Erasmus School of Economics

Churn prediction: a comparative study on the influence of using balancing techniques on the predictive performance of churn prediction models

25-07-2022

Anne van der Kleij (465765)

MSc Economics and Business

Track: Data Science & Marketing Analytics

Supervisor: dr. K. Gruber

Second Assessor: dr. N.M. Almeida Camacho

Abstract

Churn and customer retention have become increasingly important for B2B companies, as retaining a current customer has shown to have a higher return on investment than acquiring a new customer. This research uses cross-sectional data, provided by a subscription-based B2B company, to predict churn. The dataset contains 3799 observations and 15 input variables, over a time frame of 3 years. Generally, cross-sectional churn data is known to lead to a class imbalance problem. Ignoring the class imbalance in the data might lead to misleading results of the churn prediction models. Therefore, this research investigates the added value of using balancing techniques in the data pre-processing stage. Three balancing techniques are evaluated in this research, i.e. Random Over-sampling, SMOTE, and ADASYN. Furthermore, this research evaluates the performance of three modelling approaches: Decision Trees, Random Forests, and Support Vector Machines. For each modelling approach, a baseline model is built using the imbalanced training data. Additionally, each balanced dataset is used as input data for all three modelling approaches. Hence, in total, this research evaluates the performance of 12 churn prediction models.

The performance of each of the models is evaluated based on an out of sample test set, using the accuracy, recall, Matthews Correlation Coefficient (MCC), and the Area Under the ROC curve (AUC). Additionally, the geometric mean of the predictions is reported. The results show that no increased out of sample performance is found when applying balancing techniques the training data. More specifically, this research showed that a Random Forest based on the imbalanced input data yields to the highest out of sample performance in terms of accuracy, recall, MCC, and AUC. However, the geometric mean indicates a higher performance for Decision Trees based on imbalanced data, considering both the interpretability and the variation in the predictions.

Keywords: churn, class imbalance, balancing, SMOTE, Random Over-sampling, ADASYN.

Contents

List of Tables

List of Figures

1 Introduction

In the field of marketing, a commonly known goal is the acquisition of new customers. However, it has been shown that acquiring a new customer is about 5 times more costly than retaining existing customers (Gordini & Veglio, 2017; Neslin et al., 2006). This indicates that assigning resources to customer retention has a higher return on investment than merely assigning resources to acquire new customers (Ahmed & Linen, 2017). Additionally, previous research showed that decreasing customer churn by 5% can increase the company's profit by over 25% (Reichheld, 2001). For companies, it is thus not only important to assign marketing resources to the acquisition of new customers, but it is also important to assign resources in order to retain current customers. Anticipating and understanding when customers are likely to churn can be helpful for marketeers to induce new marketing strategies that aim to retain those customers.

The use of Customer Relationship Management (CRM) is one way to build, strengthen and maintain relationships with customers in the long term. CRM can be seen as using business intelligence to make customer acquisition and retention a comprehensive process to help maximize the customer value to a business. This helps companies to dive into the factors that increase the likelihood of churn, which enables them to figure out how to improve their customer loyalty and thus decrease churn (Vafeiadis et al., 2015). CRM can be used for two purposes: operational and analytical CRM. Operational CRM is used to automate business processes, while analytical CRM is used to analyze customer characteristics and behavior to support a business's customer management strategies (Ngai et al., 2009). This research focuses on analytical CRM, which helps businesses allocate their customer retention resources to the customer group with the highest likelihood to churn.

Another way to tackle customer churn is by identifying factors contributing to the likelihood that a customer churns. Past research used various machine learning techniques to predict churn in multiple industries. Coussement & Van den Poel (2008) used different machine learning models to predict churn in the field of newspaper subscriptions. They found that Support Vector Machines (SVM) outperform logistic regression when the optimal parameter settings for SVM are used. Moreover, they found that Random Forests generally outperform SVM.

Coussement et al. (2017) also investigated churn predictions in the telecommunication industry. They compared data preparation techniques to increase the predictive performance of the often-used logistic regression model. They found that a logistic regression is competitive with more advanced ensemble algorithms when using enhanced data preparation techniques. Additionally, Vafeiadis et al. (2015) compared the five most used classification methods to predict churn in the telecommunication sector. They found that artificial neural networks and Decision Tree classifiers performed best for predicting churn in the telecommunication sector. However, there is no agreement on which model has the best predictive performance in the context of churn predictions. For example, Coussement et al. (2017) found that a logistic regression is a competitive model, while Vafeiadis et al. (2015) concluded that neural networks and Decision Trees are the most suitable methods for churn prediction.

Besides research on the best methods to predict churn, it has also been stated that when making a cross-sectional comparison, churn is often a rare event, which induces an imbalanced distribution of the target variable (Chawla, 2009). Imbalanced target variables make the prediction of the rare event less straightforward because the accuracy and the generalizability of the prediction models are limited (Gordini & Veglio, 2017; Verbeke et al., 2011, 2012). Hence, the use of modelling approaches without taking class imbalance into account might give bad results in terms of predictive performance (Ahmed & Linen, 2017). Neslin et al. (2006) found significant differences in the performance when using different data pre-processing steps.

With regards to handling the imbalanced target variable, Burez & Van den Poel (2009) stated that they did not find one class distribution that performs best when random observations are deleted from the majority class (under-sampling). The optimal distribution depends on both the method and the data. Amin et al. (2016) compared the performance of different techniques that randomly add observations in the minority class (over-sampling) in the context of churn prediction. They compared six techniques using four rule generation algorithms and found the best predictive performance of the mega-trend diffusion function as a balancing technique. Additionally, Coussement et al. (2017) confirmed the effect of data preparation techniques on churn prediction performance of various models. The choice of the data preparation technique influences the performance of churn prediction models significantly, which makes the simple logistic regression model competitive with more advanced machine learning algorithms when using proper data preparation techniques (Coussement et al., 2017).

Based on past research, it can thus be concluded that the data preparation technique to handle class imbalance influences the predictive performance. Additionally, using different modelling approaches influences the predictive power of the churn prediction models. Churn prediction research has mainly been performed in the telecommunication sector and in other B2C businesses. This research extends the current churn literature by investigating the added value of using advanced balancing techniques in combination with churn prediction models, to predict churn in a B2B context. This summarizes into the following research question:

(1) What is the value added of using advanced modelling approaches for churn prediction and (2) how does the use of balancing techniques influence the performance?

To answer the research question, this research addresses two parts of the research question. Firstly, balancing techniques are applied to the training dataset. These balanced datasets are then used to predict churn using a Decision Tree. The same balanced dataset is used to predict churn using more advanced machine learning methods, to determine whether advanced balancing techniques influence the performance of machine learning techniques as a churn prediction model.

This research is conducted in collaboration with TOPdesk Nederland B.V (TOPdesk). This is a Business-to-Business company that provides a servicemanagement tool to improve their customers' IT services and processes. An IT service management (ITSM) tool is defined as "*an approach to IT operations, that is characterized by its emphasis on IT services, customers, sevice level agreements, and an IT function's handling of its daily activities through processes*" (Iden & Eikebrokk, 2013, p. 1). TOPdesk's customers have a license for their ITSM tool, and this license is subscription-based. This indicates that in the context of TOPdesk, customer churn is defined as the termination of the subscription. Within TOPdesk, customer retention is a topic that has recently gotten more attention. Past research also showed that subscription-based companies shifted away from merely using marketing as a tool to acquire new customers. The relevance and profitability of retaining customers, by identifying factors that contribute to a high likelihood of churning, was already confirmed by Coussement & Van den Poel (2008). Customer churn relates to customer retention, because insights into the factors that contribute to customer churn can help assign marketing resources to this goal. For example, Ascarza & Hardie (2013) showed that modelling churn can help marketing departments segment their customers based on their likelihood of churning.

TOPdesk uses CRM to maintain and build relationships with their customers. Moreover, they collect data from their own application and their financial administration. However, a modelling approach to predict churn based on this data is not yet used in this company. Therefore, this research is relevant to TOPdesk and other B2B subscription-based companies, because it aims at

investigating which combination of a balancing technique and a modelling approach has the best predictive performance in churn predictions. TOPdesk provided a dataset that originates from multiple sources. The data is collected from TOPdesk's own application, from its' CRM system and from their financial administration. The data from these different sources are connected using anonymous customer ID's, to protect the privacy of TOPdesk's customers. All the data used in this research is existing historical data, which means that this research is conducted in a natural environment, where the data is retrieved from a real business setting, without a manipulation and control group.

The remainder of this research is structured as follows. This research starts with theoretical background of churn and customer retention, accompanied by research on suitable machine learning techniques for churn prediction modelling. The theoretical background also covers balancing techniques suitable for churn data. This is followed by the conceptual model and the technical workflow. After this, the data description is given, which is followed by the methodology. The fifth section states the results. The sixth section discusses the main findings and the answer to the research question, accompanied by the limitations of this research.

2 Theoretical Background

This research uses modelling approaches to predict churn in a subscription-based B2B context. This section covers an overview of the existing literature on customer retention and churn, as well as the different modelling approaches and machine learning methods that previous research used in churn prediction models. Additionally, the class imbalance problem in churn is described, and an overview of different balancing techniques used in churn prediction modelling is provided. Lastly, the research design is described using a conceptual framework and the technical workflow of this research.

2.1 Customer retention and churn

Customer retention and customer churn are both relevant concepts in the current B2B world. These two concepts are often mentioned together. However, it is important to first address both concepts separately. Therefore, this section first discusses churn, which is followed by an explanation of customer retention. Lazear & Spletzer (2012) defined churn in the context of recruitment as the number of hires and the number of employees that left the company that offset each other in the company. In the telecommunication industry, churn is defined as a customer that discards the services, either because they are dissatisfied or because other providers have better offers. This indicates that the customer stops using the services of the company of interest, which induces a loss of revenue and/or profit (Umayaparvathi & Iyakutti, 2016). Furthermore, other past research defined churn as abandoning a company in favor of a competitor (Ferreira et al., 2004). A churned customer has thus moved to a competitor or simply stopped transacting with the company of interest (Dingli et al., 2017). Ascarza et al. (2018) researched the possible differences in the reactions to marketing communication between two types of churn: silent churn and overt churn. They found differences in the behaviour of silent and overt churners. However, the reason for churn is not relevant within the scope of this research. Therefore, in this research, churn is defined as a customer that decides to stop their subscription and stops using the services that the company offers.

Customer retention is defined as the propensity of a customer to stay with the company by Danesh et al. (2012). This definition was elaborated to include the marketing actions that are taken to retain existing customers by "establishing, maintaining and maximizing mutual long-term benefits that strengthen and extend the joint relationship between two parties" (Alshurideh, 2016, p. 383). Customer retention relates closely to the retention rate in a certain period, which is defined as the proportion of customers that had an active contract in the beginning of the period, and the customers that still have an active contract at the end of the period (Fader & Hardie, 2007). Within this research, customer retention is defined as all actions that are taken to retain existing customers and thus keep a high retention rate, including both marketing actions and customer relationship management aimed at building relationships, such that the customer does not leave the company.

As stated before, acquiring a new customer is more costly than retaining an existing customer (Dingli et al., 2017; Verbeke et al., 2012). Retaining a customer leads to a financial benefit, if actions can be taken to prevent the customer from churning (Ferreira et al., 2004). Besides the direct financial benefit that follows from customer retention, customers that have stayed with the company for a longer time are less likely to switch to a competitor and might even lead to the acquisition of more customers through positive word of mouth. Customer retention through accurate churn predictions can eventually thus create a competitive advantage (Ferreira et al., 2004). Additionally, in subscription-based services, customer churn leads to opportunity costs due to the loss of sales (Verbeke et al., 2012). These subscription-based companies also start to realize that their current

customer database is valuable, which increases the use of marketing strategies aimed at retaining current customers. Therefore, it is crucial for companies to accurately predict churn and to get insights into why customers churn such that actions can be taken to prevent this customer churn (Ascarza & Hardie, 2013; Coussement & Van den Poel, 2008).

Past churn research shows two main streams. The first stream focuses on determining factors that induce customer churn by investigating, among others, customer satisfaction, the induced switching cost, and demographics of the churned customers. The second stream focuses on churn prediction using modelling approaches to accurately predict churn and identify the customers that are most likely to churn in order to assign resources to retain these customers. As this research focuses on churn prediction using modelling approaches, the following section elaborates on the different modelling approaches that have been used in past research for churn prediction modelling.

2.2 Modelling approaches in churn prediction

Churn prediction models have been created and researched a lot in the past. Generally, data mining techniques can be categorized into six main categories: association, classification, clustering, forecasting, regression, sequence discovery, and visualization (Ngai et al., 2009). As stated before, churn prediction is a binary prediction problem. In the context of this research, historical data is used, which indicates a supervised learning problem. Therefore, within the field of churn prediction, classification algorithms are mainly used to predict churn. Hence, this section presents six popular modelling techniques used by previous research on churn prediction. An overview of these methods is given in Table 1. Furthermore, the class imbalance problem is often mentioned in churn prediction literature. Hence, the ability to handle imbalanced target variables is an important criterium for churn prediction methods.

2.2.1 Logistic regression

As churn prediction is a binary classification problem, the first model that is considered suitable by past researchers is a logistic regression. Logistic regression is a method that models the probability of belonging to the positive class, given the input variables. The popularity of using a logistic regression is partly explained by the ease of use (Coussement & Van den Poel, 2008). The estimated coefficients are easy to interpret, and the application of logistic regression does not require extensive knowledge of machine learning methods. Additionally, the performance of logistic regression using proper data preparation techniques shows to be competitive with other classification methods (Coussement & Van den Poel, 2008; Neslin et al., 2006; Vafeiadis et al., 2015).

A disadvantage of logistic regression is that the number of input variables increases a lot when there are categorical variables in the input data, due to the higher dimensionality of the regression (Çelik & Osmanoglu, 2019). A high-dimensional model estimates a high number of coefficients. This can induce complete separation, indicating that a combination of these dummy variables perfectly predicts the target variable. This decreases the accuracy of the estimates and increases the complexity of variable selection.

Furthermore, the simplicity of a logistic regression comes with the disadvantage of it being based on assumptions that are not always met in a real-life business setting. One of these assumptions is the requirement for a linear relationship between the input variables and the target variable (Coussement et al., 2010). A logistic regression is thus easily interpretable but not very flexible. Therefore, a logistic regression is a suitable baseline method, but past research found that data preparation techniques are required to handle the class imbalance problem in a churn prediction

context.

2.2.2 Decision Tree

Decision Trees (DT) are visual representations of decision rules inferred from the data. It is a non-linear and flexible supervised learning method that can be used for both classification and regression. Within the context of this research, DTs are used for classification because of the binary target variable Churn.

In a Decision Tree, the visual representation of the model starts with a root node. This root node indicates the first decision, and the branches that follow from the node show the value for the decision variable that should be followed. Each internal node represents a decision variable, and for each node, the branches represent the decision value again. Hence, the branches form the path from the root node. The Decision Tree ends with leaf nodes, which are reached by following the path from the root node to the leaf nodes, based on all decision splits that are shown in the Decision Tree. The final classification in the leaf nodes are based on majority voting within that leaf group. The abovementioned process thus results in a Decision Tree, where rules can be inferred from the branches in the tree to obtain more insights into what influences the class of the target variable (Kirui et al., 2013).

Decision Trees are a popular method in a binary classification context because of their interpretability. In a business context, Decision Trees are considered to have higher interpretability than logistic regression models because of the visual representation in a tree structure, which makes it easy to follow a decision path (Coussement & Van den Poel, 2008; Keramati et al., 2014; Shaaban et al., 2012). Furthermore, Decision Trees are inexpensive to build since it does not require large amounts of training data. Additionally, DT's are flexible because DT's support both numeric and categorical input variables, and no assumptions are made about the distribution of the data (Keramati et al., 2014). According to Çelik & Osmanoglu (2019), Decision Trees are one of the preferred classification algorithms because it is easy to integrate into databases.

Regarding the predictive performance, Ngai et al. (2009) found that Decision Trees do not have a good out of sample performance in the context of churn prediction. Larivière & Van den Poel (2005) and Vafeiadis et al. (2015) also stated that Decision Trees are not very robust and do not have optimal performance. Hence, Decision Trees are highly interpretable and are flexible regarding the input variables, but its' predictive performance is not very high in the context of the imbalanced target variable `Churn`.

2.2.3 Random Forests

Random Forests build on the concept of Decision Trees. It is an ensemble method that combines bootstrapping and feature selection. Random Forests were first introduced by Breiman (2001) to increase the variation of the predictions. Random Forests are an extension of Bagging. This is an ensemble method where each Decision Tree is built on a bootstrapped sample of the training data. Random Forests extend this by using only a random subset of the input variables in each split. To finally classify the data, the majority vote of the predictions is used. This indicates that the classification of each Decision Tree is taken into account, and the majority vote decides which class the Random Forest predicts.

Compared to Decision Trees, Random Forests are more robust and less sensitive to noise. Each separate Decision Tree in the Random Forest is built on a bootstrapped sample of the training data. This creates artificial out of sample predictions to improve the out of sample performance of Random

Forests, compared to single Decision Trees (Breiman, 2001). Additionally, Coussement & Van den Poel (2008) state that Random Forest models are not highly influenced by outliers because of the bootstrapped samples used as input for each separate tree.

