UPPSALA
UNIVERSITET

UPTEC STS 24006
Examensarbete 30 hp
Juni 2024

Predicting non-contractual customer churn in the tourism industry using machine learning

Hannah Liljestam
Emma Lindell

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA
UNIVERSITET

Predicting non-contractual customer churn in the tourism industry using machine learning

Hannah Liljestam
Emma Lindell

Abstract

Customer churn is a term used to describe customers leaving a company by no longer using their services or products. Companies should develop and target retention strategies towards customers at risk of churning, because customer acquisition is more costly than customer retention. At-risk customers can be identified using predictive machine learning. Previously, predictive churn modelling has typically been made for companies offering contractual products, where payments are made on a regular basis following a subscription or other contract. In these cases, the moment a customer churns is intuitively identified. Defining when a customer churns from a company offering non-contractual products, where the purchase occasions are sporadic, is more difficult, as the exact churn moment is both subjective and hard to identify. No studies of non-contractual customer churn have been made in the winter tourism industry, the industry in which non-contractual churn is defined and predicted in this thesis.

The purpose of this thesis is to define and predict non-contractual customer churn in the winter tourism industry. The purpose is fulfilled by creating two different definitions of customer churn; one where the complexity of non-contractual churn is captured through the integration of industry knowledge and the theoretical background, and one that is based solely on the theoretical background. Five frequently used machine learning classifiers are evaluated for the prediction, revealing that our first definition of churn yields the highest AUC performance when predicting customer churn in this case. We conclude that if the definition of churn is sufficiently complex, non-contractual churn in the winter tourism industry can be predicted with a high performance using an XGBoost classifier. When data of previous reservation and purchase patterns is considered, the classifier achieves what is considered to be an excellent AUC performance at nearly 86%.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Sebastian Ekroth Ämnesgranskare: Sebastian Mair

Examinator: Elísabet Andrésdóttir

Sammanfattning

I denna uppsats predikteras icke-prenumerationsbundet kundbortfall med hjälp av maskininlärningsmodeller. Begreppet kundbortfall kan användas för att beskriva det ögonblick en kunds relation med ett företag upphör, typiskt genom att kunden slutar använda företagets produkter eller tjänster. Kundförvärv är mer kostsamt för företag än att bibehålla nuvarande kunder, vilket motiverar utveckling och implementering av strategier som minskar kundbortfall. Dessa strategier bör vara riktade mot kunder som riskerar att gå förlorade; alltså mot *rätt* kunder, snarare än ett företags hela kundbas. Högriskkunder kan identifieras genom prediktiv maskininlärning, med målet att förutse vilka kunder som kommer att gå förlorade efter att ett visst tidsintervall har passerat. Tidigare har prediktiva maskininlärningsmodeller primärt använts för att förutspå kundbortfall från prenumerationsbundna kunder. En kund kan betraktas som prenumerationsbunden om den köper ett företags produkter eller tjänster via en prenumeration, eller annan form av kontrakt, som säkerställer att betalningar görs på en regelbunden basis. En icke-prenumerationsbunden kund är någon vars köptillfällen sker på en oregelbunden basis och inte är baserade på en prenumeration eller ett annat kontrakt. För att kunna prediktera kundbortfall med maskininlärning krävs en definition för när en kund kan betraktas som att den har fallit bort. Denna definition är svårare att formulera för en icke-prenumerationsbunden kund, vars exakta bortfallsögonblick är svårdefinierat och subjektivt, än för en prenumerationsbunden kund där bortfallsögonblicket ofta är intuitivt, exempelvis genom att kunden väljer att sluta prenumerera på en produkt eller tjänst. Hittills har inga studier utförts vad gäller definieringen och prediktionen av icke-prenumerationsbundet kundbortfall inom vinterturismindustrin, vilket studeras i denna uppsats.

