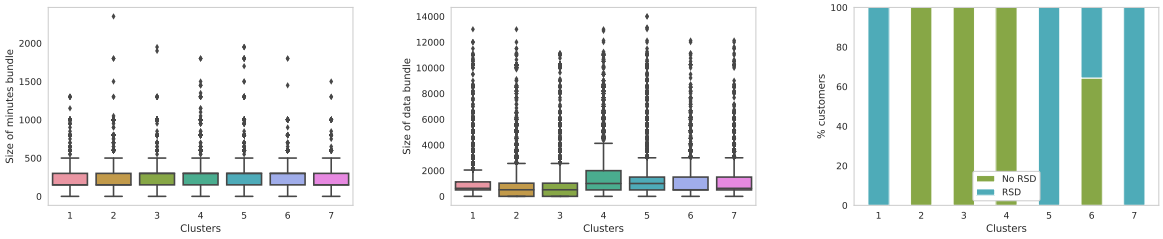


Figure 6-7: Distribution of minutes, MB and RSD customer for each of the clusters obtained with the GMM small approach.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Weighted average	Benchmark
	% churn in test set	5.40%	13.27%	13.04%	9.35%	6.96%	9.96%	6.65%		
LR	AUC churn	0.53	0.55	0.56	0.68	0.59	0.57	0.55	0.58	0.54
	AUC renew	0.73	0.81	0.81	0.65	0.73	0.72	0.74	0.75	0.76
	AUC sleep	0.73	0.77	0.77	0.74	0.74	0.73	0.74	0.75	0.76
RF	AUC churn	0.74	0.75	0.74	0.79	0.71	0.74	0.72	0.74	0.91
	AUC renew	0.86	0.91	0.91	0.75	0.88	0.84	0.85	0.86	0.99
	AUC sleep	0.87	0.89	0.89	0.78	0.88	0.84	0.85	0.86	0.95
XGBoost	AUC churn	0.82	0.85	0.82	0.88	0.86	0.81	0.86	0.84	0.95
	AUC renew	0.93	0.95	0.95	0.84	0.93	0.90	0.92	0.92	0.95
	AUC sleep	0.92	0.92	0.91	0.85	0.93	0.88	0.91	0.90	0.99
Ensemble	AUC churn	0.86	0.84	0.82	0.85	0.80	0.81	0.81	0.83	0.95
	AUC renew	0.93	0.95	0.95	0.83	0.93	0.93	0.94	0.92	0.99
	AUC sleep	0.93	0.92	0.91	0.87	0.93	0.92	0.94	0.91	0.99

Table 6-4: AUC of the predictive models for each of the clusters obtained with the GMM-small approach (on the test set).

6-4 Approach: GMM-large

Unsupervised learning

The distribution of each of the action types within the clusters obtained with the GMM large approach are illustrated in Figure 6-8. Cluster 4 has a relatively small sample size and does

not contain churners. In relation to the other clusters, clusters 1, 6 and 7 demonstrate a larger rate of churn.

In Figure 6-9 the distribution of three static features for each of the clusters are illustrated. The customers in clusters 1 and 6 have, on average, less MB's than the customers in the other clusters. Moreover, this plot illustrates that the distribution of residential customers is homogeneous among the clusters. The majority of customers that received a discount are in clusters 3 and 5.

Comparison GMM-small and GMM-large

The results of the GMM clustering on the full data set are compared with the clusters obtained from the clustering on the renewable mobile customer data set as explained in Chapter 3. The Adjusted Rand Index between the two clustering approach is 0.6. Random cluster assignments have an Adjusted Rand Index score close to 0.0, while similar clusterings have a score of 1.0. The two clustering approaches are therefore assigning most customers to the same cluster but not all. The customers on which the two approaches do not agree on, are not only the non-renewable customers. This suggests that the renewable mobile customer data set is not a cluster of the full mobile customer data set.

Supervised learning

The class to which a customer belongs to is predicted for each cluster. The results of the different supervised learning methods can be found in Table 6-5. The highest AUC is obtained using XGBoost. For all classifiers, the benchmark approach outperforms the GMM large approach.

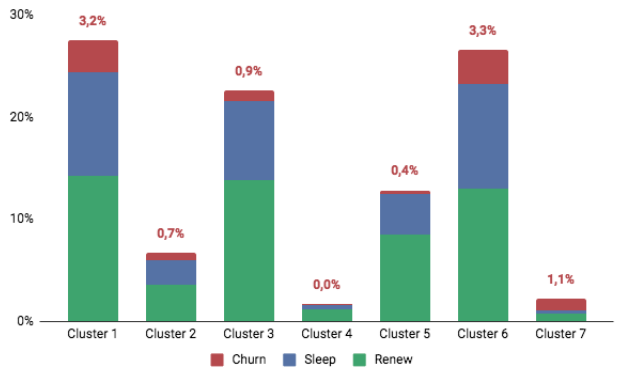


Figure 6-8: Distribution of each of the action types within the clusters obtained with GMM on the full mobile customer data set. The percentages indicate the percentage of churners per cluster.

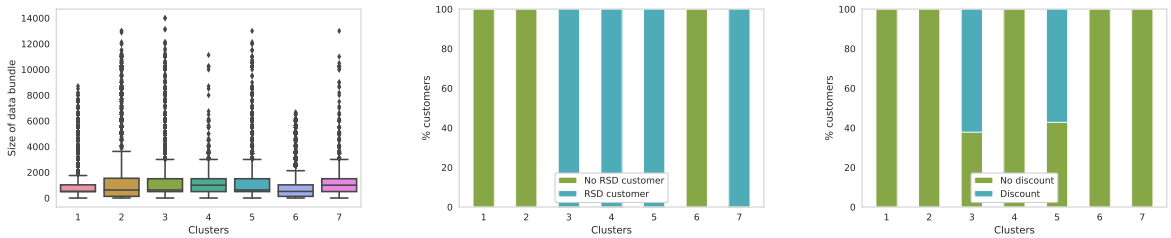


Figure 6-9: Distribution of the size of the data bundle, residential customer and combination benefit discount feature for each of the clusters obtained with GMM on the full mobile data set.

	% churn in test set	Cluster 1 11.59%	Cluster 2 12.40%	Cluster 3 4.19%	Cluster 4 2.70%	Cluster 5 2.85%	Cluster 6 12.40%	Cluster 7 52.60%	Weighted average	Benchmark
LR	AUC churn	0.59	0.52	0.47	0.58	0.48	0.56	0.48	0.51	0.54
	AUC renew	0.76	0.80	0.73	0.73	0.71	0.75	0.68	0.72	0.76
	AUC sleep	0.76	0.78	0.73	0.77	0.72	0.76	0.74	0.75	0.76
RF	AUC churn	0.77	0.69	0.76	0.56	0.77	0.75	0.69	0.71	0.91
	AUC renew	0.89	0.88	0.90	0.84	0.88	0.89	0.75	0.81	0.99
	AUC sleep	0.90	0.88	0.90	0.84	0.89	0.89	0.81	0.85	0.95
XGBoost	AUC churn	0.83	0.77	0.83	0.78	0.82	0.85	0.82	0.82	0.95
	AUC renew	0.92	0.93	0.93	0.88	0.92	0.93	0.89	0.91	0.95
	AUC sleep	0.92	0.92	0.93	0.88	0.93	0.93	0.85	0.88	0.99
Ensemble	AUC churn	0.83	0.78	0.79	0.68	0.80	0.82	0.84	0.82	0.95
	AUC renew	0.92	0.93	0.93	0.90	0.94	0.93	0.92	0.92	0.99
	AUC sleep	0.92	0.91	0.94	0.88	0.94	0.92	0.86	0.89	0.99

Table 6-5: AUC of the predictive models for each of the clusters obtained with the GMM-large approach (on the test set).