Another advantage of Random Forests is its ease of use. Only two hyperparameters need to be set: the number of input variables per split, `m`, and the number of trees to be grown, `n`, (Coussement & Van den Poel, 2008; Larivière & Van den Poel, 2005). With regards to the performance, Random Forests were found to outperform logistic regression models (Larivière & Van den Poel, 2005). This is in line with the findings of Coussement & Van den Poel (2008) in the context of churn prediction, where it was stated that Random Forests are among the best performing models for churn prediction. Furthermore, Rahman & Kumar (2020) compared multiple machine learning models in a churn prediction context. They also found that Random Forests are the best performing models, especially when oversampling techniques are used in the data pre-processing stage.

Random Forests thus have a higher out of sample performance than single Decision Trees. This increase in predictive performance does come at the cost of interpretability. Random Forests are an ensemble method, which indicates that Random Forests are not directly interpretable. More advanced global and/or local interpretation methods are required to provide information on the factors influencing churn.

2.2.4 Naive Bayes

Naive Bayes is a probabilistic classifier based on the Bayesian theorem. This classifier analyzes the relationship between the input variables and the target variable by assuming that the presence of the target variable is not related to the presence or absence of other variables. Hence, it assumes that a specific independent variable is unrelated to the target variable. This indicates that the Naive Bayes model analyzes the relationship between the independent variables and the target variable based on conditional probabilities for these relationships (Kirui et al., 2013).

More specifically, the Naive Bayes algorithm computes the probability of each class by counting the number of occurrences in the input data (the prior probability). Subsequently, the algorithm calculates the probability given a class, for each row in the input data. Assuming that the input variables are independent of this probability, this probability is computed as the product of probabilities for every input variable (Çelik & Osmanoglu, 2019). A Naive Bayes model is thus directly interpretable.

With regards to the performance of Naive Bayes in the context of churn prediction modelling, Kirui et al. (2013) found that Naive Bayes shows a better predictive performance than Decision Trees. They used various sets of input variables, but for each dataset, the Naive Bayes model outperforms the Decision Tree model. On the contrary, Vafeiadis et al. (2015) concluded that the Naive Bayes classifier did not have a good predictive performance. In the context of their research, the Naive Bayes classifier had a similar performance as the logistic regression model, while their Decision Tree and the Support Vector machine models had high predictive performance.

2.2.5 Support Vector Machines

Looking further into previous research about churn prediction modelling, Support Vector Machines (SVM) are often used. SVMs were first introduced by Boser et al. (1992), who proposed an algorithm that maximizes the margin between the training patterns and the decision boundary. Previous research stated that SVM can be used for both classification and regression tasks. Within the context of churn prediction it is used for a classification task, where the model aims to predict

whether a customer churns or stays with the company.

There are two types of Support Vector Machines (Çelik & Osmanoglu, 2019). In the basic application of SVM, where the data is linearly separable, the training data is split into two classes by creating a hyperplane defined by the support vectors. These support vectors are a linear combination of subsets of the training data, which are chosen based on the maximal margin (Huang et al., 2012; Keramati et al., 2014). In the case of non-linearly separable data, a kernel function is used to separate the data in a non-linear way by transforming the existing input data into a high-dimensional feature space. This makes sure that it is possible to classify the data (Huang et al., 2012; Keramati et al., 2014).

With regards to the performance of SVM in the context of churn predictions, Coussement & Van den Poel (2008) found that SVM has good performance and is generalizable to out of sample data. Comparing Decision Trees, SVM, and neural networks, Shaaban et al. (2012) found that SVM leads to the best results in terms of the predictive performance for churn classification. This was also confirmed by Vafeiadis et al. (2015), who concluded that SVM outperforms both Decision Trees and sometimes Artificial Neural Networks. On the other hand, Daskalaki et al. (2006) researched the performance of multiple methods when using data with class imbalance. They found that SVM did not perform well when under-sampling was used to handle this class imbalance.

Another way to handle the class imbalance problem is by using a one-class classifier SVM, introduced by Schölkopf et al. (2001) and adapted by Li et al. (2003) to detect anomalies. For anomaly detection, all data points in the positive class are mapped in a feature space, with the distance to the origin as a measure. A new data point is 'predicted' as an anomaly when the new data point matches the selected data points mapped in the feature space. In a churn prediction context, Zhao et al. (2005) used one-class SVM, and found that using one-class SVM outperformed the other methods used in their research.

Generally, SVM thus has a good performance, and adapting SVM to a one-class SVM enables SVM to handle the class imbalance problem. However, this increased performance does come at the cost of interpretability. One of the disadvantages of SVM is thus that more advanced global and/or local interpretation methods are required to provide information on the factors that influence churn. This decreases the interpretability of SVM in a business context because more advanced knowledge of machine learning is required.

2.2.6 Neural Networks

Neural networks are used in clustering and prediction problems, but they are also used for classification. Within the context of churn prediction modelling, it is thus a widely used method (Ngai et al., 2009). Neural networks are developed to simulate how the human brain works. It is an information processing system designed to imitate the functions of the neural networks in the brain (Çelik & Osmanoglu, 2019). Neural networks are different from other classification models, such as Decision Trees, because neural networks output a prediction, accompanied by a likelihood of belonging to the predicted class (Shaaban et al., 2012). Previous research confirms that various types of neural networks are able to achieve high predictive performance in the context of churn prediction modelling (Dingli et al., 2017; Keramati et al., 2014; Vafeiadis et al., 2015).

Various types of neural networks have been mentioned in previous research. For instance, Dingli et al. (2017) used a Restricted Boltzmann Machine (RBM). This is a stochastic neural network that discovers patterns in the input data. This type of neural network has three layers: the output layer, the hidden layer, and the visible layer. An advantage of this is that this model is capable of representing any distribution, which can be further improved by increasing the number of units in

the hidden layer. Besides a Restricted Boltzmann Machine, Artificial Neural Networks were applied to predict churn. Artificial Neural Networks share the advantage of having a hidden layer with RMD, which enables the model to detect patterns from the input data (Keramati et al., 2014). The flexibility of Neural Networks is thus a big advantage.

However, a disadvantage of using neural networks is the lack of explanation capability. Neural networks have a black-box nature, which indicates that the model is not (directly) interpretable (Keramati et al., 2014). Furthermore, Huang et al. (2012) stated that building and using neural networks to predict churn is very computationally expensive. The increase in predictive power compared to other churn prediction models is thus outweighed by the lack of explanatory power and the increase in the computational time and power that is required for neural networks (Huang et al., 2012).

2.2.7 Evaluation of modelling approaches

As can be taken from Table 1, this research evaluates six models for churn prediction modelling. In the context of this research, the most important criteria are the predictive power, the interpretability and the required computational power, which is indicated by the scalability. This section compares the six proposed modelling approaches to balance the predictive power with the interpretability and the required computational power of the methods. The three aforementioned criteria are considered the most important criteria in this research. However, past research also addresses several other criteria, which are included in Table 1 to be complete.

As Table 1 shows, the interpretability is an advantage of logistic regression and Decision Trees (Coussement & Van den Poel, 2008; Keramati et al., 2014; Ngai et al., 2009; Shaaban et al., 2012; Vafeiadis et al., 2015). With regards to the performance of these models, previous literature found that a logistic regression model can have competitive performance in a churn prediction context (Coussement & Van den Poel, 2008; Neslin et al., 2006; Vafeiadis et al., 2015). For the Decision Tree models, Table 1 shows that this type of model generally has a bad out of sample performance (Ngai et al., 2009; Vafeiadis et al., 2015). However, Vafeiadis et al. (2015) did find good results in the context of churn prediction modelling, which was also found by Neslin et al. (2006).

Past research does not agree on the predictive performance of logistic regression models and Decision Trees in the context of churn prediction modelling. However, as can be taken from Table 1, these two methods are the only two directly interpretable methods that have often been used in past churn prediction literature. Besides the interpretability and the predictive performance, Coussement et al. (2010) mentioned the disadvantage of the linearity assumption of the logistic regression. Furthermore, Çelik & Osmanoglu (2019) stated that a logistic regression is mainly suitable for low-dimensional data. On the other hand, Decision Trees are easy to integrate into databases and are inexpensive to build (Çelik & Osmanoglu, 2019; Keramati et al., 2014). Therefore, this research proposes to use a Decision Tree to create an interpretable baseline method for churn prediction in a B2B setting.

The other four methods shown in Table 1 are not directly interpretable. Therefore, these methods will be evaluated based on the predictive performance, the interpreting capability, and the required computational power. Naive Bayes is easy to use (Çelik & Osmanoglu, 2019). However, Kirui et al. (2013) found that Naive Bayes is outperformed by Decision Trees, which is in line with the findings of Vafeiadis et al. (2015). Furthermore, Vafeiadis et al. (2015) also found that Naive Bayes is outperformed by Support Vector Machines. Therefore, Naive Bayes is not applied in this research.

Neural Networks are an often mentioned modelling approach applied for churn prediction (see Table 1). Neural Networks have a high predictive performance (Keramati et al., 2014; Shaaban

et al., 2012; Vafeiadis et al., 2015). Additionally, NN are capable of capturing any distribution and are good at detecting patterns (Dingli et al., 2017; Keramati et al., 2014). However, NN are computationally expensive and do not provide interpretable results. NN are therefore not suitable for churn predictions in a business context, where resources might be limited. Additionally, NN are not interpretable because of their internal structure, which decreases its' suitability for business problems (Huang et al., 2012). Therefore, Neural Networks are not applied in this research.

Random Forests are more robust and are less sensitive to outliers compared to Decision Trees (Breiman, 2001; Coussement & Van den Poel, 2008). Furthermore, RF outperforms logistic regression models (Larivière & Van den Poel, 2005). Coussement & Van den Poel (2008) concluded that Random Forests are the best performing method for churn prediction. On the contrary, Support Vector Machines show bad predictive performance when undersampling is used (Daskalaki et al., 2006). However, Shaaban et al. (2012) and Vafeiadis et al. (2015) found that SVM outperforms various other classification methods. With regards to the required computational power, Larivière & Van den Poel (2005) stated that Random Forests have a reasonable computation time. For SVM, past research does not mention the computational time as a disadvantage of this method. Therefore, this research applies Random Forests and Support Vector Machines to predict churn in a B2B setting.

To summarize, a Decision Tree is used to build an interpretable baseline model to predict churn, based on the interpretability criterium. The evaluation of Naive Bayes, SVM, Random Forests, and NN based on the predictive performance, the interpretability, and the required computational power concluded that Random Forests and Support Vector Machines are the best suitable modelling approaches. Hence, this research applies Decision Trees, Random Forests and Support Vector Machines to predict churn in a B2B context.

Table 1: Overview of literature on churn prediction modelling

	Predictor equation	Interpretability	Robustness	Predictive performance	Complexity in usage	Scalability	Literature
Logistic Regression	Parametric	High	Low	Low	Low	Short	Coussement (2008), Coussement (2010), Neslin (2006), Vafeiadis (2015)
Decision Tree	Non-parametric	High	Low	Low	Low	Short	Coussement (2006), Celik (2019), Keramati (2014), Lariviere (2005), Ngai (2009), Vafeiadis (2015)
Random Forests	Non-parametric	Low	High	High	Low	Short	Breiman (2001), Coussement (2008), Lariviere (2005)
Naive Bayes	Non-parametric	Low	Medium	Medium	Low	Short	Celik (2019), Kirui (2013), Vafeiadis (2015)
Support Vector Machines	Parametric	Low	High	High	Medium	Medium	Coussement (2008), Daskalaki (2006), Keramati (2014), Shaaban (2012), Vafeiadis (2015)
Neural Networks	Parametric	Low	High	High	High	Long	Celik (2019), Dingli (2017), Huang (2012), Karamati (2014), Ngai (2009), Shaaban (2012), Vafeiadis (2015)

2.3 Handling Class Imbalance in Churn prediction

The class imbalance problem is an often mentioned phenomenon in churn prediction data. In a binary classification problem, this indicates that the target variable has an uneven distribution. Because the minority class is often the class of interest of the target variable, this uneven distribution induces challenges for modelling churn prediction. For instance, if the input data has an imbalanced target variable where 99% of the data belongs to the majority class, a classifier is already 99% accurate when it ignores the 1% minority class. Hence, handling the class imbalance problem is relevant to prevent the classification model from providing misleading results (Amin et al., 2016).

Six categories of problems are known to arise in case of the class imbalance problem. These categories are summarized by Burez & Van den Poel (2009), based on the categorization of Weiss (2004). In the context of this research, improper evaluation metrics and the lack of data on the minority class are the two most relevant problems. The first, improper evaluation metrics, will be discussed in the methods section. The second, the lack of data in the minority class, can partly be solved by using sampling techniques (Burez & Van den Poel, 2009).

There are three main approaches to tackle the class imbalance problem in binary classification models. First of all, basic sampling methods can be used. However, these basic sampling methods have their disadvantages, leading to the second approach: advanced sampling methods. Thirdly, class imbalance can be handled on an internal (algorithm) level. However, as stated by Miguéis et al. (2017), sampling is a suitable method to handle the class imbalance problem for all algorithms, making it more general. Furthermore, the internal modification of algorithms to handle the class imbalance problem is usually complicated. Therefore, this research focuses on solving the class imbalance problem on an external level in the data preparation stage.

The remainder of this section first introduces the two basic sampling methods. After this, more advanced sampling methods that have been used in previous research are discussed. Lastly, all discussed balancing techniques are evaluated to determine which balancing techniques are most suitable for this research.

2.3.1 Basic sampling methods

The first and most basic balancing techniques are random under- or oversampling to balance the target variable. The first basic sampling method, random under-sampling, eliminates observations from the majority class (Burez & Van den Poel, 2009; Nguyen & Duong, 2021; Weiss, 2004). A disadvantage of under-sampling is that important information can be removed because it discards potentially valuable observations from the majority class. Therefore, it is possible that under-sampling decreases the performance of the model (Burez & Van den Poel, 2009; Wang et al., 2021).

Regarding the effect of under-sampling on the predictive performance of churn prediction models, Ling & Yen (2001) found that under-sampling yields to the best predictive performance when the data is under-sampled to a 50/50 distribution of the two classes in the target variable. However, in the context of churn prediction modelling, Burez & Van den Poel (2009) did not find a similar result. They concluded that under-sampling does increase the predictive performance of the churn model, but it is not needed to under-sample to a 50/50 distribution in the data. Japkowicz (2000) and Chawla et al. (2002) also found that under-sampling is a suitable balancing technique. They even found that there is no need to use more advanced sampling techniques in the context of neural networks. This aligns with Miguéis et al. (2017), who found an increase in the predictive performance when undersampling was used.

Contrary to under-sampling, over-sampling randomly duplicates observations from the minority class. It thus creates additional training data, which balances the data based on the target variable. Because of the creation of additional training data, over-sampling might increase the computation time needed to build the churn prediction model. This specifically applies to large datasets. Additionally, because over-sampling duplicates observations from the minority class, there is a risk of overfitting (Chawla, 2009; Weiss, 2004).

According to Japkowicz (2000), over-sampling increases the performance of neural networks, such that there is no need to apply more advanced sampling methods. Furthermore, Gui (2017) found a similar performance for over-sampling and a more advanced sampling method. On the other hand, Amin et al. (2016) concluded that more advanced sampling methods outperform both over- and under-sampling. Past research thus does not agree upon the effect of basic sampling techniques on the predictive performance of churn prediction models. Therefore, the next section elaborates on more advanced balancing techniques.

2.3.2 Advanced sampling methods

Previous research applied various advanced balancing techniques to handle the class imbalance problem. This section briefly discusses five of these advanced balancing techniques: MTDF, SMOTE, ADASYN, MWMOTE, and CUBE are compared based on past literature on these balancing techniques in a churn prediction context.

Mega-trend diffusion function (MTDF) is an advanced sampling technique first introduced by Li et al. (2007). MTDF creates artificial data to balance the dataset. It is a function that systematically estimates domain samples. It is thus applied when an oversampling approach is desired in the churn prediction data (Amin et al., 2016). Regarding the predictive performance of MTDF, Amin et al. (2016) found that MTDF outperforms most other advanced oversampling techniques. However, the analysis of this research is performed in R, which does not provide a direct application of MTDF. Therefore, MTDF is not considered suitable in the context of this research.

One of the most often used advanced sampling techniques in a churn prediction context is the Synthetic Minority oversampling technique (SMOTE). SMOTE was first introduced by Chawla et al. (2002). It oversamples the minority class by creating new observations in the minority class. These new observations are created by calculating the weighted average of the k-nearest neighbors in the minority class. Thus, it uses the feature space to create new data instead of using duplication or replacement (Amin et al., 2016; Nguyen & Duong, 2021; Wang et al., 2021). By doing so, SMOTE reduces overfitting and minimizes the cost compared to handling the CIP on the algorithm level (Gui, 2017). On the other hand, because minority instances are created based on the k-nearest minority neighbors, it ignores the majority class instances that are close. SMOTE is thus sensitive to the data complexity (Wang et al., 2021). With regards to the performance, SMOTE increased the performance of multiple modelling approaches in a churn prediction context (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017).