Syftet med uppsatsen är att definiera och prediktera bortfall av icke-prenumerationsbundna kunder inom vinterturism. Syftet uppfylls genom en fallstudie som utförs på företaget SkiStar AB, Skandinaviens ledande semesterarrangör. SkiStar AB driver fjällanläggningar i kombination med hotellverksamhet och fastighetsutveckling. Företaget har fjällanläggningar i Sälen, Åre, Vemdalen och Hammarbybacken i Sverige, samt Trysil och Hemsedal i Norge.

Projektet inleds med en litteraturstudie av tidigare forskning inom kundbortfall. Resultatet av litteraturstudien används för att skapa två olika definitioner av icke-prenumerationsbundet kundbortfall. Den första definitionen kombinerar resultatet av litteraturstudien med kunskap från industrin, inhämtad från samtal med företaget. Definitionen fångar komplexiteten i icke-prenumerationsbundet kundbortfall, genom att ta i beaktning att inte alla kunder faller bort på samma villkor. Så kallade *besökscykler* skapas för att fånga mönster i en kunds tidigare köp- och reservationsbeteende, och bortfallsvillkoret varierar för kunden beroende på vilken besökscykel den har.

Den andra definitionen bygger helt på resultatet från litteraturstudien, där det framkommer att den mest förekommande definitionen för ett icke-prenumerationsbundet kundbortfall är via en inaktivitetsperiod, då kunden inte köper något från eller interagerar med företaget. På så vis är den andra definitionen mindre komplex än den första.

Fem maskininlärningsalgoritmer används för att genomföra prediktionen. Genom att jämföra hur väl modellerna predikterar kundbortfall enligt de olika definitionerna framkommer det att den första definitionen är att föredra över den andra. Modellen som presterar bäst är en så kallad XGBoost-modell, med ett *AUC* värde på 85,56%. AUC står för *Area Under the Curve* (sv. arean under grafen) och är ett populärt prestandamått inom maskininlärning. En modell kan få ett AUC värde mellan 0% och 100%, där ett AUC på 100% uppnås av en modell som predikterar rätt klass i alla lägen och 0% av en modell som predikterar fel klass i alla lägen. Det framkom även att de viktigaste attributen att inkludera i den data man använder för att konstruera modellen är relaterade till en kunds köp- och reservationsmönster.

I studien framkom det att en definition för icke-prenumerationsbundna kundbortfall bör ta hänsyn till att kunder kan falla bort på olika villkor. Industrispecifik kunskap bör tas i beaktning när definitionen formuleras, dels på grund av komplexiteten som nämnts ovan, men även för att en definition av icke-prenumerationsbundet kundbortfall måste skräddarsys för respektive verksamhet man beaktar.

Acknowledgements

This Master’s thesis marks the end of five years at Uppsala university and our graduation from the Master’s Programme in Sociotechnical Systems Engineering. The five years have been rewarding, both in regard to education and personal growth. We are both excited to see what the future holds for us!

We would like to extend many thanks for the support we have received from Uppsala university, particularly from our subject reviewer Sebastian Mair. Thank you for your valuable feedback, ideas, and for your suggestions on how to best complete the project. We would also like to extend our gratitude to our examiner and programme director Elísabet Andrésdóttir, who has been a source of constant inspiration throughout the five years we have attended the programme that she co-founded almost 25 years ago.

This thesis has been written in collaboration with the company SkiStar AB. We would like to thank our supervisor Sebastian Ekroth, for his invaluable advice, support, and great enthusiasm. We would also like to thank Nina Hallström and Karin Hartelius for sharing their insights with us.

Lastly, we would like to thank one another for a successful collaboration on this project, making this last semester of our five year programme a fun and memorable experience.