6-5 Approach: LCA

Unsupervised learning

The distribution of the clusters found by Latent Class Analysis (LCA) is illustrated in Figure 6-10. Cluster 6 represents a third of the customers. In relation to the other clusters, clusters 3 and 4 demonstrate a larger rate of renewers. Moreover, cluster 5 demonstrates a higher rate of sleepers and clusters 6 and 7 a higher rate of churners.

As seen in Figure 6-11, cluster 5 illustrates that the majority of customers that have a fiber connection are in cluster 5. Moreover, compared to the other clusters, cluster 4 demonstrates a slightly larger rate of customers that have a handset contract. The customers in clusters 1, 5 and 6, on average, have less MB's than the customers in the other clusters.

Supervised learning

Which customers are likely to churn, renew, or do nothing is predicted for each cluster. In Table 6-6 the AUC values for the LCA method are shown. For RF, XGBoost and the ensemble, the benchmark approach outperforms the LCA approach. For LR the AUC for the churn class is higher than for the benchmark approach.

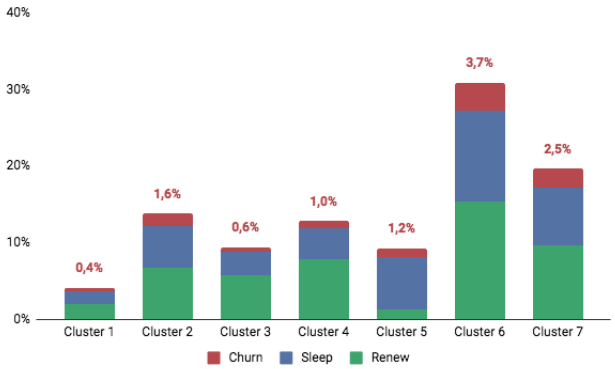


Figure 6-10: Distribution of each of the action types within the clusters obtained with LCA. The percentages indicate the percentage of churners per cluster.

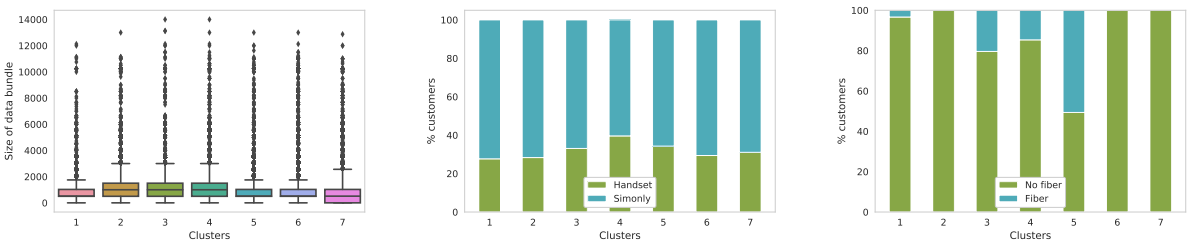


Figure 6-11: Distributions of size of the data bundle, sim only feature and fiber feature for each of the clusters obtained with LCA.

		Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Weighted average	Benchmark
	% churn in test set	10.22%	11.67%	6.71%	7.43%	5.65%	11.87%	12.53%		
LR	AUC churn	0.53	0.63	0.67	0.62	0.60	0.58	0.56	0.59	0.54
	AUC renew	0.77	0.77	0.75	0.72	0.74	0.76	0.75	0.75	0.76
	AUC sleep	0.78	0.75	0.74	0.71	0.74	0.76	0.76	0.75	0.76
RF	AUC churn	0.74	0.76	0.70	0.74	0.71	0.75	0.76	0.74	0.91
	AUC renew	0.84	0.90	0.88	0.88	0.90	0.89	0.87	0.88	0.99
	AUC sleep	0.89	0.90	0.87	0.88	0.91	0.90	0.90	0.89	0.95
XGBoost	AUC churn	0.78	0.85	0.83	0.82	0.85	0.84	0.85	0.83	0.95
	AUC renew	0.93	0.94	0.93	0.92	0.93	0.93	0.93	0.93	0.95
	AUC sleep	0.92	0.93	0.93	0.92	0.93	0.92	0.93	0.93	0.99
Ensemble	AUC churn	0.82	0.82	0.92	0.82	0.79	0.81	0.82	0.83	0.95
	AUC renew	0.93	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.99
	AUC sleep	0.92	0.92	0.93	0.92	0.93	0.92	0.93	0.92	0.99

Table 6-6: AUC of the predictive models for each of the clusters obtained with LCA (on the test set).

6-6 Approach: segmentation

Unsupervised learning

The segments are labeled by domain experts from the telecom provider. The segments and their description are illustrated in Table 6-7. The distribution of action types over the segments are illustrated in Figure 6-12. The experts are not to able to identify the most important combinations of customer characteristics as most customers belong to the "other" segment. Segment 2 consist of many churners. Segment 1 does only contain a few churners. The description of segment 1 should therefore not contain customer characteristics that are closely related to churners. Hence, the characteristics of the customers in cluster 1 are expected to discriminate renewers and sleepers from churners.

Supervised learning

In Table 6-8 the AUC values for each cluster are listed. For RF, XGBoost and the ensemble, the benchmark approach outperforms the segmentation approach. For LR the AUC for the churn class is higher than that of the benchmark approach.

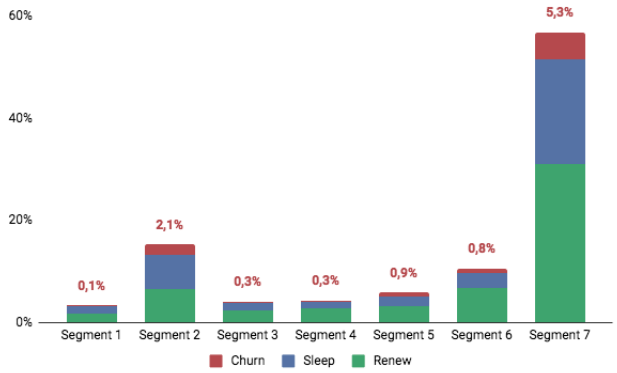


Figure 6-12: Distribution of each of the action types within the segments. The percentages indicate the percentage of churners per cluster.