The Adaptive Synthetic Sampling Approach (ADADYN) is an extension of SMOTE and was first introduced by He et al. (2008). In the SMOTE algorithm, the number of new observations to be created is a hyperparameter to set. ADASYN extends the SMOTE algorithm by automatically deciding the number of observations to be created. Furthermore, it forces the algorithm to focus on more complex observations in the dataset (Amin et al., 2016). Previous research found increased predictive performance for ADASYN compared to SMOTE (He et al., 2008). However, Amin et al. (2016) could not confirm this predictive performance increase in the context of their comparative research. It is thus interesting to investigate the performance of ADASYN compared to the SMOTE

algorithm in a churn prediction context, which is performed in this research.

Both SMOTE and ADASYN create synthetic new observations in the minority class to oversample this minority class. However, neither of these methods considers the data complexity (Amin et al., 2016). Majority Weighted Minority oversampling (MWMOTE) takes the observations in the minority class that are difficult to learn and assigns a weight to them based on the Euclidean distance from the closest observation from the majority class. MWMOTE is mainly applicable to datasets that have many categorical variables. Because churn prediction data in a B2B context is mixed data, MWMote is not considered suitable in the context of this research.

Furthermore, CUBE is an advanced sampling technique used in past churn prediction literature. The CUBE algorithm was introduced by Deville & Tillé (2004). CUBE selects balanced observations by selecting observations for which the Horvitz-Thompson estimates of the auxiliary variables are nearly equal to the population totals. However, past research did not find an increased predictive performance when the CUBE algorithm is applied to the unbalanced data (Burez & Van den Poel, 2009; Nguyen & Duong, 2021). Therefore, the CUBE sampling method is not considered in the context of this research.

2.3.3 Evaluation of sampling methods

The previous subsections, 2.3.1 and 2.3.2, discussed various balancing techniques to handle the class imbalance problem in Churn prediction data. Both basic and advanced balancing techniques are discussed. This section evaluates the aforementioned methods to determine which balancing techniques are most suitable in the context of this research, where a churn prediction model is built in a B2B context.

With regards to the basic balancing techniques, past research found that both under- and over-sampling can improve the predictive accuracy (Burez & Van den Poel, 2009; Gui, 2017; Japkowicz, 2000). However, under-sampling removes potentially valuable information because it eliminates random observations from the majority class (Wang et al., 2021). On the contrary, over-sampling generates more data, which does come at the cost of overfitting and computational time. However, over-sampling helps to solve the absolute data problem (Amin et al., 2016; Weiss, 2004). This research is performed using a dataset of limited size, so the computational time is not an issue. Therefore, over-sampling is applied in this research as a basic balancing method.

Furthermore, the previous section discussed advanced balancing techniques. MTDF is discussed and has been shown to outperform various other balancing techniques (Amin et al., 2016). However, due to the complexity of its application, this technique is not applied in this research. Furthermore, the CUBE algorithm is used in past churn prediction literature. However, various research found bad performance for the CUBE algorithm, which is why this algorithm is not applied in this research (Burez & Van den Poel, 2009; Nguyen & Duong, 2021).

SMOTE and ADASYN are the most often used advanced sampling techniques in a churn prediction context. SMOTE reduces overfitting and costs compared to handling class imbalance on an internal (algorithm) level (Gui, 2017). Furthermore, SMOTE was found to improve the predictive performance compared to basic balancing techniques (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017). Therefore, SMOTE is considered suitable in the context of this research.

ADASYN extends the SMOTE algorithm by self-determining the number of new observations to be created in the minority class (Amin et al., 2016). Amin et al. (2016) did not find an increased performance of ADASYN compared to SMOTE, but at the introduction of ADASYN by He et al. (2008) ADASYN did outperform SMOTE. Therefore, it is relevant to compare the performance of these two advanced balancing techniques in the context of churn prediction modelling.

In summary, this research applies random over-sampling, SMOTE, and ADASYN, to balance the churn data and compare the performance of these three balancing techniques. These three balancing techniques are combined with the modelling approaches that were found most suitable based on the literature review. The next section describes the research design that combines the balancing techniques with the modelling approaches.

2.4 Research Design

2.4.1 Conceptual framework

This research performs a comparative study of the effect of various combinations of balancing techniques and modelling approaches on the predictive performance of churn prediction models in a B2B context. From the literature study, Decision Trees, Random Forests and Support Vector machines showed to be the most suitable modelling approaches in a churn prediction context. The most suitable balancing techniques based on past literature are random over-sampling, SMOTE and ADASYN.

Each of these balancing techniques is applied to the training data. Subsequently, the original training dataset, and each of the balanced datasets, are used as input for the three models that resulted from the method evaluation. In total, this research thus creates 3×4 churn prediction models to compare the performance of different balancing techniques. Finally, the performance of these 12 models is evaluated and compared based on four evaluation metrics, which will be elaborated on further in this research. A visual representation of the conceptual framework of this research is provided in Figure 1.

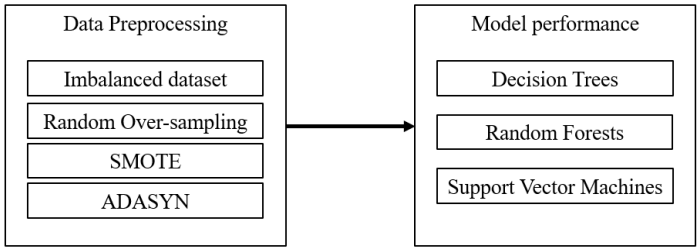

Figure 1: Conceptual Framework

2.4.2 Technical workflow

The flowchart in Figure 2 shows a general overview of the technical route followed in this research. In the first stage, the data pre-processing is done. This includes merging the datasets, data cleaning, and splitting the data into an 80% training set and a 20% test set, to evaluate the out-of-sample performance of the models based on the test set. The last part of the data pre-processing stage includes handling the class imbalance problem on an external level by applying the aforementioned balancing techniques to the training data: over-sampling, SMOTE, and ADASYN. This yields to four cleaned training datasets: one imbalanced dataset and three balanced datasets. It should be noted that no balancing techniques are applied to the test data to be able to evaluate the performance in a real life business context.

Each of these four training data sets is then used in stage 2, which represents the modelling stage. All four datasets are used as input data for each of the three proposed methods: Decision

Tree, Random Forest, and Support Vector machines. The out-of-sample performance of the 12 models is then evaluated and compared by using each of the 12 models to predict churn in the test set. The performance is evaluated on various evaluation metrics, which will be elaborated on in the Methods section.

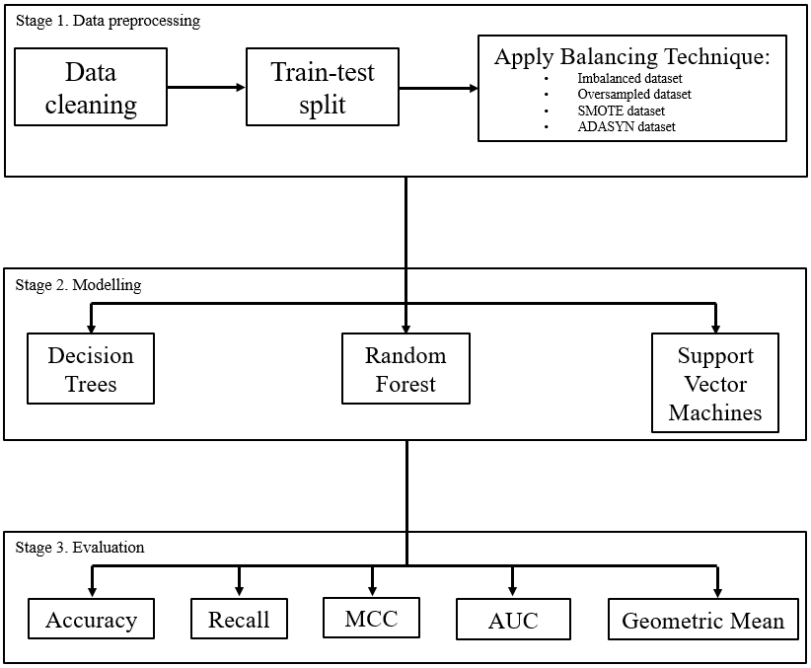

Figure 2: Technical Workflow

3 Data

This section dives into the data used in this research. First, the data is described by introducing the company background and diving into the datasets used. This is followed by a description of the data pre-processing performed on the dataset. This includes an explanation of the applied balancing techniques. Finally, this section explains the data operationalization, which explains how the data is used to answer the research question.

3.1 Data description

3.1.1 Company Background

The data used in this research is provided by TOPdesk Nederland B.V (TOPdesk). TOPdesk is a Business-to-Business company that develops and provides subscription-based service management software to their customers to improve the IT, FM, and HR services and processes. The first version of the software was developed in 1994 when the founders worked for another company. The company that is nowadays known as TOPdesk Nederland B.V. was eventually founded in 1997 by the two developers from Delft. Currently, in 2022, TOPdesk is on the 7th version of its' software, and the company has over 900 employees spread over 14 international offices.

Within the context of this research, TOPdesk's customer data is used to predict churn based on various variables. In TOPdesk's data, churn is defined as a customer that decides to stop their subscription and stops using the software that TOPdesk offers. Hence, an observation with an active contract is a current customer, and an observation that does not have an active contract but had an active contract in the past is a churned customer.

3.1.2 The dataset

The input data in this research is the result of combining 8 datasets that were provided by TOPdesk's Business Support department. In all datasets, each observation is accompanied by an anonymized customer ID. This enables the separate datasets to be merged into one final dataset. The final dataset used for analysis in this research has 3799 observations and 16 variables. A detailed description of the business meaning of these 16 variables can be found in Appendix A.

Within the final dataset, each observation represents either a current or a churned customer. The data includes all churned customers whose subscription ended after the 31st of December 2018. Furthermore, the data is exported on the 30th of May 2022, which indicates that all current customers who became a customer before the 30th of May 2022 are included in the data.

With regards to the target variable, `Churn`, there are 897 churned customers within the 3799 observations. This indicates that out of all observations in the dataset, 23.61% represents a churned customer, and 76.39% represents a current customer. Thus, there is a class imbalance ratio of 1 churned customer to 4 current customers.

The final dataset contains 9 categorical variables and 7 numeric variables. Table 2 shows an overview of the categorical variables in the input data. As can be taken from Table 2, the target variable `Churn` is a binary variable. Furthermore, Table 2 shows three categorical variables with the category "Unknown" (`Unk`). This category indicates that the variable value was either set as "Unknown" in the provided dataset, or the variable value was missing for that observation. This applies to the variables `account_type`, `invoice_frequency`, and `license_type`.

In the context of TOPdesk's data, Table 2 shows the top counts for the categorical variables. First, a customer can have 3 types of turnover. The majority of the observations shows a SaaS

turnover type (Saa). The second most occurring turnover type is a Subscription turnover (Sub); lastly, observations can have a Maintenance turnover type (Mai). Furthermore, there are 7 product lines a customer can choose. As can be taken from Table 2, the top counts are Enterprise (Ent), Engages (Eng), Professional (Pro), and Essential (Ess). Regarding the product generation, there are 5 product generations in the dataset. The most occurring product generations are, in order, the 6th, 7th, 5th, and 4th. Two more product generations are present (2th and 3rd), but these rarely occur in the dataset.

When looking at the Business Units, these categories represent the customer group. The customer group that occurs most is Industry & Retail (IR). This is followed by Professional Services (Pro), Managed Service Providers (MSP) and Healthcare (HC). The account type represents the level of investment required from TOPdesk's sales department for that observation. A low level of investment is represented by the account type Tech-Touch (Tec), followed by Mid-Touch (Mid) and High-Touch(Hig). The invoice frequency indicates the frequency of payments. An observation can have an invoice frequency of a year (12m), a month (mnd), or a quarter (3mn).

Furthermore, a customer in the dataset can have a long contract or not. Long contract being Yes indicates that the customer signed their most recent contract for a period longer than or equal to 36 months. The category No indicates a contract shorter than 36 months. Lastly, the license type shown in Table 2 can be either based on the number of end-users of the software (end) or on the number of operators that use the software (ope). The last categorical variable is the target variable Churn, which can either indicate no churn (0) or churn (1).

Table 2: Overview categorical variables

Variable Name	# Categories	Top Counts
td_customer_turnover_type	3	Saa: 2814, Sub: 528, Mai: 457
td_customer_product_line	7	Ent: 2031, Eng: 727, Pro: 450, Ess: 370
td_customer_product_generation	5	6: 2140, 7: 1137, 5: 415, 4: 95
business_unit	9	IR: 878, Pro: 743, MSP: 721, HC: 486
account_type	4	Tec: 2303, Mid: 831, Hig: 463, Unk: 202
invoice_frequency	5	12m: 2744, mnd: 879, Unk: 83, 3mn: 61
long_contract	2	Yes: 2007, No: 1792
license_type	3	end: 2013, ope: 1498, unk: 288
Churn	2	0: 2902, 1: 897

Besides the 9 categorical variables, the final dataset includes 7 numeric variables. Table 3 shows an overview of these 7 numeric input variables, accompanied by their mean and standard deviation. Additionally, Figure 3 shows a correlation plot of these numeric variables. The only numeric variable that contained missing values is days_since_last_consultancy. Investigating these missing values shows that the value for this variable is missing for observations that have not invested in consultancy in their customer lifetime. Therefore, these 464 missing values are replaced by the largest possible value in the timeframe of the dataset: 7785 days.

As can be taken from Table 3, license_bracket, total_investments_eur, total_tickets, and td_customer_arr_eur have a relatively large standard deviation compared to the mean. In addition to this, Figure 3 shows that these four variables are positively correlated with each other. In the context of TOPdesk's data, these variables all relate to the company size of the customer. The positive correlation between these variables confirms that, for instance, a customer with a higher license bracket generally has a higher ARR. The large standard deviation indicates large differences in the size of the customers included in the dataset.

Table 3: Overview numeric variables

Variable Name	Mean	St. Dev.
license_bracket (number of agents allowed)	29.333	82.637
age_months (customer age in months)	114.543	77.357
days_since_last_consultancy (time since last consultancy in days)	1862.857	2365.431
total_investments_eur (total number of euro's spend on investments)	17453.497	46427.284
total_tickets (total number of tickets in the customer lifetime)	141.396	201.650
td_customer_arr_eur (Annual Recurring Revenue in euro's)	14609.301	20139.119
invoice_time (days between send & payment date of last invoice)	43.397	32.418

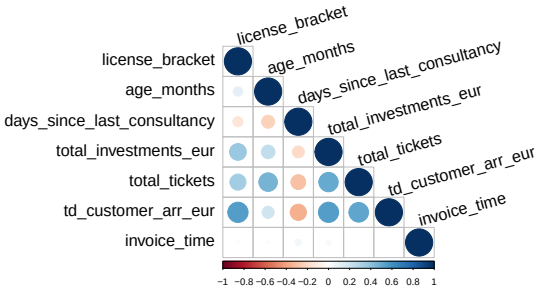

Figure 3: Correlation Plot

3.2 Data pre-processing

TOPdesk's Business Support department provided 8 separate datasets in an .xlsx format, in which each observation is accompanied by an anonymized customer ID. These datasets are merged into one dataset based on this ID. After merging, the dataset is cleaned. This includes factorizing the categorical variables in the dataset to enable R to use these variables in the modelling stage. Besides that, missing values are handled. There are 1012 missing values in `total_invesments_eur` and 40 missing values in `total_tickets`. A missing value in these variables indicates, respectively, that there were no investments or tickets made. Therefore, those missing values are replaced by 0. Furthermore, missing values in the categorical variables `account_type`, `invoice_frequency`, and `license_type` are replaced by the category "Unknown."

After merging and cleaning the data, the data is randomly split into an 80% train set and a 20% test set. No further pre-processing steps are performed on the test set. The training dataset is further pre-processed by applying balancing techniques to create three additional training datasets, such that the data is balanced based on the target variable, `Churn`. The imbalanced dataset contains 23.61% churned customers and 76.39% current customers. The following three subsections elaborate on the three applied balancing techniques: Random Over-sampling, SMOTE, and ADASYN.

3.2.1 Random Over-sampling

Basic sampling techniques randomly eliminate or duplicate observations in the training data to minimize the class imbalance. The first applied balancing technique, basic Random Over-sampling, thus decreases the rarity of the target class by randomly duplicating observations from the minority class. In the context of this research, random over-sampling is used to randomly duplicate churned customers to balance the distribution of the target variable, `Churn`. The Random Over-sampled training dataset contains 50.3% churned customers and 49.7% current customers.

3.2.2 Synthetic Minority Over-sampling Technique

The second balancing technique applied in the context of this research is Synthetic Minority Over-sampling Technique (SMOTE). SMOTE is an over-sampling approach first introduced by Chawla et al. (2002). The SMOTE algorithm generates new data in the minority class by operating in the feature space instead of over-sampling using duplicates. In the context of this research, the SMOTE algorithm generates artificial churned customers to balance the distribution of the target variable, `Churn`.