Hannah Liljestam & Emma Lindell
Uppsala, June 2024

Table of contents

1. Introduction

Customer *churn* describes the moment in which a customer leaves a company by no longer using their services or products. Churn is, in other words, synonymous with the loss of a customer. The opposite of customer churn would be customer *retention*, or keeping a customer's business (Gold, 2020, Chapter 1). Companies began battling churn with intention and allocated resources after the publication of the paper *Zero Defections: Quality Comes to Services* by Reichheld and Sasser (1990) (Kumar & Petersen, 2012). What Reichheld and Sasser (1990) referred to as customer *defection* is known in most papers today as customer churn. Reducing churn by only 5% can increase a company's profitability by 25% to 85%, depending on the industry. What is more, the longer a customer stays with a company, the more profit they may generate. Churn can affect a company's profits more than factors that are normally linked to a competitive advantage, such as market share, scale, and unit costs (Reichheld & Sasser 1990). García et al. (2017) state that both anticipating churn and launching strategies for customer retention may lead to a competitive advantage. A company's main asset is its customers and thus the company "must prolong the life expectancy of their customer portfolio as much as possible", according to García et al. (2017).

Churn is most interesting to study for companies with repeat customers, according to Gold (2020, Chapter 1). *Contractual* products and services are subscription-based, meaning purchases are performed periodically based on a contract and continue until that contract is terminated or not renewed (Ahn et al., 2020). Contractual churn is thus intuitively defined as the moment a subscription is terminated (Gold, 2020, Chapter 1). Companies who offer *non-contractual* products or services deal with customers that are not bound to them by a subscription or contract, making non-contractual customer churn much harder to detect than contractual churn (Buckinx & Van den Poel, 2005). Defining churn for a company may be difficult, since there is no clear indicator of when a customer transitions into being a churned one (Reichheld & Sasser, 1990). Buckinx and Van den Poel (2005) state that defining churn is particularly difficult for non-contractual customer churn.

Companies within the tourism industry offer mainly non-contractual products and services. The tourism industry is highly competitive, as the demands and expectations of the customers are constantly increasing and companies have to compete for market shares on both national and international scale. Dursun-Cengizci and Caber (2024) explain that the customers are only willing to establish long-term relationships with the company whose services offer unique experiences of high quality. Customer relationship management is thus crucial for companies in the tourism industry to maintain long-term customer relationships and increase profitability through customer retention (Dursun-Cengizci & Caber, 2024).

Machine learning models are frequently used to combat customer churn through predictive classification (Gold, 2020, Chapter 1; Huang et al, 2012). Being able to identify which specific customers are at risk of churning allows the company to take targeted action to retain that specific customer, thus making the retention process more cost efficient than if actions were put in place equally for all customers (Berry & Linoff, 2004, p.6). However, Gold (2020, Chapter 1) stresses that churn prediction should not be viewed as an easy fix for a company to reduce their churn rate. Rather, churn prediction is a tool that can be used to identify which customers are at risk of churning, but then the real churn reduction is made by taking action based on that identification. Such actions may include product improvement, campaigns, and rightsizing prices (Gold, 2020, Chapter 1).

The goal of this thesis is to predict future customer churn for a winter tourism company offering non-contractual products, using machine learning algorithms to design, train, test, validate, and evaluate predictive models that achieve acceptable AUC performances. The goal is achieved by developing predictive machine learning models for the company SkiStar AB. As no universal definition for non-contractual customer churn exists, achieving the goal cannot be done without first defining non-contractual customer churn for this research case. Hence, defining customer churn will be an essential part of this research project.

1.1 Previous studies on predicting customer churn

Predictive customer churn modelling has previously been studied in several research papers. According to Ahn et al. (2020), the field is somewhat complicated for researchers to approach, since the field is a mixture of an engineering issue and a business administration issue. Liu et al. (2023) strengthen this statement, discussing how customer churn used to be studied from a customer relationship management perspective. In the early 2000's, Berry and Linoff (2004, p.17-18) described data mining as being the key for companies in figuring out which customers have the incentive to churn, which customers will stay regardless of retention efforts, and which customers the company should let go. With the development of artificial intelligence and its many applications, the research on customer churn that is being performed today has evolved into using artificial intelligence (Liu et al., 2023).