Segment	Characteristics
1	Age > 50 years, household size < 2, rural area, combination benefit discount, residential customer, sim only
2	Age > 50 years, no residential customer, < 300 minutes and < 500 MB
3	City, low income, residential customer, age between 25-50 years
4	Fiber connection, residential customer, household size > 2, age < 50s
5	Low income, 500-1000 MB, age < 35 years
6	Age < 35 years, more than 1000 MB
7	Other

Table 6-7: Segments defined by domain experts.

		Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6	Segment 7	Weighted average	Benchmark
	% churn in test set	2.81%	13.56%	8.58%	6.75%	14.76%	7.72%	9.41%		
LR	AUC churn	0.60	0.56	0.56	0.58	0.65	0.47	0.59	0.58	0.54
	AUC renew	0.75	0.84	0.69	0.73	0.76	0.65	0.74	0.75	0.76
	AUC sleep	0.76	0.82	0.69	0.76	0.76	0.66	0.74	0.75	0.76
RF	AUC churn	0.74	0.77	0.73	0.70	0.72	0.69	0.78	0.74	0.91
	AUC renew	0.87	0.92	0.84	0.86	0.91	0.81	0.89	0.88	0.99
	AUC sleep	0.90	0.91	0.87	0.86	0.91	0.83	0.91	0.89	0.95
XGBoost	AUC churn	0.76	0.81	0.84	0.85	0.85	0.84	0.85	0.83	0.95
	AUC renew	0.94	0.95	0.91	0.91	0.94	0.89	0.93	0.93	0.95
	AUC sleep	0.94	0.92	0.91	0.91	0.92	0.90	0.93	0.92	0.99
Ensemble	AUC churn	0.77	0.82	0.84	0.79	0.84	0.82	0.83	0.82	0.95
	AUC renew	0.95	0.96	0.92	0.92	0.95	0.90	0.93	0.94	0.99
	AUC sleep	0.95	0.93	0.92	0.92	0.93	0.91	0.93	0.93	0.99

Table 6-8: AUC of the predictive models for each of the segments (on the test set).

6-7 Approach: LCA as feature

Supervised learning

This approach uses the cluster labels obtained with LCA that represent the identity of clusters as input to the classifier for the prediction of churn. The results are listed in table 6-9.

The performance of the LCA as feature approach is similar to the performance of the benchmark models. This suggests that the cluster label is an uninformative feature. For XGBoost, the cluster label does not belong to the top 100 most important features indicating that the cluster label is indeed not informative for the prediction of churn.

		LCA as feature	Benchmark
LR	AUC churn	0.54	0.54
	AUC renew	0.76	0.76
	AUC sleep	0.78	0.76
RF	AUC churn	0.86	0.91
	AUC renew	0.94	0.99
	AUC sleep	0.95	0.95
XGBoost	AUC churn	0.96	0.95
	AUC renew	0.99	0.95
	AUC sleep	0.99	0.99
Ensemble	AUC churn	0.94	0.95
	AUC renew	0.99	0.99
	AUC sleep	0.99	0.99

Table 6-9: AUC of the predictive models for the LCA clusters added as feature approach (on the test set).

6-8 Probability threshold

To show how a model can be used to make informative decisions, Figure 6-13 illustrates the performance of RF on the full data set (i.e. the RF benchmark model) as a function of the threshold on the predictive probability to churn. Note that the analysis is performed using pairwise comparison (i.e. churn. vs all other classes). A threshold of 0.4 reflects that all customers with a probability of 0.4 or greater are predicted to churn. For a threshold of 0.4:

1. Approximately 38% of the customers will be queued. If there are 10000 customers scored for churn each day, approximately 3800 will be selected to get a churn reduction call/message.
2. The precision is around 38%, illustrating that 38% of the customers predicted to churn will churn.
3. The recall is around 90%, illustrating that of all customers who actually churn, 90% is predicted to churn.

Business value of best model

Suppose that the churn prevention strategy of the telecom provider is to make direct phone calls to each predicted churner. The assumption is that if a user who was going to churn gets a phone call, he or she will not churn.

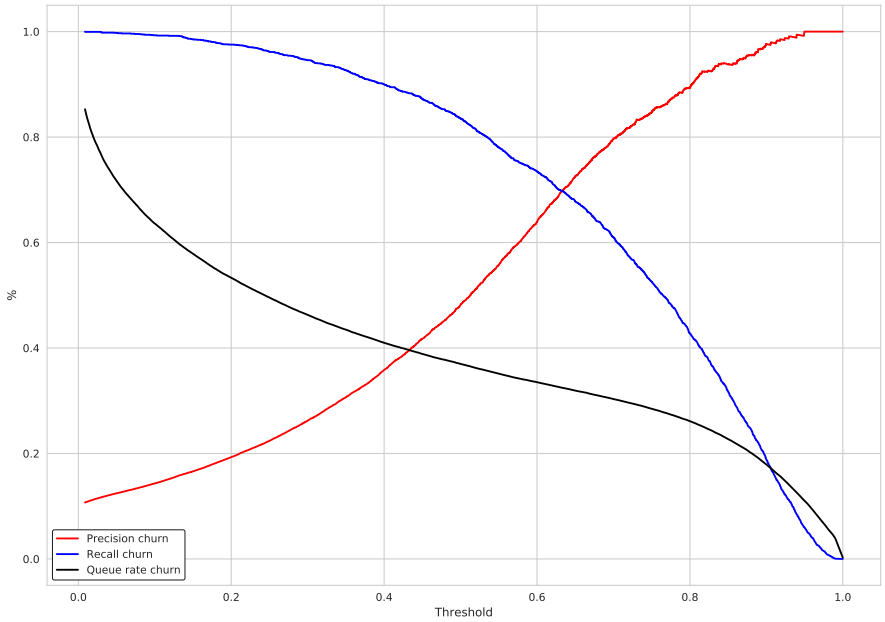
Scenario
The retention team of the telecom provider consists of 8 members. The number of customers that should be scored each period is: 10000. The cost of labor in making a phone call is €20. If a user churns, the telecom provider loses €100. In Table 6-10 all information is listed.

Suppose that each of the retention team members can make 250 calls per period. As a result, the retention team can make 2000 calls per period. This means that the queue rate should be around 2000/10000 = 20%. Combining this information with the top plot of Figure 6-13 the threshold should be around 0.88. The total cost is then around €20 x 2000 =€40000. As the precision is around 92%, the saving the telecom provider will make is around 0.92 x €100 x 2000 = €184.000. This means that the total profits are €184.000 - €40.000 = €144.000. In the bottom plot of Figure 6-13 the payout as a function of the threshold can be found. The optimal threshold for the specific cost estimates from the scenario example is 0.73, which gives the telecom provider an expected profit of about €180.000.