These artificial observations are created by looking at each observation in the minority class and selecting the `k` nearest minority neighbors. The required oversampling percentage determines how many of these `k` neighbors are selected. For instance, if the required oversampling percentage is 200%, the SMOTE algorithm selects 2 of the `k` nearest neighbors. After selecting these observations, the SMOTE algorithm generates a new synthetic observation in the feature space between the selected minority observation and each of the selected neighbors. Each new observation thus is a convex combination of the selected minority sample and each of the selected neighbors. Detailed information on the SMOTE algorithm can be found in Chawla et al. (2002).

After applying the SMOTE algorithm to the training dataset, the SMOTE dataset contains 42.9% churned customers and 57.1% current customers.

3.2.3 Adaptive Synthetic Sampling

The SMOTE algorithm was extended to an Adaptive Synthetic sampling (ADASYN) approach by He et al. (2008), which is the third balancing technique applied in this research. In the SMOTE algorithm, the oversampling percentage is a pre-set hyperparameter. The ADASYN algorithm automatically determines the required number of synthetic samples to generate in the minority class. Furthermore, the ADASYN algorithm creates more observations based on the minority observations that are hard to learn compared to the easy-to-learn minority observations. Hence, in the context of this research, the ADASYN algorithm automatically determines the percentage of new churned customers to create. Additionally, the ADASYN algorithm focuses on the complex existing churners while creating artificial churned customers. Detailed information about the ADASYN algorithm can be found in He et al. (2008).

After appying the ADASYN algorithm to the training dataset, the ADASYN dataset contains 50% churned customers and 50% current customers.

3.3 Data Operationalization

This research is a comparative study of the predictive performance when using different combinations of balancing techniques and modelling approaches in churn prediction models. Hence, this research applies three balancing techniques to the training data to capture the constructs in the conceptual framework. This results in four input datasets for the three models that are applied. The first dataset is the original 80% training dataset, where the class imbalance problem applies. The remaining datasets are the result of applying balancing techniques to the 80% training data. Hence, the second dataset is a dataset where Random Over-sampling is applied. The third dataset is the result of applying the SMOTE algorithm to the training data, and the fourth dataset is the result of applying the ADASYN algorithm to the training data.

After creating the 80% training dataset and the three balanced training datasets, each dataset is used as input data for the three modelling approaches: Decision Trees, Random Forests, and Support Vector Machines. For each of these modelling approaches, four models are built based on the four different input datasets.

This research thus creates 12 models in total. To answer the research question, the performance of each of these models is compared. To evaluate the models, the 20% test set is used to create confusion matrices and compare the performance based on the accuracy, the recall, the MCC, and the AUC. The next section elaborates on the methods and the evaluation metrics used in this research.

4 Methods

This section dives into the methods used throughout this research. This research is a comparative study of the predictive performance of using different balancing techniques and modelling approaches in churn prediction models. Therefore, this section includes an explanation of the modelling approaches for the analysis. First, Decision Trees are explained. Secondly, this section elaborates on Random Forests. Thirdly, Support Vector Machines are explained. Finally, this section discusses the evaluation metrics used to compare the out of sample performance.

4.1 Decision Trees

Decision Trees are first introduced by Breiman et al. (1984). They can be used for both classification and regression problems, but in the context of the binary target variable `Churn`, a binary classification problem is handled. Therefore, this research builds classification trees to predict the class of the target variable `Churn`.

A DT is a graphical model that consists of nodes and branches. It starts with a root node representing the first rule and splits the data into mutually exclusive subgroups. Each subgroup is recursively split based on the most informative split within that subgroup. Each split results in internal nodes, and out of each node, branches represent the variable value for the node. Thus, branches form the path from the root node to the final classification. This final classification is indicated by the terminal nodes (leaf nodes), based on the majority vote within that leaf group.

While building a DT, the two most important choices are the variable to split on and when to stop splitting. The variable splits are chosen based on a cost function representing an impurity criterium because the DT algorithm aims to have the purest leaf nodes as possible. In the context of this research, the Gini coefficient is used. The Gini coefficient has a range from 0 to 1. A value of 0 indicates a pure node with only correct predictions, whereas a value of 1 indicates that all predictions are randomly distributed. To minimize the impurity, the DT thus aims to find the variable split with the largest decrease in the Gini coefficient. The Gini coefficient is defined as 1 minus the sum of all squared probabilities of belonging to a class, see Formula (1).

$$\text{Gini} = \quad 1 - \sum_{i=1}^{n} (p_i)^2 \tag{1}$$

where p_i is the chance of being in class i and n is the number of target classes. Within the context of the binary target variable `Churn`, `i` is either 0 (No Churn) or 1 (Churn). Therefore, `n` is 2.

The second important aspect is the stopping criterium which prevents the DT from overfitting to the training data. Firstly, the minimum number of training observations in a leaf node can be set to determine a stopping criterium. Secondly, the maximum depth of the tree is a hyperparameter that can be set to prevent the DT from overfitting. However, tuning these hyperparameters can still lead to large, overfitted trees. Therefore, pruning based on the complexity parameter, `cp`, is used in this research to determine a stopping criterium. The `cp` introduces a penalty for the complexity of the DT. The tree is pruned if the cost of adding another split is higher than the calculated `cp`. The cost of adding another split is calculated as shown in Formula (2).

$$\sum_{i=1}^{T} misclass_i + \alpha |T| \tag{2}$$

where α is a positive constant that represents the chosen `cp` and T is the number of leaf nodes.

As can be taken from Formula (2), there is a negative relationship between α and the tree size. A smaller α indicates that the penalty of adding another split is lower, which allows for a more complex, larger tree and vice versa. To determine the best `cp`, the model is built using a range of splits, from 1 to the maximum number of splits, accompanied by the corresponding `cp`. The `cp` that leads to the lowest sum of the 10-fold cross-validated, scaled, x-error and the x-std is the final `cp`.

As stated in the method evaluation, Decision Trees are not very robust. This is because small changes in the data can influence the variable splits, which in turn influences all subsequent variable splits. Hence, small changes in the data can affect the whole structure of the Decision Tree. Decision Trees are thus highly interpretable, but their out of sample performance is not very high. Therefore, other methods explained in this section are applied next to the simple, interpretable Decision Tree models.

4.2 Random Forests

The second method used in this research are Random Forests. This ensemble method consists of multiple Decision Trees and was first introduced by Breiman (2001). Similar to Decision Trees, Random Forests can be used for both regression and classification tasks. In the context of this churn prediction research, it is used for classification because of the binary target variable `Churn`.

A Random Forest builds on the concept used in bootstrap aggregating (Bagging), where each Decision Tree is built on a bootstrapped sample of the training data. Random Forests extend this by using feature randomness. Only a random subset of the input variables is used to build each Decision Tree. A Random Forest thus creates `n` different fully grown Decision Trees, based on `n` bootstrapped samples of the training data. To decorrelate these `n` trees and make the model less dependent on dominant variables, a Random Forest selects a random subset `m` of the `p` variables to build each Decision Tree. The final prediction of the Random Forest is based on the majority vote of all `n` Decision Trees. Hence, in the context of this research, the `Churn` prediction for each customer is given using the majority vote of all `n` classification trees.

When building a Random Forest, there are three main hyperparameters to set. Firstly, the node size represents the minimum number of observations required in the leaf nodes of the separate Decision Trees. Secondly, the number of trees `n` needs to be chosen. In the context of this research, where the analysis is performed in R, the default value for the node size is 1 and the default value for `n` is equal to 500. This research uses the default values of the node size and `n`, because Probst & Boulesteix (2017) found a negligible gain in the predictive performance when tuning `n` and the node size in Random Forests.

The third hyperparameter is the number of variables per split, `m`. For classification problems, the default `m` is based on a rule of thumb: $\sqrt{p}$. Probst et al. (2019) found the largest increase in predictive performance when `m` is tuned, compared to tuning the other hyperparameters of Random Forests. Therefore, this research uses 10-fold cross-validation to determine the `m` that leads to the highest predictive power.

With regards to the interpretation, Random Forests are black-box models and are therefore not directly interpretable. Hence, further interpretation methods are required to interpret Random Forest models. However, this is outside the scope of this research, where a performance comparison is made.

4.3 Support Vector Machines

Support Vector Machines are the last method applied in this research to predict churn. SVM is a black-box model that can be used for both regression and classification tasks, but in the context of this research, it is used as a classifier to predict the binary target variable `Churn`. There are multiple types of SVM, including Support Vector Classifiers and Support Vector Machines.

The basic idea behind using a Support Vector Classifier as a binary classifier is to minimize the number of misclassifications by finding an optimal hyperplane with a maximal margin, that separates the data into the two classes. Within the context of this research, a Support Vector Classifier thus aims to find an optimal hyperplane that separates the churned customers from the current customers.

A hyperplane is a subspace of dimension $p-1$, where p indicates the dimension of the space. The mathematical definition of a hyperplane is $x_i^T\beta + \beta_0 = 0$, where x_i is the input vector, β is a normal vector perpendicular to the hyperplane, and β_0 is an intercept. As can be taken from this formula, any combination of $(x_1, x_2, ..., x_p)$ that satisfies the condition is positioned exactly on the hyperplane. To classify the data in the context of this research, 'Churn' is labeled as +1 and 'No Churn' is labeled as -1. The position of an observation relative to the hyperplane determines the predicted class.

To find the hyperplane with the maximum margin, the largest minimum distance to the training data is determined. The data points with this minimum distance are the support vectors. Based on these support vectors, the margin lines are defined as $x_i^T\beta + \beta_0 + \Delta$ and $x_i^T\beta + \beta_0 - \Delta$. If the two classes are linearly separable, then the margin M, represented by Δ, should be maximized. However, in a real-life business setting, the classes are often not linearly separable. In that case, no solution leads to $M > 0$ (James et al., 2013, p. 373).

In the case of training data that is not perfectly linearly separable, one can allow for misclassifications. This is known as a soft margin, which is mathematically shown by including allowance for misclassifications while taking a penalty into account. The objective function of finding the maximal margin hyperplane using a soft margin is shown in Formula (3).

$$\begin{aligned} \min \quad & \sum_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s. t.} \quad & y_i(x_i^T\beta + \beta_0 \geq 1 - \xi_i)\forall i, \\ & \xi_i \geq 0, i = 1, ..., n \end{aligned} \tag{3}$$

where C is the non-negative penalty term that represents the trade-off between the misclassifications and the width of the margin, and ξ is the proportional amount by which a prediction on the wrong side of the hyperplane is penalized.

The trade-off term C has a positive relationship with the allowed misclassifications. If C is set to 0, no misclassifications are allowed, and a large C allows for a higher number of misclassifications. In this research, the use of a soft margin classifier indicates that the Support Vector Classifier allows some churned customers to be classified as non-churners and vice versa.

However, in the case of non-linear decision boundaries in the data, Support Vector Classifiers using a hyperplane do not lead to high predictive performance (James et al., 2013, p. 379). Therefore, Support Vector Machines can either increase the feature space or use a kernel. The linear kernel represents the Support Vector Classifier as explained above. There are various other options for kernels, but in the context of this research, a linear kernel is used to predict `Churn`.

Other kernels are not often used in practice, because it generally shows poor efficiency in a real-life business setting. Hence, this research applies Support Vector Machines with a linear kernel, where misclassifications are allowed up until a certain point, which is determined by tuning the Cost parameter C. Mathematically, the linear kernal is defined as shown in Formula (4).

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij}, x_{i'j} \tag{4}$$

4.4 Performance Evaluation

This section discusses the methods used to evaluate the created models. Before building the models, the data is split into an 80% train set and a 20% test set. The models are evaluated based on the test set, to which no balancing techniques are applied, to represent the out of sample performance. First, the concept of a confusion matrix is explained. Secondly, evaluation metrics based on the confusion matrix are discussed: the accuracy, the recall, and the Matthews Correlation Coefficient (MCC). Lastly, this section discusses the Area under the ROC curve (AUC) as an evaluation metric.

4.4.1 Confusion Matrix

A confusion matrix is a tabular representation of the actual class of the data and the predicted class by the model, see Table 4. The True Positives (TP) indicate the observations that were predicted in the positive class and are actually positive. The True Negatives (TN) indicate the observations that are predicted negative and are actually negative. The False Positives (FP) indicate the observations that have been predicted as positive, but are actually negative. Lastly, the False Negatives (FN) indicate the observations that are predicted as negative, but are actually positive.

Table 4: Confusion Matrix

		Actual	
		Negative	Positive
Predicted	Negative	TN	FN
	Positive	FP	TP

In the context of this research, the confusion matrix shown in Table 4 is interpreted as follows. A TP indicates that the model predicted churn, and that the observation actually is a churned customer. A TN represents a current customer that is indeed predicted as "No Churn." An FP represents an observation that is predicted as churn, but is actually a current customer. Lastly, an FN indicates a customer that was predicted as a current customer, but actually churned.

4.4.2 Accuracy, Recall, Matthews Correlation Coefficient

To evaluate the data, three evaluation metrics based on the confusion matrices of the test set are used. The most commonly used evaluation metric in binary classification problems is the accuracy (Coussement & Van den Poel, 2008). This represents the fraction of correctly classified observations, see Formula (5).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{n} \tag{5}$$

where n indicates the total number of observations. In the context of this research, the accuracy thus represents the observations that are correctly classified as either churn or no churn, relative to the total number of observations in the test set.

This research is performed in a business context and tries to predict churn, in order to help the business make decisions. Therefore, it is important to determine how good the model is at detecting churn. Hence, the recall of each model is used to determine the proportion of the positives that are discovered by the model, see Formula (6).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

Within the context of this research, the recall thus indicates the proportion of the churned customers that are classified as churners by the model. A recall of 0 indicates that none of the churned customers are classified as churners, whereas a recall of 1 indicates that all churned customers are classified as churn by the model.

Although various balancing techniques are used to balance the churn data, the data is still not perfectly balanced. Therefore, it is important to also use an evaluation metric that balances the false negatives and the false positives. Chicco & Jurman (2020) stated various advantages of using the Matthews Correlation Coefficient (MCC) as an evaluation metric for unbalanced data, compared to the accuracy. The MCC is calculated as shown in Formula (7).

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

As can be taken from Formula (7), the MCC ranges from -1 to 1, where -1 indicates a perfect negative correlation between the actual class and the predicted class and 1 indicates a perfect correlation between the two. On the other hand, an MCC of 0 indicates no correlation between the predicted class and the actual class. Within the context of this research, an MCC of 1 indicates that the predicted churn and the actual churn is perfectly correlated. Hence, an MCC close to 1 is desired.

4.4.3 Area under the ROC curve and Geometric Mean

The aforementioned metrics depend on the chosen threshold to classify the observations as either the positive class or the negative class. The most common threshold is 0.50, which indicates that if the predicted probability of the positive class is ≥ 0.50, the observation is predicted as positive and vice versa. The ROC curve extends this information by plotting the False Positive Rate (FPR) against the True Positive Rate (TPR) using all possible thresholds, see Figure 4 (James et al., 2013). Within the ROC curve, the dashed diagonal indicates the line that does not provide information. Hence, the desired ROC curve approaches the top left corner because this indicates a high TPR and a low FPR for the different thresholds.

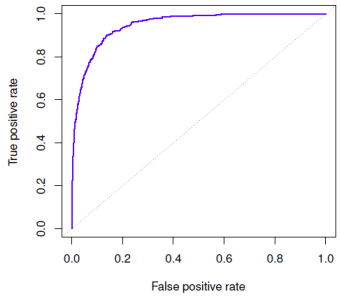

Figure 4: ROC curve

From the ROC curve, the Area Under the Curve is used as a performance metric. The AUC ranges from 0 to 1 and it has a positive relationship with the performance of the binary classifier. The mathematical calculation of the AUC is shown in Formula (8).

$$\text{AUC} = \int_0^1 TPR(FPR)\, dFPR \tag{8}$$

As can be taken from Formula (8), a larger AUC indicates better performance, because this indicates a low FPR and a high TPR. As can be taken from the 45-degree dashed line, an AUC of 0.5 represents a non-informative model that does not perform better than chance.

In addition to the AUC, the geometric mean of the predictions on the test set is provided for each model, to include a performance measure that is not dependent on the decision threshold. In the context of this research, the geometric mean of the Decision Tree models and the Random Forest models indicate the geometric mean of the predicted probabilities of the test set. As stated before, Support Vector Machines use the position of an observation relative to the hyperplane to determine the predicted class. Therefore, the geometric mean of the predictions by Support Vector Machine models are calculated based on the logarithmized position relative to the hyperplane of the observations in the test set. The relative position calculated the SVM does not have a scale from 0 to 1, which differs from the scale of the predicted probabilities by the Decision Tree and Random Forest models. To enable a performance comparison across all models, the decision values of the Support Vector Machines are thus normalized to a scale of 0 to 1 when calculating the geometric mean.

The geometric mean has a positive relationship with the variation in the prediction. Hence, a smaller geometric mean indicates that the predictions are similar. On the contrary, a large geometric mean indicates larger variances in the predictions. Mathematically, the geometric mean is calculated as shown in Formula (9).

$$\text{Geometric Mean} = exp[\frac{\sum_{i=1}^{n} ln(p_i)}{n}] \tag{9}$$

where p_i indicates the probability of observation i of belonging to the target class (Churn), or in case of SVM, the normalized relative position to the hyperplane of observation i. Furthermore, n is the total number of observations. In the context of this research, the performance is evaluated on the test set, to which no balancing techniques are applied. Hence, n indicates the total number of observations in the test set, which equals 678.