Compared to the amount of studies on contractual churn, few cases of non-contractual churn have been academically studied. Examples include Tamaddoni Jahromi et al. (2010) (telecommunication industry), Larivière and Van den Poel (2004) (banking industry), Lee et al. (2019) (gaming industry), Yang et al. (2019) (free online gaming industry) and Buckinx and Van den Poel (2005) (grocery retail industry). A trend within previous work is that the researchers are studying "simple" and straightforward definitions of churn. In their non-contractual study, Larivière and Van den Poel (2004) define churn for customers with savings and investments accounts as occurring when the customer has closed all their accounts. Lee et al. (2019), who studied churn in the gaming industry, states that a customer who plays a particular game has churned after not having played the game for five weeks. Yang et al. (2019) define churn for a free online game as a customer who has not played the game for more than three days, since they found that over 95% of their players did not return after three days of inactivity.

Previous studies have also been performed within the tourism industry, mainly focusing on churn from hotels. Predictions of customer churn have been made for repeat customers at hotels (Durun-Cengizci & Caber, 2024), hotel reservation websites (Han, 2018), and three to five star hotels (Taherkhani et al., 2024). The studies that were found used either prolonged inactivity to define non-contractual churn or applied clustering algorithms to identify churners (Durun-Cengizci & Caber, 2024). As the variation of non-contractual churn is limited, we find there is room to explore other types of definitions of non-contractual customer churn. Durun-Cengizci and Caber (2024) further explain that factors that affect customer retention are, among others, customer satisfaction, quality of service, prices, loyalty programme, customer engagement, recency, accommodation type, monetary and socio-demographic factors.

As far as we can conclude, there have been no studies of churn for companies on the global alpine skiing market, such as the company in this research case.

1.2 Purpose and research questions

With the goal of this project and the above previous work in mind, the purpose of this thesis is to define and predict non-contractual customer churn in the winter tourism industry.

The following two research questions are investigated to fulfil the purpose of this thesis:

1) How can non-contractual customer churn be defined to enable prediction of future churn for a company in the winter tourism industry?
2) How can supervised machine learning be used to predict non-contractual customer churn in the winter tourism industry?

1.3 SkiStar AB

This thesis follows a research case performed at the company SkiStar AB (from here on out referred to as SkiStar), a Swedish holiday organiser operating the biggest alpine mountain resorts in Scandinavia. Their mountain resorts are located in Sälen, Åre, Vemdalen and Hammarbybacken in Sweden, and Trysil and Hemsedal in Norway. The company offers alpine skiing at their resorts in the winter season and active holidays for the summer season.[1]

[1] SkiStar Corporate, *About SkiStar*, SkiStar AB, https://www.skistar.com/en/corporate/about-skistar/ [Retrieved: 2024-02-12]

The company was founded in 1975, then named Sälenstjärnan AB. The company started at the destination Sälen, which is where their expansion began and over the years the company has acquired the biggest skiing resorts in Sweden and Norway. In 2001, the company acquired the name SkiStar AB. In 2021, the company's summer concept was launched.[2]

SkiStar operates within the tourism industry, more specifically in the global alpine skiing market. The global tourism industry is one of the largest industries in the world and since 1995 it has grown approximately 166%. In Sweden, 2.5% of the GDP was accounted for by tourism in 2019. The alpine skiing market is a big market, with an estimated 135 million skiers worldwide and an estimated 400 million skier days per year. In the 2018/2019 winter season, 19 million skier days were accounted for by Sweden, Norway and Finland.[3]

SkiStar offers their customers several different kinds of products. They offer lodging opportunities (different kinds of accommodations and hotels), skipasses, skiing lessons, retail products and sports goods, equipment rental, travel arrangements, insurance and more. These products are mainly mediated through the company's website, are non-contractual in nature, and can be described as infrequently purchased goods that are dependent on seasonality.[4]

1.4 Delimitations

As with any project with a set deadline, time restrictions forced us to make limitations to the scope, for example the number of machine learning models that would be explored. The number of definitions of customer churn that were created for this thesis were also limited by time-constraints coupled with the lack of preexisting studies and the complexity of the research case. Simply understanding the company's business and data well enough to be able to concretize feasible and realistic definitions was a time-consuming process, making it infeasible to explore more than two definitions of customer churn.