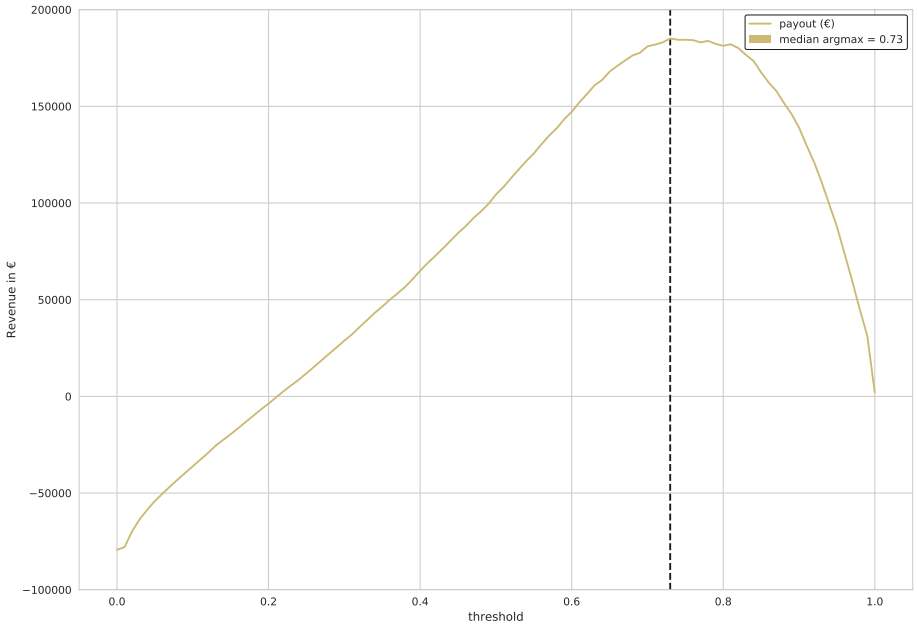
To conclude: the right plot in Figure 6-13 helps the telecom provider to choose the most profit-driven threshold for their particular business application. This analysis can also be performed for other classification techniques than RF. Moreover, when other loss matrices are considered, the expected profits will be different.

Number of customers to predict each period	10000
Retention team members	8
Costs of labor per call	€20
Costs of churn	€100

Table 6-10: Scenario example information.



(a) Performance of RF as a function of the threshold.



(b) Payout as a function of the threshold.

Figure 6-13: Figures used for the threshold analysis.

Chapter 7

Discussion and conclusion

Section 7-1 will focus on the discussion of the results. In Section 7-2 concluding remarks will be given. Moreover, in Section 7-2-1 the limitations of this study will be given as well as suggestions for further research.

7-1 Discussion

To achieve the main goal in this study, a comparison of the different approaches has to be made. In Table 7-1 the Area Under the Curve (AUC) values for the churn class are listed.

Compared to random predictions, where 10% of the customers are predicted to churn (see figure 5-1 for the distribution), all performed analyses in this study result in a higher AUC. This means that, compared to randomly predicting, it is beneficial for the telecom provider to implement one of the approaches from this study.

Regarding Random Forest (RF), XGBoost and the ensemble, there is no added value of combining unsupervised and supervised learning with respect to the accuracy of predicting churn. XGBoost and RF are tree based methods that look for differences between the customers and the artificial homogeneous group of customers. The final leaves of a tree based method can be seen as a cluster with the path to the final leaves being the characteristics of a cluster. With the benchmark approach, XGBoost and RF use the class labels to define the best differentiator in the input variables. As a result, it is understandable that RF and XGBoost are able to make better splits than the unsupervised learning approaches.

For Logistic Regression (LR), the clustering approaches (i.e. t-Distributed Stochastic Neighbor embedding (t-SNE)-Gaussian Mixture Model (GMM), GMM and Latent Class Analysis (LCA)) do not outperform the partitioning of customers by the domain experts. The homogeneous groups defined by domain experts contains less noise as they are identified using explicit definitions. This might explain why, for the not so flexible LR, the partition by domain experts results in a higher AUC. Moreover, for LR, three of the hybrid approaches outperform the benchmark approach. This implies that a LR fitted for each group is able to

model the complex relations in the data set better than that of the LR for the whole data set.

The cluster label itself was found out not to be very informative for the prediction of churn. However, the clusters can be used to determine a churn reduction message. This is outside the scope of this thesis but future studies should use the results from the unsupervised learning stage to identify needs of each of the obtained clusters in order to optimize churn reduction via personalization.

There is no agreement found on the type of unsupervised learning technique that results in the largest AUC for churn. For example, for LR the GMM-small and segmentation achieve the highest AUC. For RF this is the t-SNE-GMM approach. Thus, for the four examined supervised learning techniques, different unsupervised learning techniques result in a larger AUC. This suggests that the choice of the supervised learning technique is more important than the choice of the unsupervised learning technique.

Of the supervised learning techniques, XGBoost and the heterogeneous ensemble outperform LR and RF in terms of the AUC for churn. This is consistent with prior literature as XGBoost and the ensemble are the most advanced predictive models used [28]. XGBoost and the ensemble show similar performance. As the interpretation of an individual model is easier than the interpretation of an ensemble, XGBoost is the preferred model. To conclude, this study suggests that XGBoost is a potentially valuable tool for churn prediction based on the data sets used.

	Benchmark	t-SNE-GMM	GMM-small	GMM-large	LCA	Segmentation	LCA clusters as feature
LR	0.54	0.57	0.58	0.51	0.54	0.58	0.53
RF	0.91	0.77	0.74	0.71	0.74	0.74	0.85
XGBoost	0.95	0.85	0.84	0.82	0.83	0.83	0.96
Ensemble	0.95	0.83	0.83	0.82	0.83	0.82	0.94

Table 7-1: AUC values of the classifiers for each of the approaches.

Managerial implications

From a practical perspective, this study indicates several instruments for the telecom provider to identify churners, renewers and sleepers. First of all, a combination of features identifies to which outcome class a customer belongs. Churners can be segregated from renewers and sleepers based on the number of days before the end of the contract and the size of the minutes and data bundle. To be more specific, churners have a smaller bundle and decide to terminate their contract when the contract end date is expired. Combining this with the number of views for the shop and terminate contract page, churners can be distinguished from renewers.

The characteristics of a homogeneous group that demonstrate a large rate of churn can be used to understand churners. For example, the rate of churn in segment 2 is larger compared to the other segments. Segment 2 consists of customers that are older than 50, do not have residential products and have less than 300 minutes and 500 MB. The exploratory analysis showed that churners view, on average, more terminating contract pages in the 29 days before the action (with a median value of 2) than the other classes. Moreover, the total number of

pages viewed regarding the account divided by the total number of pages viewed is on average larger for churners (with a median of 0.62). A suggestion for the telecom provider is therefore to keep an eye on customers with the characteristics of segment 2 that view the terminating contract page for the second time and 62% of their total views is is on the account page in less than 29 days.

Lastly, from a practical side, the threshold probability analysis showed how the best predictive model can be used to make decisions that improve the return on investment.