5 Results

This research investigates the added value of using advanced balancing techniques on the predictive performance of `Churn` prediction models. In this section, the results of this research are discussed. Three balancing techniques are applied to the `Churn` data: Random Over-sampling, SMOTE, and ADASYN. Additionally, the original, imbalanced training dataset is used. Hence, this results in four training datasets used as input data to predict `Churn`. Each of these datasets is used as input for the three modelling approaches to build a churn prediction model: Decision Tree, Random Forests, and Support Vector Machines. Hence, in total, this research builds 12 models to predict the target variable `Churn` and compares the performance of each of these models.

For each of the three modelling approaches, a default model is built using the default hyperparameters. Additionally, hyperparameter tuning is performed to tune the models. For the Decision Tree models, the complexity parameter is tuned to prune the DTs. The default `cp` is 0.01. The optimal `cp` is chosen based on the lowest sum of the cross-validated, scaled, x-error and x-std. An overview of all visualized Decision Trees, see Appendix C.

In the Random Forest models, the number of randomly chosen variables per split, `m`, is tuned using 10-fold cross-validation. The other hyperparameters are kept at their default. This indicates that the number of trees built (`ntree`) is 500, the minimum number of observations in a leaf (`nodesize`) is 1 and the number of variables per split (`m`) is equal to 4. For the SVM models, a linear kernel is used, and the cost parameter is tuned for this modelling approach. The default cost is equal to 1.

In the context of this research, the hyperparameters in the tuned models do not change a lot compared to the default hyperparameters. Therefore, this section focuses on the performance of the tuned models. For an overview of the performance of the default models, we refer to Appendix C. This section thus discusses the out of sample performance of each of the created models. This out of sample performance is measured by using the models to predict the target variable `Churn` in the test set. It should be noted that no balancing techniques are applied to the test set, to make sure that each model is evaluated on its' performance for real life business data. The final performance comparison is made based on the accuracy, recall, Matthews Correlation Coefficient, the Area Under the ROC curve and the geometric mean of the test set. The first four evaluation metrics all range from 0 to 1, and have a positive relationship with the predictive performance. A value closer to 1 thus indicates a better performance in terms of that evaluation metric. The geometric mean has a positive relation with the variation in the predictions. Hence, a small geometric mean indicates little variation in the predictions, which indicates lower performance.

The remainder of this section is structured as follows. The first subsection dives into the results of using the imbalanced dataset as input data for the modelling approaches. This is followed by the results of using the Random Over-sampled dataset as input dataset. Thirdly, the results of using the SMOTE dataset are stated, and the fourth subsection dives into the results of the models that used the ADASYN dataset as input data. Finally, the out of sample performance of all the 12 models is evaluated and compared based on the accuracy, the recall, the Matthews Correlation Coefficient the Area under the ROC curve and the geometric mean of the predictions on the test set. Furthermore, the last subsection links the results of this research to the findings of past research.

5.1 Imbalanced dataset

The imbalanced training dataset contains 23.89% churned customers and 76.11% current customers. The imbalanced training data is used to build Decision Trees, Random Forests and Support Vector Machines. This subsection dives into the results of these models. An overview of

the performance of the models that use the imbalanced dataset as input data is given in Table 5. Furthermore, the ROC plots of the tuned models are shown in Figure 5.

5.1.1 Decision Tree - Imbalanced Data

Firstly, the imbalanced input data is used to build a DT using the `rpart` function in R. To tune the `cp`, 10-fold cross validation is used. The tuned DT based on the imbalanced data yields to an out of sample accuracy of 0.91. This indicates that the DT predicts 91% of the observations in the test set correctly as either churned customers or current customers. Furthermore, the recall is 0.934, which indicates that, out of all churned customers in the test set, 93.4% is discovered by the model. When looking at the Matthews Correlation Coefficient (MCC), the DT shows an MCC of 0.723. Within the context of this research, this indicates that the correlation between the actual class and the predicted class (Churn or No Churn) is equal to 0.723. The ROC curve of the DT has an AUC of 0.933. Hence, the AUC of the DT is close to 1, which indicates a good performance in terms of the AUC. However, when looking at the geometric mean of the predicted probabilities, the geometric mean of 0.192 indicates that the variance in the predicted probabilities in the test set is not high. This indicates that the DT predicts a lot of similar probabilities of churning for the observations in the test set.

5.1.2 Random Forest - Imbalanced Data

Secondly, a Random Forest is built using the `RandomForest` package in R. To tune the Random Forest, `mtry` is tuned using 10-fold cross-validation. Using the tuned Random Forest to predict `Churn` in the test set yields to an accuracy of 0.94. Hence, the Random Forest predicts 94% of the observations in the test set correctly as either churned customers or current customers. When looking at the proportion of churned customers discovered by the Random Forest, the recall of 0.976 indicates that 97.6% of the churned customers are correctly classified as churners by the model. Furthermore, the Random Forest yields to an MCC of 0.813 on the test set, indicating a positive correlation of 0.813 between the actual class and the predicted class of the target variable `Churn`. Lastly, the AUC of 0.969 indicates an Area Under the ROC Curve of 0.969 when using the Random Forest to predict churn in the test set. This AUC is very close to 1, which would indicate a high performance of the Random Forest. However, the geometric mean of 0 indicates very little variation in the predicted probabilities of churn for the observations in the test set. This indicates that the Random Forest predicts a lot of similar probabilities of churning for the observations in the test set.

5.1.3 Support Vector Machines - Imbalanced Data

The third modelling approach used to predict `Churn` based on the imbalanced input data is Support Vector Machines, using the `svm` function in R. The SVM model is tuned using 10-fold cross-validation, which shows that a cost parameter of 0.5 leads to the highest 10-fold cross-validated accuracy. This is lower than the default cost of 1, which indicates that the tuned SVM allows for fewer misclassifications than the default SVM. The tuned SVM model yields to an accuracy of 0.855 on the test set, indicating a correct prediction for 85.5% of the observations in the test set. The recall of applying the SVM model to the test set is 0.951. Hence, 95.1% of the churned customers in the test set are discovered by the SVM model. The correlation between the actual class and the predicted class of the test set is 0.522. Lastly, the AUC of 0.912 shows that the Area Under the ROC curve of the tuned SVM is equal to 0.912. With regards to the geometric mean of the position

relative to the hyperplane, the SVM model shows a geometric mean of 0. Similar to the geometric mean of the Random Forest, this indicates very little variation in the predictions on the test set.

5.1.4 Performance Comparison - Imbalanced Data

To summarize, the results show that when building models with the imbalanced dataset as input data, the out of sample performance when predicting `Churn` in the test set differs per model. Table 5 gives an overview of the results obtained on the test set, where the highest values per evaluation metric are shown in blue. The Random Forest model shows the best performance for the first four performance metrics. Hence, using the imbalanced input data, this research shows that a Random Forest with `mtry` equal to 4 achieves the highest out of sample performance in terms of the accuracy, recall, MCC and AUC. This is indicated by an accuracy of 0.94, recall of 0.976, MCC of 0.813 and AUC of 0.969. It should however be noted that the geometric mean of this model is 0. The Random Forest model thus does show a high out of sample performance when predicting churn in the test set, but the variation in the predicted probabilities is very low. When looking at the variation in the predictions, the Decision Tree model shows the highest variation in the predicted probabilities, indicated by a geometric mean of 0.192.

Table 5: Out of sample performance - Imbalanced Data

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - imb (tuned)	0.91	0.934	0.723	0.933	0.192
RF - imb (tuned)	0.94	0.976	0.813	0.969	0
SVM - imb (tuned)	0.855	0.951	0.522	0.912	0

Additionally, Figure 5 shows the ROC plots of the three tuned models. As can be taken from Figure 5, the ROC curve of the Random Forest is the curve on the top left. This indicates that for all possible decision boundaries, the Random Forest has the highest True Positive Rate and the lowest False Positive Rate, compared to the Decision Tree and Support Vector Machine models. Hence, Figure 5 also confirms that the Random Forest model is the best performing model when using the imbalanced dataset as input data. However, the highest geometric mean of 0.192 is shown by the Decision Tree model, which shows an ROC a little below the ROC of the RF. In a business context, this indicates that when using imbalanced input data to build churn prediction models, there is a trade-off between the increase in performance in terms of the first four evaluation metrics for Random Forests, and the higher variance of the predicted probabilities shown by the geometric mean of the Decision Tree model. Furthermore, a Random Forest model is more complex than a Decision Tree, which should also be taken into account when predicting churn in a business context.

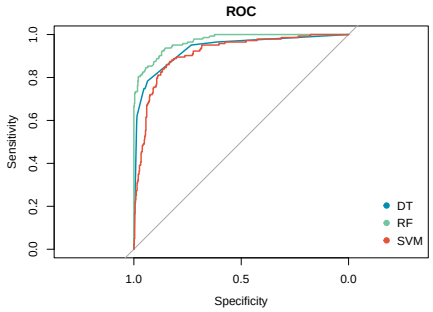

Figure 5: ROC plots of the tuned models - Imbalanced Data

5.2 Random Over-sampled dataset

After applying the Random Over-sampling algorithm to the training data, the Random Over-sampled dataset contains 50.3% churned customers and 49.7% current customers. This Random Over-sampled dataset is used as input for Decision Trees, Random Forests, and Support Vector Machines. An overview of the out of sample performance of these models can be found in Table 6. Besides that, the ROC curves of the tuned models are plotted in Figure 6. It should be noted that the out of sample performance is based on the test set, to which no balancing techniques are applied to ensure an out of sample performance comparison based on real life business data.

5.2.1 Decision Tree - Random Over-Sampled Data

The first modelling approach applied to the Random Over-sampled data is a Decision Tree. Using the tuned DT to predict `Churn` in the test set yields to an out of sample accuracy of 0.783, which indicates that 78.3% of the observations in the test set are correctly classified as either churned customers or current customers. The recall on the test set of 0.96 indicates that out of all churned customers in the test set, 96% is discovered by the model. Furthermore, the MCC of the DT is 0.537. This indicates a positive correlation of 0.537 between the predicted class and the actual class, where the actual class indicates whether the observation is a churned customer or a current customer. Lastly, the Area Under the ROC Curve is equal to 0.824. Additionally, the geometric mean of the predicted probabilities on the test set is 0.315, which indicates that there is variation in the predicted probability of churn in the test set.

5.2.2 Random Forest - Random Over-Sampled Data

Using the Random Over-sampled dataset as input data, two Random Forests are built: a default RF and an RF where `mtry` is tuned. The Random Forest model is tuned by choosing the `mtry` that yields to the highest 10-fold cross-validated accuracy. This shows that the optimal number of variables per split is `3`, which is smaller than the default `mtry` of `4`. The tuned RF is used to predict `Churn` in the test set, to test the out of sample performance of this model. This yields to an accuracy of 0.785, implying that 78.5% of the observations in the test set are predicted correctly as either a churned or current customers. The recall of 0.751 indicates that the tuned RF model predicts 75.1% of the churned customers in the test set correctly. Furthermore, the tuned RF shows an out of sample MCC of 0.553, which indicates a positive correlation of 0.553 between the actual class and the predicted class of the target variable `Churn` in the test set. The Area Under the ROC

Curve, is 0.906. This is close to 1, which indicates a good out of sample performance in terms of the AUC. However, the geometric mean of 0 indicates that there is very little variation in the predicted probabilities of churn in the test set. Hence, the Random Forest based on the Random Over-sampled data shows high performance in terms of the accuracy, recall, MCC and AUC. However, the predicted probabilities of churn show similar predictions for the observations in the test set, which decreases the business value of the model.

5.2.3 Support Vector Machines - Random Over-Sampled Data

The third modelling approach used to predict churn based on the Random Over-sampled input data is Support Vector Machines. The Random Over-sampled data is used to build both a default SVM and a tuned SVM, where the Cost parameter is tuned. Tuning the cost parameter in the SVM model shows that the 10-fold cross-validated accuracy is highest when the Cost parameter is set at 0.1. This is smaller than the default cost of 1, which indicates that the tuned model allows for fewer misclassifications than the default model. The tuned model yields to an out of sample accuracy of 0.811 on the test set. Hence, the tuned SVM model classifies 81.1% of the observations in the test set correctly. Out of all churned customers in the test set, the tuned SVM model discovers 80.2%, which is indicated by the recall of 0.802. With regards to the MCC, the tuned SVM model shows a positive correlation of 0.56 between the actual and predicted classes in the test set. Lastly, the tuned model shows an AUC of 0.905 when applied to predict `Churn` in the test set. The AUC is close to 1, which indicates a high performance in terms of the AUC. However, the geometric mean of 0 indicates a very high similarity for all predictions. This indicates that the SVM is likely to predict the same for the observations in the test set.

5.2.4 Performance Comparison - Random Over-Sampled Data

To summarize, Table 6 shows the results of using Random Over-sampled input data to predict `Churn`, where the highest values per evaluation metric are shown in blue. It shows that the SVM model performs best when looking at the out of sample accuracy and MCC. The highest recall is achieved by Decision Trees, but the AUC shows the highest out of sample value when using Random Forests to predict `Churn`. When looking at the geometric mean of the predictions, the Decision Tree model shows the highest value of 0.315, which indicates that the Decision Tree shows the largest variation in the predicted probabilities of churn for the observations in the test set. The out of sample performance of using Random Over-sampled data to build churn prediction models thus differs per modelling approach.

Hence, in a business context, the most suitable modelling approach depends on the goal of predicting churn, when using Random Over-sampled training data. As shown by the recall and the geometric mean, the Decision Tree model is most suitable when insights into the factors contributing to churn are the goal. Additionally, the Decision Tree provides the largest variation in the predicted probability of churn. On the other hand, when the goal is to predict whether an individual customer is likely to churn, the SVM model shows the highest accuracy and correlation between the actual class and the predicted class.

Additionally, the ROC curves of the tuned models based on the Random Over-sampled input data are shown in Figure 6. The Decision Tree does not yield to a smooth ROC curve. This indicates that before a certain decision boundary, the DT has a relatively high False Positive Rate compared to the True Positive Rate. After that decision boundary, the TPR and the FPR are more balanced towards a high TPR and a low FPR. Furthermore, the AUC of the DT model is 0.824, which is

Table 6: Out of sample performance - Random Over-sampled Data

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - ROS (tuned)	0.783	0.96	0.537	0.824	0.315
RF - ROS (tuned)	0.785	0.751	0.553	0.906	0
SVM - ROS (tuned)	0.811	0.802	0.56	0.905	0

lower than the AUC of the RF and the SVM models. However, the geometric mean of the predicted probabilities on the test set is not dependent on the decision boundaries, and it shows the highest value for the Decision Tree models. Hence, the AUC combined with the geometric mean of the test set predictions shows that Decision Trees are the most suitable modelling approach when using Random Over-sampled input data to predict churn.

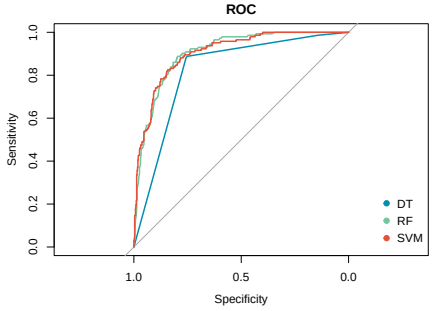

Figure 6: ROC plots of the tuned models - Random Over-sampled Data

5.3 SMOTE data

The training dataset that is balanced using the SMOTE algorithm contains 42.9% churned customers and 57.1% current customers. This dataset is used as input for building DTs, Random Forests and Support Vector Machines. For an overview of the out of sample performance based on the test set, we refer to Table 7. Furthermore, Figure 7 shows the ROC plots for the optimal, tuned, DT, RF, and SVM. It should be noted that for evaluating the out of sample performance in a real life business setting, no balancing techniques are applied to the test set.

5.3.1 Decision Tree - SMOTE Data

Firstly, Decision Trees are used as a modelling approach to predict `Churn` based on the SMOTE data. The DT is pruned using 10-fold cross-validation to find the optimal `cp`. The tuned DT based on the SMOTE dataset uses the same `cp` as the default `cp`: 0.01. This DT yields to an accuracy of 0.876 when the model is used to predict `Churn` in the test set. This indicates that 87.6% of the observations in the test set is correctly classified as a churned or current customer. Table 7 shows a recall of 0.947 on the test set. Hence, 94.7% of the churned customers in the test set are indeed classified as churners by the DT. Furthermore, the out of sample MCC of the DT is 0.659. This indicates a positive correlation of 0.659 between the predicted `Churn` and the actual `Churn`. Lastly, the Area Under the ROC Curve of the test set predictions is equal to 0.89. Looking at the variation in the predicted probabilities, the geometric mean of 0.257 shows that there is some variation in the predicted probabilities of churn in the test set, but the variation is not very large.