Another delimitation was the inability to test how well the definitions of customer churn formulated in this project represent reality. Ahn et al. (2020) highlights the importance of including an observation period after churn that is used to determine the accuracy of the definition, but such a window could not fit within the allocated time for the project. As a result, we cannot conclude if the definitions that are produced during the project actually reflect reality through practical testing. Despite this, we are confident in our results, since we have found support in academic literature, analysed and detected

[2] SkiStar Corporate, *Our history*, SkiStar AB, https://www.skistar.com/en/corporate/about-skistar/our-history/ [Retrieved: 2024-02-12]

[3] SkiStar Corporate, *Our industry*, SkiStar AB, https://www.skistar.com/en/corporate/about-skistar/our-industry/ [Retrieved: 2024-02-12]

[4] SkiStar Corporate, *Business concept*, SkiStar AB, https://www.skistar.com/en/corporate/about-skistar/business-concept/ [Retrieved: 2024-02-19]

patterns in customer behaviour in historical data, and consulted the knowledge of experienced people working within the industry and who are familiar with their customers' behaviours.

The value of predicting customer churn lies in being able to increase customer retention by launching retention strategies based on the results (García et al., 2017). The time constraint of the project makes it impossible to follow up on any potential retention strategies launched by the company, which will not allow us to determine the monetary value that this project generated for the company.

1.5 Project overview

Researching customer churn requires an understanding of both the engineering aspect (mainly data science and machine learning) and the business administration aspect (mainly customer relationship management) (Ahn et al., 2020). We approach previous work conducted within this field by initially separating the engineering aspect of the field from the business administration aspect. Within the business administration aspect, we describe definitions of customer churn in its various forms and try to gain an understanding of what churn entails for different kinds of businesses and products. Within the engineering aspect, we study the different data scientific approaches researchers have previously taken when modelling customer churn and which specific machine learning models have previously been applied.

We perform our research through initially conducting a literature review, where we investigate previous work on customer churn as a research field, machine learning, and how machine learning has been used to predict customer churn. This literature review will give us the necessary knowledge needed to have productive discussions with the company about how customer churn is best described for them. It also allows us to decide which machine learning models to use in our work, based on which models have been proven useful in previous work. We then decide on a definition of customer churn for the company through meetings and brainstorming sessions where we analyse, criticise, and redefine the definition iteratively. After having defined customer churn, we analyse and process the data provided by the company before we begin developing our machine learning models using the processed data. We train, validate, and test our developed models before evaluating them. We then analyse and discuss our results to gain interesting insights about our case. Finally, we choose the most accurate model and hand over our work and our findings to the company. A more detailed description of the project execution can be found in the third chapter of this thesis, *3 Method.* A description of differences in what our main responsibilities within the project are can be found in Appendix.

2. Theoretical background

In this chapter, we provide an overview of the existing knowledge and the previous work done in the field of customer churn when predicted using machine learning, as well as the background knowledge that we have determined necessary for the reader to possess to understand the specific research field. We limit ourselves to describing the kinds of machine learning algorithms that are used in this project, which centres around a classification problem.

2.1 Customer churn

Customers leaving a company is counteractive to the company's growth and can result in the growth stagnating, or even becoming negative (Gold, 2020, Chapter 1). In his book *Fighting Churn with Data*, Gold (2020, Chapter 1) defines the loss of customers, or *churn*, as "when a customer quits using a service or cancels their subscription." In the Cambridge Dictionary, *churn* – when concerning customers – is defined as "the number of customers who decide to stop using a service offered by one company and to use another company, usually because it offers a better service or price".[5]

As mentioned in the introduction to this thesis, fighting churn was popularised by a paper published by Reichheld and Sasser (1990), where they described how reducing churn by only a few percent may raise a company's profitability a significant amount (Kumar & Petersen, 2012). Churn also affects a company's profits more than other factors influencing a company's competitive advantage, and the longer the customer stays, the more profit is generated (Reichheld & Sasser, 1990).