7-2 Conclusion

The main goal of this study is to evaluate the added value of combining unsupervised and supervised learning with respect to the accuracy of predicting churn. First, groups of customers who share the same characteristics are identified using different unsupervised learning techniques. Secondly, for each homogeneous group is predicted which customers are most likely to churn using different supervised learning techniques. Lastly, several combinations of unsupervised and supervised learning techniques are evaluated by comparing the accuracy of predicting churn.

The results revealed that for the flexible models (i.e. RF, XGBoost and the ensemble), the hybrid approach does not have an added value. However, the AUC of churn increases when a less flexible model (i.e. LR) is used. Moreover, the different (combinations) of techniques show similar predictive performance which suggest that the general set-up is more important than the choice of a model.

If the objective is to achieve the highest accuracy of predicting churn, the results of this study reveal that a non-hybrid approach should be used. To be more specific, XGBoost is the recommended method (resulting in an AUC of 0.95). Further, the characteristics of homogeneous groups with a high rate of churn helps to understand the characteristics of churners. This implies that a hybrid approach might be valuable when the goal is to understand churn.

7-2-1 Limitations and further research

The proposed approach in this research leaves room for improvement in terms of the data set used, the models, and the general framework.

There are limitations regarding the data set. In this research, the models are able to predict 3 days before a customer undertakes an action if he is going to churn. However, in practice it is impossible to know when the 3 days before the action are and thus the model might not be able to accurately predict which customers are likely to churn. A suggestion is to implement this model in practice and monitor if the predicted churners are actually going to churn. If this is not the case, extending the 3 day communication period could be informative to make the predictions less dependent on the communication period.

An enhanced feature set is necessary to increase the reliability of the predictions and more accurate results will be achieved when additional data sources are used. It would be informative to add features about bundle usages and invoices. Moreover, there is no information

about prices from competitors or releases of new phones. In addition, it is necessary to include additional touch points with the customer (e.g. call center data). Lastly, there is no information loop in this study. For example, if during the communication period a customer contacts the telecom provider, this information is not included although this could be essential to accurately predict the action type.

This study used the engineered features from the online data set for prediction. As a result, the sequential information from online behavior is partly lost. There are models available which can model sequential data (i.e. LSTM models) and a suggestion would be to use these type of models to extract even more information.

Another limitation is how the heterogeneous ensemble is implemented in this study. Successful ensembles consists of non-correlated models [35]. However, this study uses a RF and XGBoost in the ensemble, which are likely to be correlated. Adding, for example, a Neural Network to the heterogeneous ensemble might result in a better model. Moreover, in this study the weights of the heterogeneous ensemble are determined by the best-worst weighted voting system. However, it would be interesting to add a more sophisticated weighting system.

Lastly, this study uses the combination of unsupervised and supervised learning to achieve a high predictive performance. Moreover, the focus of this study is on the first step in the churn management framework: predict which customers are most likely to churn. However, the ultimate objective is not to predict which customers are most likely to churn but to actually reduce the churn rate. It would be interesting for further research to use the results from the unsupervised learning stage to identify needs of each of the obtained clusters in order to send the right churn reduction message. This methodology focuses on the whole churn management framework: both predicting which customers are most likely to churn as well as targeting the predicted churners with the right message to induce them to stay.

Appendices

Appendix A

t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets [7]. t-SNE consists of two stages. During the first stage, a probability distribution is constructed such that similar objects have a high probability of being picked. In the second phase, t-SNE defines a probability distribution over all points in a low dimensional map.

Given a set of N high-dimensional objects $x_i, ...x_N$, t-SNE, first computes the probabilities p_{ij} that are proportional to the similarity of objects x_i and x_j. As a result, if the distance between x_i and x_j is small, the conditional probability should be high. k represents the number of nearest neighbors. The conditional probability is defined by:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq 1} \exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)} \tag{A-1}$$

The conditional probabilities depend on the distances between data points as well as the perplexity. Perplexity is a hyper-parameter which can be set by the user and can be seen as a smooth measure of the effective number of neighbors. Perplexity can be used to balance the attention between local and global aspects of the data set. The σ_i in the conditional probability function, is the variance of a Gaussian distribution centered on x_i. The σ_i is defined in terms of the perplexity. The perplexity of the distribution is:

$$\text{Perplexity}(P_i) = 2^{H(P_i)} \tag{A-2}$$

$$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i} \tag{A-3}$$

where $H(P_i)$ is the Shannon entropy of P_i.

Once all pairwise conditional probabilities are computed, the conditional probabilities will be made symmetric using:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \tag{A-4}$$

The next step is for the low-dimensional y_i and y_j of the high-dimensional data points x_i and x_j to compute the similar conditional probability. This conditional probability for low-dimensional data $q_{j|i}$ is given by:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq 1} \exp\left(-\|y_i - y_j\|^2\right)} \tag{A-5}$$

If the points produced by the low-dimensional map represent the proximity between the data points in the high dimensional map, then the conditional probabilities $p_{j|i}$ and $q_{j|i}$ are equal.

The goal of t-SNE is to minimize the Kullback-Leibler divergence. The cost function C is defined as:

$$C = KL(P|Q) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{A-6}$$

The cost function in the t-SNE algorithm uses a Student-t distribution with one degree of freedom. To minimize the cost function C gradient descent is applied. The gradient of the cost function is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_{j \neq 1} (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{A-7}$$

Appendix B

Gaussian Mixture Modelling

A Gaussian mixture model is a probabilistic model that assumes all data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Gaussian Mixture Model (GMM) allows for uncertainty in cluster membership as it assigns a probability for each cluster. Therefore, GMM is particularly useful when true clusters overlap and when the data is spread out.

There are several definitions of GMM available. According to [56] a GMM is a weighted sum of M component Gaussian densities as given by the equation:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i) \tag{B-1}$$

where x is a D-dimensional continuous-valued data vector, w_i with $i = 1, ..., M$ are the mixture weights, and $g(x|\mu_i, \Sigma i)$ with $i = 1, ..., M$ are the component Gaussian densities with mean vector μ_i and covariance matrix Σ_i. Each component density is a D-variate Gaussian function of the form:

$$g(x|\mu_i, \Sigma i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)' \Sigma_i^{-1} (x-\mu_i)} \tag{B-2}$$

with mean vector μ_i and covariance matrix Σ_i. The covariance matrix, Σ_i, can can be full rank or constrained to be diagonal. Additionally, parameters can be shared, or tied, among the Gaussian components, such as having a common covariance matrix for all components. In this study, each component has its own general covariance matrix (i.e. covariance type full). The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$.

To learn the parameters $\lambda = \{w_i, \mu_i, \Sigma_i\}$ with $i = 1, ..., M$ that best matches the distribution of the feature vectors the famous Expectation-Maximization Algorithm (EM) is used [65]. The EM algorithm is a technique for maximum likelihood estimation which guarantees that direct maximization of the likelihood is possible. The EM algorithm alternates between the

steps of guessing a probability distribution over completions of missing data given the current model (i.e. the E-step) and then re-estimating the model parameters using these completions (i.e. the M-step) [65].