5.3.2 Random Forest - SMOTE Data

Secondly, Random Forests are built using the SMOTE data as input data. Tuning the `mtry` using 10-fold cross-validation indicates that the out of bag error is lowest when 3 random variables are chosen. As can be taken from Table 7, the tuned model has an accuracy of 0.901. Hence, 90.1% of the observations in the test set is classified correctly by the tuned RF. The recall of the tuned model is 0.912, which implies that 91.2% of the churned customers in the test set is indeed predicted as a churned customer by the RF. With regards to the MCC, the tuned model shows a positive correlation of 0.727 between the actual `Churn` class and the predicted `Churn` class in the test set. Lastly, the Area Under the ROC Curve of the tuned RF is 0.965. This is a high AUC, which indicates good performance in terms of the AUC. However, the geometric mean of the predicted probabilties of churn in the test set is 0, which indicates very little variation in the predicted probabilities of churn in the test set. Hence, this indicates that the Random Forests predicts similar probabilities for the observations in the test set.

5.3.3 Support Vector Machines - SMOTE Data

Support Vector Machines is the third and last modelling approach using the SMOTE input dataset. When looking at the out of sample accuracy of the tuned SVM, Table 7 shows that 86.4% of the observations in the test set is predicted correctly. Furthermore, the tuned SVM model discovers 89.7% of all the churned customers in the test data, indicated by the recall of 0.897. The correlation between the actual `Churn` class and the predicted `Churn` class is 0.612 when using the tuned model to predict `Churn` in the test set. Lastly, the Area Under the ROC Curve of the test set is 0.913. However, there is a high similarity in the predictions by the SVM model, which is indicated by the geometric mean of 0. This indicates that the SVM model predicts a lot of similar relative positions to the hyperplane when predicting churn in the test set.

5.3.4 Performance comparison - SMOTE Data

To summarize, Table 7 shows the results of using the SMOTE balanced dataset as input data to predict `Churn` in the test set, where the highest values per evaluation metric are shown in blue. The performance measures show that the best performing model in terms of the out of sample accuracy, MCC, and AUC is the Random Forest model. However, regarding the recall and the geometric mean, the Decision Tree model shows the highest performance.

Hence, when using the SMOTE algorithm to balance the dataset, using Random Forests as a modelling approach shows the best out of sample performance for three of the five evaluation metrics: the accuracy (0.901), MCC (0.727), and the AUC (0.965). On the contrary, Decision Trees perform best regarding the recall(0.947) and the geometric mean (0.257). This indicates that the increase in performance of a Random Forest compared to a Decision Tree does come at the cost of the proportion of churned customer that are discovered in the test set (recall) and the variation in the predicted probabilities of churn in the test set (geometric mean). Therefore, in a business context, both Random Forests and Decision Trees based on SMOTE input data are suitable, but the most suitable modelling approach is dependent on the goal of predicting churn. In case the prediction model is used to gain insights on the factors contributing to churn, Decision Trees are more suitable. On the other hand, if the model is used to predict the probability of churn for separate customers, then the Random Forest model shows a higher performance on three of the five performance metrics.

Figure 7 shows the ROC curves of the tuned models. This shows that the ROC curve of the Random Forest model is closest to the top left of the space, which indicates that the RF model has

Table 7: Out of sample performance - SMOTE Data

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - SMOTE (tuned)	0.876	0.947	0.659	0.89	0.257
RF - SMOTE (tuned)	0.901	0.912	0.727	0.965	0
SVM - SMOTE (tuned)	0.864	0.897	0.612	0.913	0

the highest TPR and the lowest FPR when using a range of decision boundaries. Hence, this also confirms the suitability of using Random Forests when the input data is balanced using the SMOTE algorithm. However, the low variation in the predicted probabilities is a potential disadvantage when using Random Forests based on SMOTE training data to predict churn.

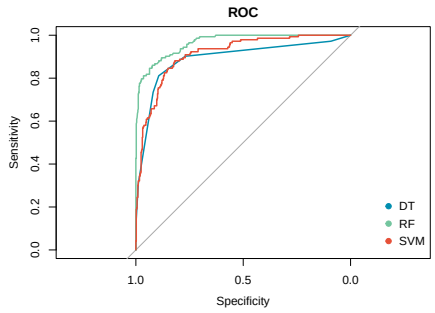

Figure 7: ROC plots of the tuned models - SMOTE Data

5.4 ADASYN data

After applying the ADASYN algorithm to the training data, the ADASYN balanced data consists of 50% churned customers and 50% current customers. The ADASYN balanced dataset serves as input for building DTs, Random Forests, and Support Vector Machines. An overview of the out of sample performance is given in Table 8. Furthermore, the ROC plots of the tuned DT, RF, and SVM are shown in Figure 8. It should be noted that the out of sample performance is evaluated by predicting churn in the test set, to which no balancing techniques are applied to simulate the performance in real life business data.

5.4.1 Decision Tree - ADASYN Data

The first modelling approach applied to the ADASYN balanced dataset is Decision Trees.Performing 10-fold cross-validation shows that a `cp` of 0.01 yields to the lowest sum of the cross-validated x-error and x-std. This is equal to the default `cp`. The tuned DT yields to an out of sample accuracy of 0.863, indicating a correct `Churn` prediction for 86.3% of the observations in the test set. The recall of 0.944 indicates that out of all churned customers in the test set, 94.4% is discovered by the DT. The correlation between the actual `Churn` class and the predicted `Churn` class in the test set is 0.631, as shown by the out of sample MCC. Lastly, the Area Under the Curve of the DT is 0.89. In addition to the AUC, the geometric mean of 0.18 shows that there is little variation in the predicted probabilities of churn in the test set.

5.4.2 Random Forest - ADASYN Data

Secondly, Random Forests are used to predict `Churn` based on the ADASYN balanced data. Because of the created dummies for each category, the ADASYN dataset contains 44 input variables. Hence, the default `mtry` for the RF is 6. Tuning the RF using 10-fold cross-validation indicates that the mtry that leads to the lowest 10-fold cross-validated out of bag error is 13, which is larger than the default `mtry` of 6. This indicates that the tuned RF randomly selects more variables per split than the default RF. Using the tuned RF to predict `Churn` in the test set yields to an out of sample accuracy of 0.919, which indicatest a correct churn prediction for 91.9% of the observations in the test set. The recall of 0.951, indicates that 95.1% of the churned customers in the test set are discovered by the RF, and the MCC of 0.754 indicates a positive correlation between the actual churn class and the predicted churn class in the test set. With regards to the Area Under the ROC curve, the Random Forest shows that when predicting churn in the test set, the AUC is 0.967. This indicates a high performance in terms of the AUC. However, the geometric mean of 0 indicates that there is very little variation in the predicted probabilities of churn in the test set.

5.4.3 Support Vector Machines - ADASYN Data

As a third modelling approach, Support Vector Machines are built using the ADASYN balanced input data, where the cost parameter is tuned using 10-fold cross-validation. Predicting `Churn` in the test set using the tuned SVM leads to an out of sample accuracy of 0.82, which indicates that the model provides a correct prediction for 82% of the observations in the test set. Additionally, out of all 143 churned customers in the test data, 81.3% is predicted as a churned customer by the SVM, indicated by the recall of 0.813. Furthermore, the correlation between the actual `Churn` class and the predicted `Churn` class is 0.574, which is indicated by the MCC. Lastly, the ROC curve of the SVM shows an Area Under the Curve of 0.9. This is close to 1, which indicates a good performance in terms of the AUC. However, the geometric mean of the predicted relative positions in the test set is 0, which indicates very little variation in the predictions in the test set. Hence, the SVM model predicts similar relative positions to the hyperplane in the test set.

5.4.4 Performance comparison - ADASYN Data

To summarize, Table 8 shows the out of sample performance of all `Churn` prediction models that used the ADASYN balanced data as input data, where the highest values per evaluation metric are shown in blue. As can be taken from this table, using Random Forests to predict `Churn` yields to the highest out of sample performance when ADASYN balanced data is used as input data, for four of the five evaluation metrics. The Random Forest model yields to the highest out of sample accuracy (0.919), recall (0.951), MCC (0.754), and AUC (0.967). However, as can be taken from Table 8, the geometric mean of the predicted probabilities of churn in the test set is 0, which indicates very little variation in the predictions in the test set. When looking at the geometric mean, the Decision Tree model based on the ADASYN balanced input data yields to the highest variation in the predicted probabilities when predicting churn on the test set. The geometric mean of 0.18 indicates that there is more variation in the predicted probabilities of churn on the test set when using the DT to predict churn. In a business context, this indicates that the Random Forest model shows the best predictive out of sample performance, but there is a trade-off between the predictive performance and the variation in the predictions, when using the models to get insights into the factors contributing to churn.

Table 8: Out of sample performance - ADASYN Data

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - ADASYN (tuned)	0.863	0.944	0.631	0.89	0.18
RF - ADASYN (tuned)	0.919	0.951	0.754	0.967	0
SVM - ADASYN (tuned)	0.82	0.813	0.574	0.9	0

In addition to the performance metrics shown in Table 8, the ROC curves of the tuned models based on the ADASYN input data are plotted in Figure 8. The ROC curves of the DT and the SVM model show the worst performance in balancing the FPR and the TPR, because an ROC curve close to the top left of the ROC plot is desired. On the contrary, the ROC curve of the Random Forest is the curve closest to the top left of the ROC space. This confirms that the RF model is the best performing model, because this indicates that the RF curve has a low FPR and a high TPR when using a range of varying decision boundaries. However, it should be noted that the geometric mean of 0 indicates very little variation in the predicted probabilities of churn.

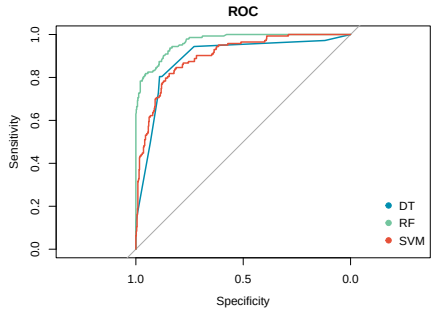

Figure 8: ROC plots of the tuned models - ADASYN Data

5.5 Overall Performance Comparison and Discussion

In past research, the first applied balancing technique, Random Over-sampling, was found to increase the performance of churn prediction models (Gui, 2017; Japkowicz, 2000). The SMOTE algorithm, a more advanced balancing technique, is one of the most often mentioned balancing techniques in churn prediction literature. This algorithm generates synthetic new observations in the minority class, which decreases the risk of overfitting compared to Random Over-sampling. Past research showed an increase in the predictive performance of multiple modelling approaches, when SMOTE balanced input data was used (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017). The third balancing technique applied is the ADASYN algorithm, which focuses more on the complex observations in the dataset (Amin et al., 2016). An increased predictive performance of using the ADASYN balanced input data compared to the SMOTE algorithm was found by He et al. (2008). This was however not confirmed by Amin et al. (2016).

Past research thus shows some advantages of using balancing techniques to handle the class imbalance problem on an external level. However, possible disadvantages of using balancing techniques are also discussed. Random Over-sampling does not create additional information and thus induces the risk of overfitting to the training data (Chawla, 2009; Weiss, 2004). The SMOTE algorithm ignores the majority class instances that are close to the minority observations (Wang et

al., 2021), and the ADASYN algorithm was not always found to increase the predictive performance (Amin et al., 2016).

In this research, the 12 created models are evaluated on their out of sample performance by predicting `Churn` in the test set. It should be noted that no balancing techniques are applied to the test set to ensure valid out of sample performance evaluation based on real-life business data. Predicting `Churn` in the test set leads to five evaluation metrics: the accuracy, the recall, the Matthews Correlation Coefficient (MCC), the Area Under the ROC Curve (AUC), and the geometric mean. The first four evaluation metrics range from 0 to 1 and have a positive relationship with the predictive performance. Hence, for each of these performance metric, a value closer to 1 indicates a better out of sample performance in terms of that evaluation metric. The last performance metric, the geometric mean, indicates the variation in the predictions. A small geometric mean indicates little variation in the predictions and vice versa. In the context of this research, where the class imbalance problem is researched, a larger variation in the predicted probabilities is desired. This is because the class imbalance problem generally induces misleading results, related to an underestimation of the probability of belonging to the minority class (churn). An overview of the predictive performance of the 12 models is given in Table 9, where the highest value per evaluation metric is shown in blue.

Firstly, using Decision Trees as a modelling approach, Table 9 shows that the imbalanced input data yields to the highest performance in terms of accuracy, MCC, and AUC. Hence, considering the out of sample performance in terms of these three evaluation metrics, imbalanced input data yields to the highest out of sample performance on the test set, when using Decision Trees to predict `Churn`. This is not in line with previous research, where it was found that applying balancing techniques to churn data increases the predictive performance of Decision Tree models (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017).

Further comparing the performance of the applied balancing techniques shows that, when using Decision Trees, the SMOTE algorithm has a higher MCC than ADASYN, but it has an equal AUC. Past research found an increased performance for ADASYN compared to SMOTE (He et al., 2008). This research does not confirm an increased out of sample performance when using ADASYN compared to using SMOTE, which aligns with the findings of Amin et al. (2016).

However, looking at the overall performance of Decision Trees, this research shows that not applying any balancing techniques to the input data yields to the highest out of sample performance in terms of the accuracy, the MCC and the AUC. This research thus shows that neither Random Over-sampling, nor SMOTE, nor ADASYN increases the predictive performance of churn prediction models using Decision Trees. However, Table 9 does show in increase in both the recall and the variation of the predictions when using Random Over-sampled input data, compared to the imbalanced data. This is shown by the increased recall of 0.96 and the increased geometric mean of 0.315.

In a business context, the results of using a Decision Tree to predict churn show that imbalanced input data yields to the highest predictive performance when the accuracy, MCC and AUC are chosen as important evaluation metrics. Hence, when using a Decision Tree model to predict whether an individual customer has a high probability of churning, this research does not show added value of applying balancing techniques to the training data. However, using Random Over-sampled data leads to a higher percentage of churners discovered by the model (recall) and a higher variation in the predicted probabilities of churn (geometric mean). This indicates that this research found that Random Over-sampling provides added value when using Decision Trees to gain overall insights into the factors contributing to churn.

Secondly, when using Random Forests as a modelling approach, Table 9 shows that the

performance in terms of the first four evaluation metrics is highest when using the imbalanced data as input data for the Random Forest. This indicates that using imbalanced input data yields to the highest out of sample performance when using Random Forests to predict `Churn`. Further investigating the out of sample performance of the Random Forests that are based on the three balanced input datasets, Table 9 shows that the ADASYN balanced data yields to a higher out of sample performance compared to SMOTE and Random Over-sampling. Hence, when using Random Forests to predict `Churn`, this research found that ADASYN outperforms both SMOTE and Random Over-sampling, which aligns with the findings of He et al. (2008). However, looking at the general out of sample performance, this research shows that not applying any balancing techniques to the input data yields to the highest out of sample performance when using Random Forests to predict `Churn`. Hence, using imbalanced input data is the most suitable for predicting `Churn` using Random Forests. This research thus does not confirm added value of applying balancing techniques to churn data. This contradicts various past research, where it was stated that the use of balancing techniques improves the performance of churn prediction models (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017; Rahman & Kumar, 2020)

On the contrary, the geometric mean shows to be 0 for all four input datasets. This indicates that although Random Forests show a high predictive performance in terms of the accuracy, recall, MCC and AUC, the variation in the predictions is very low. This contradicts previous research, where it was stated that Random Forests increase the variation in the predictions compared to Decision Trees (Breiman, 2001). In a business context, this indicates that Random Forests are a suitable method for predicting churn on the individual customer level, but this does not apply when the goal of modelling churn is to gain overall insights into the factors contributing to churn. Additionally, interpreting Random Forests requires more advanced interpretation techniques, because of the black-box nature of this ensemble method.

The two aforementioned modelling approaches, Decision Trees and Random Forests, both show that using imbalanced input data yields to the highest out of sample performance on the test set in terms of the accuracy, MCC, and AUC. Hence, for these two modelling approaches, there is little added value of applying balancing techniques to the churn data regarding the out of sample performance. The third modelling approach, Support Vector Machines, shows a different result. SVM based on the imbalanced dataset shows the highest performance in terms of recall. However, for all three other performance metrics (accuracy, MCC and AUC), the SVM built using the SMOTE input data shows the highest out of sample performance. Therefore, when using Support Vector Machines to predict `Churn`, this research finds an increased performance of using SMOTE compared to imbalanced data, Random Over-sampled data and ADASYN data.

This aligns with the findings of multiple past research papers where it was stated that applying the SMOTE algorithm increases the out of sample performance (Amin et al., 2016; Gui, 2017; Miguéis et al., 2017). It does however contradict He et al. (2008), who stated that ADASYN has an increased performance compared to SMOTE. With regards to the variation in the predictions, neither of the input datasets yield to a geometric mean higher than 0. This indicates that SVM based on SMOTE input data is suitable for predicting churn on the individual level, but similar to Random Forests, gaining overall insights into the factors contributing to churn is harder when using SVM as a modelling approach. This is due both the low variation in the predictions, as well as the black-box nature of SVM, which requires advanced interpretation methods to gain insights into the effects of individual variables.