Convergence is achieved when the value of the log-likelihood does not change from one iteration to the next. The EM algorithm is an ascent method for maximizing the likelihood, but is only guaranteed to converge to a stationary point of the likelihood function [66]. As a result, GMM can be stuck in a local maxima and different initial values should be used.

Using Bayes' theorem and the estimated model parameters, the posteriori component assignment probability can be estimated. This property makes the GMM useful for clustering. It gives the possibility to identify for each observation if is likely to be from one component distribution versus another. Cluster assignment is determined by the component with the highest probability for the observation.

Appendix C

Latent Class Analysis

Latent Class Analysis (LCA) is a technique which can be used to identify and characterize clusters of similar cases when dealing with multivariate categorical data. LCA seeks to stratify the cross-classification table of observed (i.e. manifest) variables by an unobserved (i.e. latent) unordered categorical variable that eliminates all confounding between the manifest variables [9].

According to [9] the definition of LCA is as follows. Suppose J polytomous categorical variables are observed, each of which contains K_j possible outcomes, for individuals $i = 1..N$. Y_{ijk} is the observed values of the J manifest variables such that $Y_{ijk} = 1$ if respondent i gives the kth response to the jth variable and $Y_{ijk} = 0$ otherwise, where $j = 1,...,J$ and $k = 1,...,K_j$.

LCA approximates the observed joint distribution of the manifest variables as the weighted sum of a finite number, R, of constituent cross-classification tables. Let π_{jrk} denote the class-conditional probability that an observation in class $r = 1, ..., R$ produces the kth outcome on the jth variable. Within each class, for each manifest variable $\sum_{k=1}^{K_j} \pi_{jrk} = 1$. The probability that an individual i in class r produces a particular set of J outcomes on the manifest variables, assuming local independence, is the product

$$f(Y_i; \pi_r) = \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}} \tag{C-1}$$

The probability density function across all classes is the weighted sum

$$Pr(Y_i|\pi, p) = \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}} \tag{C-2}$$

The parameter estimated by LCA are p_r and π_{jrk}.

Given estimates $\hat{p_r}$ and $\hat{\pi}_{jrk}$ of p_r and π_{jrk}, the posterior probability that each individual belongs to each class, conditional on the observed values of the manifest variables, can be calculated using Bayes' formula:

$$\hat{P}(r_i|Y_i) = \frac{\hat{p_r} f(Y_i; \hat{\pi}_r)}{\sum_{q=1}^{R} \hat{p_q} f(Y_i; \hat{\pi}_q)} \tag{C-3}$$

LCA can be estimated by maximizing the log-likelihood function:

$$lnL = \sum_{i=1}^{N} ln \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}} \tag{C-4}$$

with respect to p_r a π_{jrk} using the expectation-maximization (EM) algorithm. This log-likelihood is identical in form to the standard finite mixture model log-likelihood [9].

Appendix D

Logistic Regression

Logistic Regression (LR) is a popular model to classify observations. It is based on the logistic function [58]. The logistic function is an S-shaped curve that can take any real input x and map it into a value between 0 and 1. As a result, it can be interpret as a probability which can be used to assign classes to observations.

LR uses input values x and combines them using weights to predict an output value y. In the case of one predictor x, the logistic regression equation becomes:

$$y = \frac{\exp^{\beta_0+\beta_1 \cdot x}}{1+\exp^{\beta_0+\beta_1 \cdot x}} \tag{D-1}$$

where y is the predicted output, β_0 is the bias and β_1 is the single input value x.

The LR models the probability that an input X belongs to the default class $Y = 0$. The regression coefficients β are estimated using maximum likelihood estimation [5]. Using equation (D-1) the probability of $Y = 0$ and $Y = 1$ can be calculated as follows:

$$Pr(Y_i = 0) = \frac{\exp^{\beta_0+\beta_1 \cdot x}}{1+\exp^{\beta_0+\beta_1 \cdot x}} \tag{D-2}$$

$$Pr(Y_i = 1) = 1 - Pr(Y_i = 0) = \frac{1}{1+\exp^{-\beta_0+\beta_1 \cdot x}} \tag{D-3}$$

The predicted probabilities will be transformed to the class label in order to actually make a probability prediction. The threshold value determines which probability assigns an observation to one of the classes.

In this study, there are n features. As a result, the expression $\beta_0 + \beta_1 x_1$ can be rewritten in $\beta_0 + \beta_1 x_1 + ... + \beta_n x_n$.

Moreover, multinomial classification is performed where each observation can be classified into one out of three classes. Therefore standard LR (with a binary output) should be

converted into a multinomial problem. This is done by applying softmax regression which is a generalization of LR that can be used for multi-class classification. With softmax regression, the sigmoid logistic function is replaced by the softmax function:

$$P(y = j|t^{(i)}) = \sigma_{softmax}(t^i) = \frac{e^{t(i)}}{\sum_{j=1}^{k} e^{t_k^{(i)}}} \tag{D-4}$$

where the net input t is defined as:

$$t = w_0x_0 + w_1x_1 + ... + w_nx_n = \sum_{l=0}^{n} w_1x_1 = w^T x \tag{D-5}$$

with w the weight vector, x the feature vector of one training sample and w_0 the bias unit. The softmax function computes the probability that an observation $x(i)$ belongs to class j given the weight and net input $t(i)$. This probability is calculated for each class label $j = 1..k$. The cost function J that is minimized is defined as:

$$J(W) = \frac{1}{m} \sum_{i=1}^{m} H(T_i, O_i) \tag{D-6}$$

which is the average of all cross entropies over m observations. The cross-entropy is defined as:

$$H(T_i, O_i) = -T_i \cdot \log(O_i) \tag{D-7}$$

with T the true class label and O the computed probability for each of the classes. The cost function can be minimized using gradient descent which calculates the derivative of the cost function:

$$\nabla w_j J(W) \tag{D-8}$$

which can be used to update the weights in opposite direction of the gradient:

$$w_j : w_j - \alpha \nabla w_j J(W) \tag{D-9}$$

$$j \in 0, ..., n$$

with the stepsize $\alpha > 0$.

The final cost gradient is defined as:

$$\nabla w_j J(W) = -\frac{1}{m} \sum_{i=0}^{m} [x^{(i)}(T_i - O_i)] \tag{D-10}$$

which can be updated iteratively until predefined threshold is reached.

The multi class predicted probabilities will be transformed to the class label which has the highest probability.

Appendix E

Random Forest

Random Forest (RF) is based on the concept of a decision tree. Decision trees are a high variance model as they have the risk of being tuned to the training set. The idea in RF is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much [59]. RF is a substantial modification of the bagging techniques that builds a large collection of de-correlated trees and then averages them [5].

Given training observations x_i and a class label y_i, a decision tree tries to partition the space such that the observations from the same class are grouped together. Let data at node m be represented by Q. For each candidate split $\theta = (j, t_m)$ consisting of a feature j and threshold t_m, partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets.

$$Qleft(\theta) = (x, y)|x_j <= t_m \tag{E-1}$$

$$Qright(\theta) = Q \backslash Q_{left}(\theta) \tag{E-2}$$

The impurity at m is computed using an impurity function (H). The choice of the impurity function makes how a decision tree decides to split a node in two or more sub-nodes. For classification tasks, the impurity function can be the Gini index and entropy. The Gini index favors larger partitions while the entropy favors smaller partitions with many distinct values. The choose of the impurity function depends on the use case and can be estimated by randomized search cross validation.

RF is a bagging technique for reducing the variance of a decision tree. Bagging is used to reduce the variance while retaining the bias. Decision trees are a high variance model as a decision tree has the risk of being tuned to the training set. In general, decision tree do not generalize very well. With RF, a committee of trees each cast a vote for the predicted class. The idea in RF is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much [5]. The algorithm of the RF according to [5] is listed in algorithm 1.

To make a prediction for an observation x:

$\hat{C}_b(x)$ = the class prediction of the bth RF tree.
Then $\hat{C}_{rf}^B(x)$ = majority vote $\hat{C}_b(x)_1^B$

1. **for** *b=1 to B:* **do**
   (a) Draw a boostrap sample Z* of size N from the training data
   (b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree until the minimum node size n_min is reached
   1. Select m variables at random from the p variables
   2. Pick the best variable/split-point among the m
   3. Split the node into two daughter nodes

end

2. Output the ensemble of trees Tb_1^B

Algorithm 1: Random Forest

Appendix F

XGBoost

XGBoost is a modification of the gradient boosting algorithm, which in addition to exploiting parallelized computing also applies a separate form of the procedure. Formally, described by [10], the objective of the XGBoost model is:

$$Obj(\theta) = L(\theta) + \delta(\theta) \tag{F-1}$$

where L is the training loss function and δ is the regularization term. The training loss measures how good the model is in predicting the training data. In the case of a multi-class classification task L is the multiclass log loss. The regularization component controls overfitting of the model, by punishing too big values of the model parameters.

According to [10] the output of a tree ensemble is given by:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{F-2}$$

where f_k is decision tree k where $k \in K$ trees. The objective function can thus be redefined to:

$$L_t(y_i, \hat{y_i}) = \sum_{i=1}^{N} l(y_i, \hat{y_i}) + \sum_{k} \delta(f_k) \tag{F-3}$$

$$\delta(f) = \gamma L + \frac{1}{2}\lambda ||w||^2 \tag{F-4}$$

where L is the number of leaves, K the number of trees fitted, w the leaf weight and both λ and γ the regularization constants.

XGBoost uses gradients to identify the shortcomings of the existing learners which mean that the individual trees are fitted against the gradients $\hat{y_{i,t}}$. At iteration i the loss function l can be approximated by a function f_t :

$$L_t(y_i, \hat{y}_i) = \sum_{i=1}^{N} l(y_i, \hat{y}_{i,(t-1)}) + f_t(x_i)) + \delta(f_t) \tag{F-5}$$

With the loss function l the multi class log loss. Given a predefined tree structure XGBoost takes a second order approximation of the loss function l, and derives the optimal weights w.

Bibliography

[1] Kiansing Ng and Huan Liu. Customer retention via data mining. *Artificial Intelligence Review*, pages 569–590, 2000.

[2] A. Berson, S. Smith, and K. Thearling. Building data mining applications for CRM. *New York, NY: McGraw-Hill.*, 2000.

[3] L Yan, R Wolniewicz, and R Dodier. Predicting customer behavior in telecommunications. *IEEE Intelligent Systems*, pages 50–58, 2004.

[4] C Wei and I Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications*, 23(10):103–112, 2002.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning Theory*. Springer - Verlag, 2009.

[6] Yaswanth Kumur Alapati. Combining clustering with classification: a technique to improve classification accuracy. *International journal of computer science engineering*, 5(6):2319–7323, 2016.

[7] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(2579-2605), 2008.

[8] Michael Fop and Thomas Brendan Murphy. Variable selection methods for model-based clustering. *eprint arXiv:1707.00306*, 2017.

[9] Drew A. Linzer and Jeffrey Lewis. poLCA: Polytomous variable Latent Class Analysis. *Journal of statistical software*, 42(10), 2011.

[10] T. Chen and C. Guestrin. XGBoost: a scalable tree boosting system. *arXiv:1603.02754v3*, 2016.

[11] Chich-Fong Tsai and Yu-Hsin Lu. Data mining techniques in customer churn prediction. *Recent patents on computer science*, 3(28-32), 2010.

[12] Jeffrey David Ullman Jure Leskovec, Anand Rajaraman. *Mining of Massive Datasets*. Cambridge University Press New York, NY, USA, 2014.

[13] C. Shearer. The CRISP-DM model: the new blueprint for data mining. *J Data Warehousing*, 5(13-22), 2000.

[14] Y. Fu. Data mining: tasks, techniques and applications. *IEEE Potentials*, (18-20), 1997.

[15] David J. Hand. Principles of data mining. In *Drug safety*, volume 30, pages 621–622, 2007.

[16] Vishal Mahajan, Dr. Richa Misra, and Dr. Renuka Mahajan. Review of data mining techniques for churn prediction in telecom. *JIOS*, 37(2):183–197, 2015.

[17] S Hung, D Yen, and H Wang. Applying data mining to telecom churn manageement. *Expert Systems with Applications*, 31(3):515–524, 2006.

[18] Guangli Nie, Wei Rowe, Lingling Zhang, Yingjie Tian, and Yong Shi. Credit card churn forecasting by Logistic Regression and Decision Tree,. *Expert Systems with Applications*, Volume 38(Issue 12), 2011,.

[19] Marcin Owczarczuk. Churn models for prepaid customers in the cellular telecommunication industry using large data marts. *Expert Systems with Applications*, 37(6):4710–4712, 2010.

[20] Niccolò Gordini and Valerio Veglio. Customers churn prediction and marketing retention strategies. an application of support vector machines based on the AUC parameter-selection technique in B2B e-commerce industry. *Industrial Marketing Management*, Volume 62:Pages 100–107, 2017.

[21] Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren. Customer churn prediction using improved one-class Support Vector Machine. In Xue Li, Shuliang Wang, and Zhao Yang Dong, editors, *Advanced Data Mining and Applications*, pages 300–306, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[22] Guo en XIA and Wei dong JIN. Model of customer churn prediction on Support Vector Machine. *Systems engineering theory and practice*, 28(1):71–77, 2008.

[23] Yu-Hsin Lu Chih-Fong Tsai. Customer churn prediction by hybrid Neural Networks,. *Expert Systems with Applications,*, Volume 36(Issue 10,):Pages 12547–12553,, 2009,.

[24] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, and K.Ch. Chatzisavvas. A comparison of machine learning techniques for customer churn prediction. *Simulation modelling practice and theory*, 55:1–9, 2015.

[25] Mohammed Hassouna, Ali Tarhini, Tariq Elyas, and Mohammad Saeed AbouTrab. Customer churn in mobile markets: A comparison of techniques. *International Business Research;*, 8(6), 2015.