Looking at the overall performance of all the models, Table 9 shows that the Random Forest based on the imbalanced input data has the highest performance on the test set in terms of the accuracy,

recall, MCC and AUC. Hence, this research shows that predicting `Churn` leads to the highest out of sample performance when using a Random Forests, built on imbalanced input data. However, when looking at the variation in the predictions, the highest geometric mean (0.315) is achieved by the Decision Tree model based on Random Over-sampled input data. This contradicts the findings of Breiman (2001), who stated that Random Forests increase the variation in the predictions, compared to Decision Trees.

To summarize, this research shows that the imbalanced input data yields to the highest out of sample performance when predicting `Churn` on the test set, to which no balancing techniques are applied. In the context of this research, the use of (advanced) balancing techniques in the training data thus does not provide added value in terms of the predictive performance of churn prediction models. More specifically, as Table 9 shows, the Random Forest model using the imbalanced data shows the highest out of sample performance in terms of four evaluation metrics. In the context of this research, the highest accuracy (0.94), recall (0.976), Matthews Correlation Coefficient (0.813) and AUC (0.969) are thus achieved by using a Random Forest with `mtry` equal to `3`, combined with the imbalanced training data as input data.

In a business context, this research shows that a Random Forest based on the imbalanced input data is the most suitable combination when predicting churn for individual customers. However, the variation in the predictions is higher for Decision Trees, which indicates that Decision Trees are more suitable when overall insights into the factors contributing to churn are desired. Additionally, Random Forests are an ensemble method, that require advanced interpretation methods to gain insights into the factors contributing to churn. On the other hand, Decision Trees are highly interpretable and do not require extensive knowledge of machine learning (Coussement & Van den Poel, 2008; Keramati et al., 2014; Shaaban et al., 2012). Hence, the increased out of sample performance of Random Forests does come at the cost of lower interpretability, a lower variation in the predictions, and a more complex implementation, compared to Decision Trees.

Table 9: Overview results

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - imb (tuned)	0.91	0.934	0.723	0.933	0.192
DT - ROS (tuned)	0.783	0.96	0.537	0.824	0.315
DT - SMOTE (tuned)	0.876	0.947	0.659	0.89	0.257
DT - ADASYN (tuned)	0.863	0.944	0.631	0.89	0.18
RF - imb (tuned)	0.94	0.976	0.813	0.969	0
RF - ROS (tuned)	0.785	0.751	0.553	0.906	0
RF - SMOTE (tuned)	0.901	0.912	0.727	0.965	0
RF - ADASYN (tuned)	0.919	0.951	0.754	0.967	0
SVM - imb (tuned)	0.855	0.951	0.522	0.912	0
SVM - ROS (tuned)	0.811	0.802	0.56	0.905	0
SVM - SMOTE (tuned)	0.864	0.897	0.612	0.913	0
SVM - ADASYN (tuned)	0.82	0.813	0.574	0.9	0

6 Conclusion

Churn is often known as a rare event which induces a class imbalance problem that needs to be taken into account when building churn prediction models. If not handled correctly, the class imbalance problem can induce misleading results in churn prediction models (Amin et al., 2016). This research investigates the performance of handling the class imbalance problem on the external level by creating additional observations in the minority class (Churn), using over-sampling-based balancing techniques on the training data. Three balancing techniques are applied to the training data, which leads to four input datasets: the imbalanced dataset, the Random Over-sampled dataset, the SMOTE balanced dataset, and the ADASYN balanced datasets. These four datasets are used as input data for Decision Trees, Random Forests and Support Vector Machines.
This research contributes to the existing churn prediction literature by investigating and comparing the predictive performance of churn prediction models, combined with three balancing techniques applied in the pre-processing stage. Hence, the research question investigated in this research is:

(1) What is the value added of using advanced modelling approaches for churn prediction and (2) how does the use of balancing techniques influence the performance?

The remainder of this section is structured as follows. The first subsection discusses the main findings of this research. The second subsection discusses the limitations of this research, accompanied by suggestions for further research.

6.1 Main findings

The main findings of this research are split up based on the two parts of the research question. First, the added value of using advanced modelling approaches is discussed. Secondly, the added value of using balancing techniques is discussed. Regarding the out of sample performance of the modelling approaches, the results of this research show that the best performing model for churn prediction modelling is a Random Forest model. The second-best performing model is the Decision Tree model and the Support Vector Machine model shows the lowest out of sample performance. This out of sample performance is based on the churn predictions in the test set, to which no balancing techniques are applied to simulate the out of sample performance in a real life business setting.

With regards to the first part of the research question, this research thus shows that the use of Random Forests yields to improved out of sample performance for predicting churn, compared to using Decision Trees. This aligns with the findings of Rahman & Kumar (2020), who found that Random Forests are the best performing model when predicting churn. On the contrary, this research does not show an increased out of sample performance of using Support Vector Machines to predict churn, compared to Decision Trees.

Looking further into the influence on the out of sample performance of applying balancing techniques to the input data, this research does not show an increased out of sample performance after applying balancing techniques to the training data when predicting churn using Random Forests. The use of the original, imbalanced input data yields to the highest out of sample performance in terms of the accuracy, recall, MCC and AUC. When using Decision Trees to predict churn, the imbalanced data shows the highest out of sample performance in terms of the accuracy, MCC and AUC. Lastly, using SVM, the imbalanced data yields to the highest out of sample recall. Hence, this indicates that no increase performance was found in terms of the accuracy, recall, Matthews Correlation Coefficient and Area Under the ROC Curve, when applying balancing techniques to the training data. This is in line with a recent study by Goorbergh et al. (2022). They showed that

the use of balancing techniques induces overestimation of the probability of being in the minority class in a clinical context. Hence, this research confirms the findings of Goorbergh et al. (2022) and extends this in the context of churn prediction modelling in a subscription-based B2B company.

Combining the results of both parts of the research question, this research thus shows that Random Forests based on the original, imbalanced input data yield to the highest predictive performance in terms of the accuracy, recall, Matthew Correlation Coefficient and the Area Under the ROC Curve. Hence, the more advanced modelling approach Random Forests does show added value over the use of the more basic modelling approach, Decision Trees. Furthermore, this research shows that applying balancing techniques to the training data generally does not yield to an increased predictive out of sample performance. The predictive performance of the best performing model in this research, a Random Forest based on imbalanced input data, is an accuracy of 0.94, a recall of 0.976, a Matthews Correlation Coefficient of 0.813 and an AUC of 0.969.

However, this research also evaluates the out of sample performance of the models based on the geometric mean, which indicates the variation in the predictions. This research shows that using Decision Trees to predict churn yields to the highest variation in predictions, which is indicated by the largest geometric mean compared to Random Forests and Support Vector Machines. In a business context, this means that Decision Trees provide more insights into the overall factors contributing to churn, compared to the more advanced Random Forest and Support Vector Machines. Additionally, this research is performed in the context of a B2B subscription-based company. Therefore, the interpretability and ease of use of modelling approaches are important secondary criteria. Decision Trees are known to be highly interpretable and easy to use, while Random Forests and Support Vector Machines are harder to interpret and harder to implement. Hence, businesses should be aware that the increased performance in terms of the accuracy, recall, MCC and AUC when using the more advanced Random Forest does come at the cost of the interpretability, the variation in the predictions and the ease of use, compared to using Decision Trees.

This research uses a dataset provided by TOPdesk Nederland B.V., which contains 23.61% churned customers. Hence, if there are 100 customers, without using churn prediction models, we would expect 23 of these customers to churn. This research shows that B2B subscription-based companies can use a Random Forest built using the original training data as input to predict churn. Applying the Random Forest model to these 100 customers predicts 94 of these customers correct as either customers that will churn, or customers that will stay. Assuming that indeed 23% of the customers will churn, this research can correctly identify 97.6% (22) of these customers as potential churners.

Additionally, this research uses Decision Trees, which create visual representations of decision rules. This research shows that Decision Trees are suitable to gain overall insights into which type of customers are most likely to churn. For example, more resources can be allocated to a certain customer group if the decision path shows that customers with certain characteristics have a high probability of churning.

In practice, this research thus recommends subscription-based B2B companies not to apply balancing techniques to the training data when predicting churn. Furthermore, this research recommends Random Forests as a modelling approach to predict churn in a B2B context. However, further interpretation techniques are required to interpret the Random Forests. Therefore, if the company requires an interpretable churn prediction model, this research shows that Decision Trees are an interpretable alternative with more variation in the predicted probabilities of churn.

6.2 Limitations and Future Research

This research is performed using data from a B2B company that sells subscription-based software. The research is performed carefully, and I am confident about the results in the context of the current dataset. However, trade-offs and choices are inevitable when performing research. This leads to limitations and at the same time, this provides openings for further research. This subsection addresses some limitations and provides suggestions for future research. Firstly, the generalizability of this research is questionable. The performance comparison of both the balancing techniques and the modelling approaches is based on a dataset from one data source. The data originates from one subscription-based B2B company: TOPdesk Nederland B.V. Hence, to increase the generalizability of the results of this research in the context of subscription-based B2B companies, further research should include a performance comparison based on churn data from multiple subscription-based B2B companies. Additionally, further research could include usage data to predict churn, which was not included in the input data of this research. Ascarza & Hardie (2013) showed that modelling customer churn based on the usage behaviour of customers can help companies to segment their customers based on their likelihood to churn, and identify the most common patterns that customers show before churning. This research can be performed in a B2B subscription-based company to validate their findings in the context of B2B subscription-based companies.

Secondly, this research found that applying Random Over-sampling, SMOTE or ADASYN in the data pre-processing stage do not increase the predictive performance of churn prediction models. This is generalized to the conclusion that the use of balancing techniques does not yield to higher out of sample performance in the context of churn prediction modelling. However, past research evaluated more balancing techniques, which are not included in this research. For instance, Amin et al. (2016) found an increase in predictive performance when the Mega-Trend Diffusion Function (MTDF) is used as an over-sampling technique. This over-sampling technique is not considered in this research, but it should be incorporated in further research, to investigate whether the use of MTDF increases the predictive performance of churn prediction models.

Furthermore, this research is based on cross-sectional data. This indicates that this research does not investigate the changes in the predictive performance of churn prediction models over time. The dataset provided has a limited timeframe, and all the observations in the training data are used to build the models. Hence, no distinctions are made between periods in time. In previous research, Risselada et al. (2010) showed that the predictive power of churn models declines after a certain period. This indicates that using churn prediction models to forecast only provides an accurate forecast up until that point in time. Risselada et al. (2010) performed their research in the context of an internet service provider and a health insurance company. On the other hand, this research is performed in the context of a subscription-based B2B company. Hence, further research on the predictive power of churn prediction models over time should be performed in the context of subscription-based B2B companies, to investigate the forecasting power of churn prediction models in a broader context than the context of Risselada et al. (2010)'s research. For this, longitudinal data should be used.

Additionally, this research addresses the class imbalance problem on an external level. This indicates that the data is balanced in the data pre-processing stage. Another way to handle the class imbalance problem is by addressing this on the internal, algorithm, level. Past research used for example a One-Class Support Vector Classifier to predict churn (Zhao et al., 2005). Furthermore, Quantile Random Forests have been applied to classify imbalanced data (O'Brien & Ishwaran, 2019). Further research could compare the performance of handling the class imbalance on the external level with the performance of handling the class imbalance problem using, for example, One-Class Support

Vector Classifiers and Quantile Random Forests.

Lastly, this research does not investigate possible differences in the reasons to churn. Previous research found that there are differences in the interaction with marketing communication between silent and overt churners (Ascarza et al., 2018). Hence, further research could investigate these possible differences in the context of B2B subscription-based companies, to investigate whether the marketing interactions differ across the different types of churners.

7 Bibliography

Ahmed, A., & Linen, D. M. (2017). A review and analysis of churn prediction methods for customer retention in telecom industries. *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1–7.

Alshurideh, M. T. (2016). Is customer retention beneficial for customers: A conceptual background. *Journal of Research in Marketing*, *5*(3), 382–389.

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., Hawalah, A., & Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access*, *4*, 7940–7957.

Ascarza, E., & Hardie, B. G. (2013). A joint model of usage and churn in contractual settings. *Marketing Science*, *32*(4), 570–590.

Ascarza, E., Netzer, O., & Hardie, B. G. (2018). Some customers would rather leave without saying goodbye. *Marketing Science*, *37*(1), 54–77.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *Classification and regression trees (CRC, boca raton, FL)*.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626–4636.

Çelik, O., & Osmanoglu, U. O. (2019). Comparing to techniques used in customer churn analysis. *Journal of Multidisciplinary Developments*, *4*(1), 30–38.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. *Data Mining and Knowledge Discovery Handbook*, 875–886.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 1–13.

Coussement, K., Benoit, D. F., & Van den Poel, D. (2010). Improved marketing decision making in a customer churn prediction context using generalized additive models. *Expert Systems with Applications*, *37*(3), 2132–2143.

Coussement, K., Lessmann, S., & Verstraeten, G. (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. *Decision Support Systems*, *95*, 27–36.

Coussement, K., & Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, *34*(1), 313–327.

Danesh, S. N., Nasab, S. A., & Ling, K. C. (2012). The study of customer satisfaction, customer trust and switching barriers on customer retention in malaysia hypermarkets. *International Journal of Business and Management*, *7*(7), 141–150.

Daskalaki, S., Kopanas, I., & Avouris, N. (2006). Evaluation of classifiers for an uneven class distribution problem. *Applied Artificial Intelligence*, *20*(5), 381–417.

Deville, J.-C., & Tillé, Y. (2004). Efficient balanced sampling: The cube method. *Biometrika*, *91*(4), 893–912.

Dingli, A., Marmara, V., & Fournier, N. S. (2017). Comparison of deep learning algorithms to

predict customer churn within a local retail industry. *International Journal of Machine Learning and Computing*, *7*(5), 128–132.

Fader, P. S., & Hardie, B. G. (2007). How to project customer retention. *Journal of Interactive Marketing*, *21*(1), 76–90.

Ferreira, J., Vellasco, M. M., Pacheco, M. A. C., Carlos, R., & Barbosa, H. (2004). Data mining techniques on the evaluation of wireless churn. *ESANN*, *28*, 483–488.

Goorbergh, R. van den, Smeden, M. van, Timmerman, D., & Van Calster, B. (2022). The harm of class imbalance corrections for risk prediction models: Illustration and simulation using logistic regression. *arXiv Preprint arXiv:2202.09101*.

Gordini, N., & Veglio, V. (2017). 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*, *62*, 100–107.

Gui, C. (2017). Analysis of imbalanced data set problem: The case of churn prediction for telecommunication. *Artif. Intell. Res.*, *6*(2), 93.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328.

Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, *39*(1), 1414–1425.

Iden, J., & Eikebrokk, T. R. (2013). Implementing IT service management: A systematic literature review. *International Journal of Information Management*, *33*(3), 512–523.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. *Proc. Of the Int'l Conf. On Artificial Intelligence*, *56*, 111–117.

Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., & Abbasi, U. (2014). Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing*, *24*, 994–1012.

Kirui, C., Hong, L., Cheruiyot, W., & Kirui, H. (2013). Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining. *International Journal of Computer Science Issues (IJCSI)*, *10*(2 Part 1), 165.

Larivière, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, *29*(2), 472–484.

Lazear, E. P., & Spletzer, J. R. (2012). Hiring, churn, and the business cycle. *American Economic Review*, *102*(3), 575–579.

Li, K.-L., Huang, H.-K., Tian, S.-F., & Xu, W. (2003). Improving one-class SVM for anomaly detection. *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03ex693)*, *5*, 3077–3081.

Ling, R., & Yen, D. C. (2001). Customer relationship management: An analysis framework and implementation strategies. *Journal of Computer Information Systems*, *41*(3), 82–97.

Miguéis, V. L., Camanho, A. S., & Borges, J. (2017). Predicting direct marketing response in banking: Comparison of class imbalance methods. *Service Business*, *11*(4), 831–849.

Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., & Mason, C. H. (2006). Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, *43*(2), 204–211.

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

Nguyen, N. N., & Duong, A. T. (2021). Comparison of two main approaches for handling imbalanced data in churn prediction problem [j]. *Journal of Advances in Information Technology*, *12*(1).

O'Brien, R., & Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern Recognition*, *90*, 232–249.

Probst, P., & Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *J. Mach. Learn. Res.*, *18*(1), 6673–6690.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), e1301.

Rahman, M., & Kumar, V. (2020). Machine learning based customer churn prediction in banking. *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1196–1201.

Reichheld, F. (2001). Prescription for cutting costs. *Harvard Business School Publishing*.

Risselada, H., Verhoef, P. C., & Bijmolt, T. H. (2010). Staying power of churn prediction models. *Journal of Interactive Marketing*, *24*(3), 198–208.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471.

Shaaban, E., Helmy, Y., Khedr, A., & Nasr, M. (2012). A proposed churn prediction model. *International Journal of Engineering Research and Applications*, *2*(4), 693–697.

Umayaparvathi, V., & Iyakutti, K. (2016). A survey on customer churn prediction in telecom industry: Datasets, methods and metrics. *International Research Journal of Engineering and Technology (IRJET)*, *3*(04).

Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, *55*, 1–9.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, *218*(1), 211–229.

Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, *38*(3), 2354–2364.

Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of classification methods on unbalanced data sets. *IEEE Access*, *9*, 64606–64628.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *ACM Sigkdd Explorations Newsletter*, *6*(1), 7–19.

Zhao, Y., Li, B., Li, X., Liu, W., & Ren, S. (2005). Customer churn prediction using improved one-class support vector machine. *International Conference on Advanced Data Mining and Applications*, 300–306.

Appendix

Appendix A: Variable description

Numeric variables

- license_bracket: TOPdesk sells its' software to customers with a license bracket. The license bracket indicates the number of active users (operators or end-users) that are allowed to use the software.
- age_months: the total number of months that the (former) customer has (or had) an active contract. Hence, this indicates the customer lifetime.
- days_since_last_consultancy: the number of days since the start of the most recent consultancy process. This indicates an investment from the customer to either implement or optimize the use of their TOPdesk software.
- total_investments_eur: the total number of euro's that the (former) customer invested in consultancy, trainings or other services that TOPdesk offers for the customers.
- total_tickets: the total number of support tickets that a customer made in their total customer lifetime.
- td_customer_arr_eur: the total number of Annual Recurring Revenue in euro's. This indicates the amount of money a customer pays (or paid) yearly for their most recent subscription to TOPdesk's software.
- invoice_time: the time between the send date of the most recently paid invoice and the payment date of this invoice.

Categorical variables

- td_customer_turnover_type: the type of subscription that the (former) customer has. Can be either SaaS or On Premise (indicated by either "Subscription" or "Maintenance").
- td_customer_product_line: the product line used by the (former) customer.
- td_customer_product_generation: the version of the software that the (former) customer uses.
- business_unit: the customer group of the customer is indicates by the business unit, because TOPdesk serves its' customers using different business units.
- account_type: the level of resources needed to manage the (former) customer.
- invoice_frequency: the invoice frequency that the customer chose to pay their annual fee. Either yearly, half-yearly, quarterly or monthly.
- long_contract: whether the customer signed a contract for more than 36 months (1) or less than 36 months (0)
- license_type: the subscription basis chosen by the customer. Either end-users or operators. This relates to the license bracket: if the license type = end-users, then the license_bracket indicates how many end-users are allowed to use the TOPdesk software of the customer. If the license_type = operators, then the license_bracket indicates how many operators are allowed to use the TOPdesk software.
- Churn: indicates whether the customer still has an active contract (Churn = 0) or had an active contract in the past, but does not have an active contract anymore (Churn = 1).

Appendix B: Literature overview methods

The table below shows an overview of the most important findings per method, where each row represents a past research paper.

Table 10: Overview of previous findings on churn prediction models

	Logistic Regression	Decision Tree	Random Forests	Naive Bayes	Support Vector Machines	Neural Networks
Breiman (2001)	NA	NA	More robust than DT, less sensitive to noise	NA	NA	NA
Coussement & Van den Poel (2008)	Ease of use, interpretable, competitive performance with data prep techniques	Interpretable for decision making in business context	Less influence of outliers, only 2 hyperparameters to set, best performing method for churn prediction	NA	Good out of sample performance	NA
Coussement et al. (2010)	Linear relationship assumed	NA	NA	NA	NA	NA
Celik & Osmanoglu (2019)	Suitable for low dimensional data	Easy to integrate into databases	NA	Easy to use	NA	Imitates the functioning of the human brain
Daskalaki et al. (2006)	NA	NA	NA	NA	Bad performance when undersampling is used for data prep	NA
Dingli (2017)	NA	NA	NA	NA	NA	NN is capable of representing any distribution, good performance in other contexts
Huang et al. (2012)	NA	NA	NA	NA	NA	Computationally expensive, not suitable for large datasets
Keramati et al. (2014)	NA	Interpretable, inexpensive, flexible	NA	NA	Kernel function is most suitable for churn prediction modelling	Good at detecting patterns, high predictive performance, limited explanation capability
Kirui et al. (2013)	NA	NA	NA	Outperforms DT	NA	NA
Larivière & Van den Poel (2005)	NA	Not robust	Reasonable computation time, ease of use, outperform logistic regression models	NA	NA	NA
Neslin et al. (2006)	Positively associated with predictive performance	NA	NA	NA	NA	NA
Ngai et al. (2009)	NA	Bad out of sample performance, interpretability is a great advantage	NA	NA	NA	Suitable in a wide range of CRM applications
Shaaban et al. (2012)	NA	Interpretable	NA	NA	SVM outperforms Decision Trees and Neural Networks	Provides prediction with its likelihood, outperforms logistic regression and Decision Trees
Vafeiadis et al. (2015)	Competitive performance, easily interpretable	Bad performance	NA	NB is outperformed by DT and SVM	SVM outperforms Naive Bayes, Decision Trees and sometimes Neural Networks	Outperforms logistic regression and Decision Trees

Appendix C: Results

Appendix C.1 Tuning the models

Each model is tuned using 10-fold cross validation to find the optimal hyperparameters for that model. This Appendix shows the results of tuning the hyperparameters in Table 11, where the optimal hyperparameters are shown for each model and each input dataset. Regarding the hyperparameter `m`, that is tuned in the Random Forest model, it should be noted that the default `m` differs for the Random Forest based on the ADASYN input data, because this dataset contains dummy variables for each category of the categorical variables. The default `m` for the RF based on ADASYN input data is 6.

Table 11: Results of hyperparameter tuning

	DT - cp	RF - m	SVM - Cost
Default	0.01	4	1.0
Imbalanced	0.01	3	0.5
ROS	0.01	4	0.1
SMOTE	0.01	3	2.0
ADASYN	0.01	13	2.0

In addition to the optimal hyperparameters, Table 12 shows an overview of the performance of both the default models and the tuned models. These results show little difference between the performance of the tuned models and the performance of the default models.

Table 12: Overview results including default models

	Accuracy	Recall	MCC	AUC	Geometric Mean
DT - imb	0.917	0.943	0.748	0.938	0.19
DT - imb (tuned)	0.91	0.934	0.723	0.933	0.192
DT - ROS	0.783	0.96	0.537	0.824	0.315
DT - ROS (tuned)	0.783	0.96	0.537	0.824	0.315
DT - SMOTE	0.876	0.947	0.659	0.89	0.257
DT - SMOTE (tuned)	0.876	0.947	0.659	0.89	0.257
DT - ADASYN	0.863	0.944	0.631	0.89	0.18
DT - ADASYN (tuned)	0.863	0.944	0.631	0.89	0.18
RF - imb	0.94	0.976	0.813	0.969	0
RF - imb (tuned)	0.94	0.976	0.813	0.969	0
RF - ROS	0.785	0.751	0.553	0.909	0
RF - ROS (tuned)	0.785	0.751	0.553	0.906	0
RF - SMOTE	0.903	0.912	0.732	0.964	0
RF - SMOTE (tuned)	0.901	0.912	0.727	0.965	0
RF - ADASYN	0.919	0.951	0.754	0.962	0
RF - ADASYN (tuned)	0.919	0.951	0.754	0.967	0
SVM - imb	0.854	0.95	0.517	0.912	0
SVM - imb (tuned)	0.855	0.951	0.522	0.912	0
SVM - ROS	0.814	0.807	0.561	0.903	0
SVM - ROS (tuned)	0.811	0.802	0.56	0.905	0
SVM - SMOTE	0.861	0.893	0.606	0.913	0
SVM - SMOTE (tuned)	0.864	0.897	0.612	0.913	0
SVM - ADASYN	0.82	0.813	0.574	0.9	0
SVM - ADASYN (tuned)	0.82	0.813	0.574	0.9	0

Appendix C.2 Decision Tree output

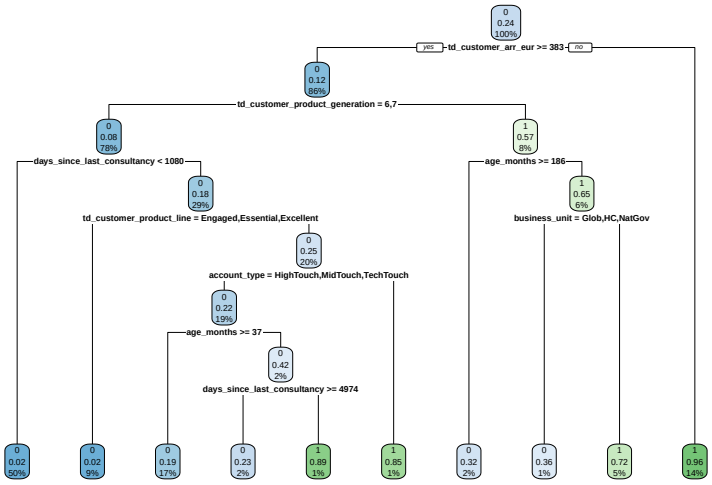

Figure 9: Default DT - Imbalanced Data

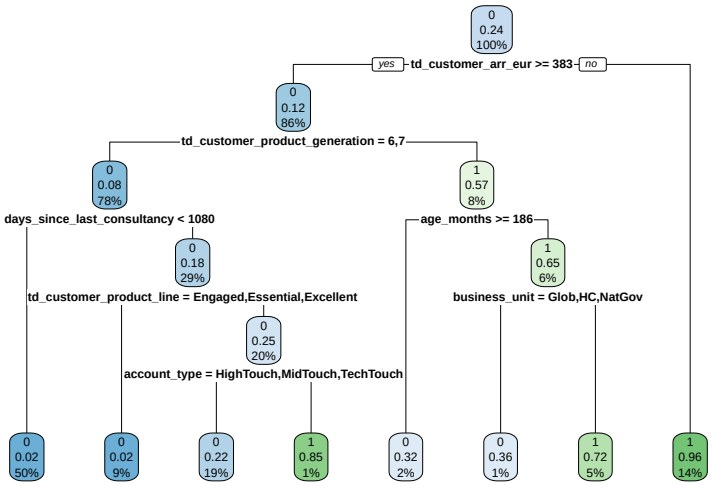

Figure 10: Pruned DT - Imbalanced Data

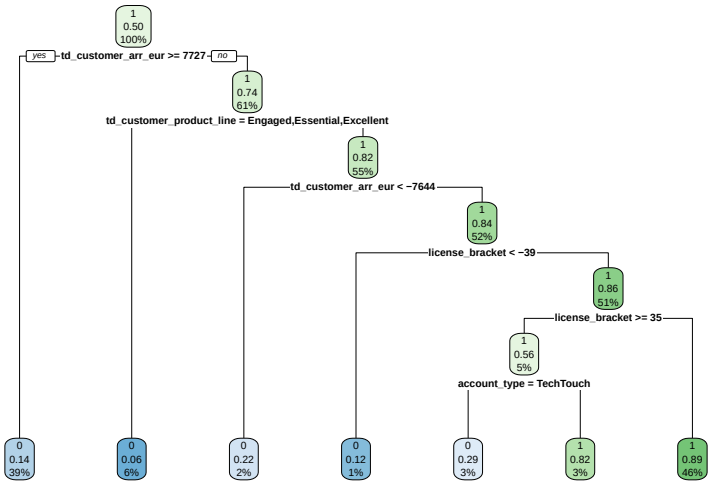

Figure 11: Default DT - ROS Data

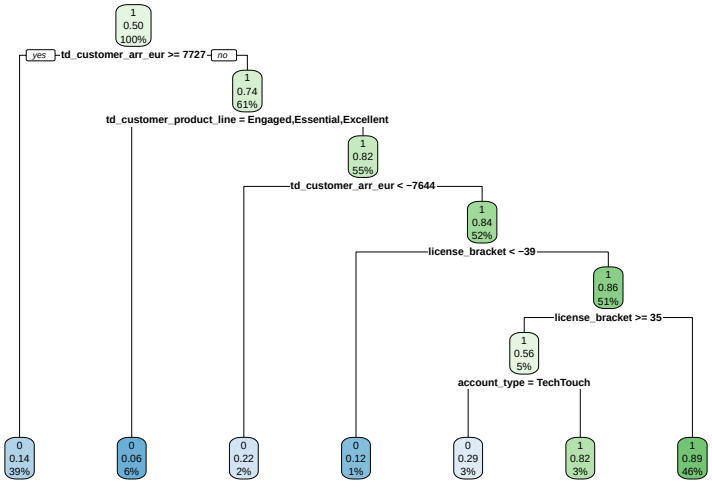

Figure 12: Tuned DT - ROS Data

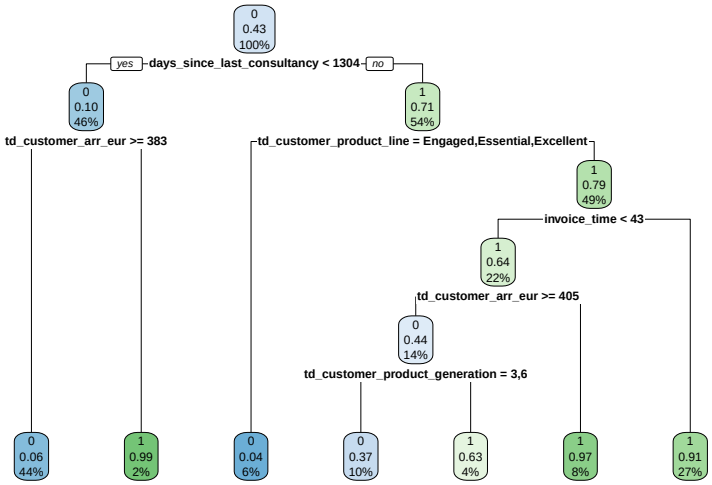

Figure 13: Default DT - SMOTE Data

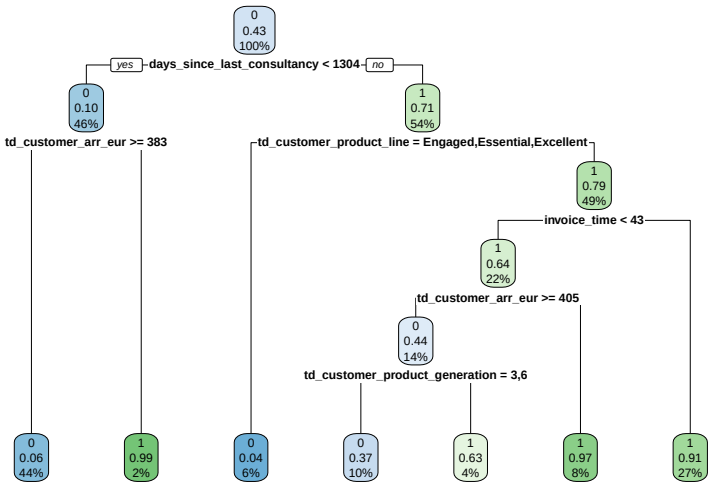

Figure 14: Tuned DT - SMOTE Data

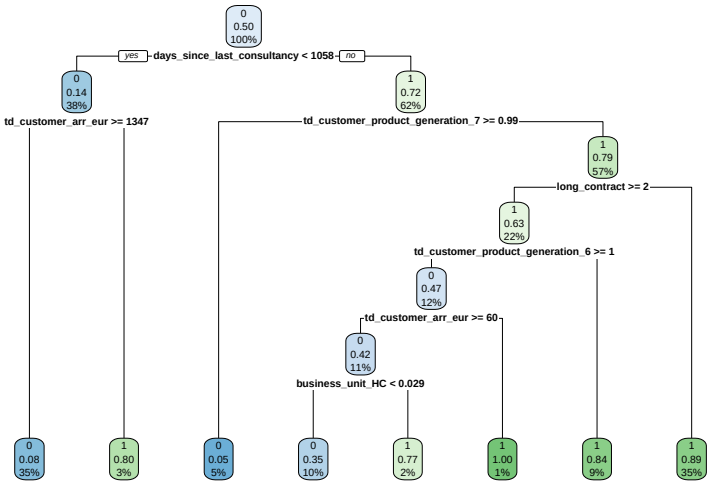

Figure 15: Default DT - ADASYN Data

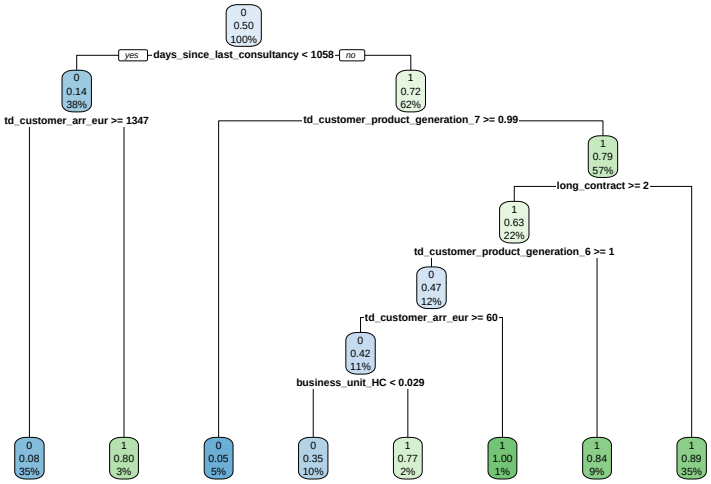

Figure 16: Tuned DT - ADASYN Data