[26] Scott A. Neslin, Sunil Gupta, Wagner Kamakura, Junxiang Lu, and Charlotte H. Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models,. *Journal of Marketing Research,*, 43,(2,):204–211, 2006,.

[27] Rich Curuana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *ICML '06 Proceedings of the 23rd international conference on Machine learning*, 161-169, 2006.

[28] Kristof Coussement and Koen de Bock. Customer churn prediction in the online gambling industry: the beneficial effect of ensemble learning. *Journal of business research*, 66(9):1629–1636, 2013.

[29] YongSeog Kim. Toward a succesful CRM: variable selection, sampling and ensemble. *Decision support systems*, 41:542–553, 2006.

[30] L.I. Kuncheva. Combining pattern classifiers: Methods and algorithms. *John Wiley and Sons, Hoboken, New Jersey*, 2004.

[31] B. Lariviere and D. van den Poel. Predicting customer retention and profitability by using Random Forests and regression forests techniques. *Expert Systems with Applications*, 29(2):472–284, 2005.

[32] Shun Bian and Wenjia Wang. On diversity and accuracy of homogeneous and heterogeneous ensembles. *International Journal of Hybrid Intelligent Systems*, 4(2):103–128, 2007.

[33] Aurélie Lemmens and Christophe Croux. Bagging and boosting classification trees to predict churn,. *Journal of Marketing Research,*, 43,(2,):276–286,, 2006,.

[34] Yaya Xie, Xiu Li, E.W.T. Ngai, and Weiyun Ying. Customer churn prediction using improved balanced Random Forest. *Expert Systems with Applications*, 36(3):5445–5449, 2009.

[35] García V. Marqués, A. and J. S. Sánchez. Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications*, 39(11):10244–10250., 2012.

[36] M. Braun and D.A. Schweidel. Modeling customer lifetimes with multiple causes of churn. *Marketing science*, 30(5):881–902, 2011.

[37] Dirk van den Poel and Wouter Bunckinx. Predicting online-purchasing behaviour. *European journal of operational research*, 166(2):557–575, 2005.

[38] Ricardo Filipe Fernandes and Costa Magalhães Teixeira. Using clickstream data to analzye online purchase intentions. Master's thesis, Universidade do porto, 2015.

[39] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. Data preprocessing for supervised learning. *International journal of computer science*, 1(1):1306–4428, 2006.

[40] Indranil Bose and Xi Chen. Hybrid models using unsupervised clustering for prediction of customer churn. In *IMECS*, 2009.

[41] Amjad Hudaib, Reham Dannoun, Osama Harfoushi, Ruba Obiedat, and Hossam Faris. Hybrid data mining models for predicting customer churn. *Int. J. Communications, Network and System Sciences*, 8:91–96, 2015.

[42] G. Punj and D.W. Stewart. Cluster analysis in marketing research: review and suggestions for application. *Journal of marketing research*, 20(2):134–148, 1983.

[43] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys*, 31(3):264–323, 1999.

[44] Fred van Raaij and T.M.M. Verhallen. Domain-specific market segmentation. *Proceedings IAREP/SASE Conference on Interdisciplinary Approaches to the Study of Economic Problems, Stockholm*, 1991.

[45] Peter R. Dickson and James L. Ginter. Market segmentation, product differentiation and marketing strategy. *American marketing association*, 51(2):1–10, 1987.

[46] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 2012.

[47] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(13):281–305, 2012.

[48] Rushi Longadge, Snehlata S. Dongre, and Latesh Malik. Class imbalance problem in data mining: review. *International journal of computer science and network*, 2(1):2277–5420, 2013.

[49] Gary M. Weiss and Foster Provost. The effect of class distribution on classifier learning: an empirical study. Technical report, Rutgers university, 2001.

[50] E. Ramentol, N. Verbiest, Y. Caballero, and C. Cornelis. SMOTE-FRST: A new resampling method using fuzzy rough set theory. Technical report, WSPC, 2012.

[51] N. Japkowicz. The class imbalance problem: Significance and strategies. *Proceedings IAREP SASE Conference on Interdisciplinary Approaches to the Study of Economic Problems, Stockholm*, 2000.

[52] J. Burez and D. van den Poel. Handling class imbalance in churn prediction problem. *Expert Systems with Applications*, 36(4626-2636), 2009.

[53] Dmitry Ulyanov. Multicore-t-SNE. https://github.com/DmitryUlyanov/Multicore-TSNE, 2016.

[54] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. *Distill*, 2016.

[55] Justina žurauskienė and Christopher Yau. pcaReduce: hierarchical clustering of single cell transcriptional profiles pcaReduce: hierarchical clustering of single cell transcriptional profiles pcaReduce. *BMC Bioinformatics*, 7(140), 2016.

[56] G. McLachlan. *Mixture Models, Year = 1998*. Marcel Dekker New York.

[57] Dominique Haughton, Pascal Legrand, and Sam Woolford. Review of three latent class cluster analysis packages: Latent gold, poLCA and MCLUST. *The american statistician*, 63(81-91), 2009.

[58] DW Hosmer and S Lemeshow. *Applied Logistic Regression*. Wiley New York, 2 edition, 2000.

[59] L Breiman. Random Forests. *Machine Learning*, 45:123–140, 2001.

[60] F. Moreno-Seco, J. M. Iñesta, P. J. P. de León, and L. Micó. Comparison of classifier fusion methods for classification in pattern recognition tasks,. *Proceedings of Joint IAPR International Conference on Structural, Syntactic, and Statistical Pattern Recognition*, pages 705–713, 2006.

[61] CBS. Data neighbourhoods. www.cbsinuwbuurt.nl, 2016.

[62] Statistics Netherlands. Gemeentegrootte en stedelijkheid. *https://www.cbs.nl/nl-nl/onze-diensten/methoden/classificaties/overig/gemeentegrootte-en-stedelijkheid*, 2018.

[63] R. Cooley, B. Mobasher, and J. srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[64] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.

[65] Chuong B Do and Serafim Batzoglou. What is the Expectation Maximization algorithm? *Nature biotechnology*, 26, 2008.

[66] Chi Jin, Yuchen Zhang, and Sivaraman Balakrishnan. Local maxima in the likelihood of the Gaussian Mixture Models: structural results and algorithmic consequences. In *29th Conference on Neural Information Processing Systems Barcelona*, 2016.

Glossary

List of Acronyms

LCA	Latent Class Analysis
GMM	Gaussian Mixture Model
LR	Logistic Regression
EM	Expectation-Maximization Algorithm
RF	Random Forest
t-SNE	t-Distributed Stochastic Neighbor embedding
churners	customers who terminate their contract
sleepers	customers who do not take any action regarding their contract
renewers	customers who retain their contract
ROC	Receiver Operating Curve
AUC	Area Under the Curve
CRISP-DM	Cross-Industry Standard Process For Data Mining
FPR	False Positive Rate
TPR	True Positive Rate