Customer acquisition used to be less costly, but the price of acquisition has been rising rapidly because of globalisation and international competition (Ahn et al., 2020). This is confirmed by Gold (2020, Chapter 1), who states that many products and services today – particularly subscription-based – are offered by several competitors competing for the same customers. Gold (2020, Chapter 1) also points out that there are usually several ways today to achieve the same ends when it comes to these products or services, compared to the twentieth century, e.g. by a company developing its own systems rather than subscribing to a service offering a similar system. Additionally, many of these services and products are things we can simply do without. Because of today's landscape, Gold (2020, Chapter 1) stresses that companies today cannot only focus on acquiring new customers in order to grow and be successful, but also have to make an effort to reduce churn in order to fully realise the company's potential. Analysing churn and making attempts to reduce it is thus one way for companies to improve their business outcomes (Ahn et al., 2020).

[5] Cambridge Dictionary, *Churn*, https://dictionary.cambridge.org/dictionary/english/churn. [Retrieved: 2024-02-02]

To fight churn, one needs a definition for when a customer can be considered to have churned. In some cases, designating a definition of churn for a company may be difficult, since there is no clear indicator of when a customer churns. Reichheld and Sasser (1990) illustrate this through the example of a railroad business. A customer who used to ship 100% of their shipments via the railroad business and now only ships 20% of their shipments via the business is not per se a churned customer, but at the same time they cannot be seen as a retained customer either (Reichheld & Sasser, 1990).

2.1.1 Fighting churn through prediction

Fighting churn is, at its core, a customer relationship management issue. However, the research field has evolved into including data mining and, in recent years, artificial intelligence. Modelling churn today can be done using two basic approaches, according to Berry and Linoff (2004, p.119-120); *predicting a customer's estimated remaining lifetime* through so-called survival analysis or *predicting which customers will leave*, which produces a binary-outcome prediction problem of who will leave and who will stay. To model the latter kind of prediction, a suitable time horizon or time window is necessary to define. Usually the time horizon is fairly short; somewhere between 60 and 90 days, but Berry and Linoff (2004, p.10, 119) also exemplify this using a time horizon of six months. However, the time horizon may differ; Tamaddoni Jahromi et al. (2010) study churn over a three month-period in non-contractual telecommunication, while Larivière and Van den Poel (2004) study churn through survival analysis over a period of eleven years.

To fight churn, one also needs a way to measure it. *Churn rate* refers to "the proportion of customers departing in a given period", which is where the term churn originates from (Gold, 2020, Chapter 1). The term is a commonly used indicator to determine customer churn (Ahn et al., 2020), and Reichheld and Sasser (1990) described defection (churn) rate as "an accurate leading indicator of profit swings". Churn rate can be shown through the following equation (Gold, 2020, Chapter 2):

$$\text{Churn rate } = \frac{\text{Number of churned customers}}{\text{Number of customers at the start of the period}} \tag{Eq. 1}$$

Customer *retention* is the opposite of customer churn (Gold, 2020, Chapter 1). Retaining a customer is significantly less costly for a company than it is for them to acquire a new customer, according to Kumar and Petersen (2012). The goal of managing customer retention is not to achieve zero cases of customer churn, but rather to optimise customer equity, according to Blattberg et al. (2002). Meaning, it is not worth it for a company to target all potential churning customers with retention strategies, but rather the company should focus on targeting the *right* customers with their retention strategies. The relationship between churn rate and retention rate is that they, together, should equal 100% of the customers the company had at the start of an observation (Gold, 2020, Chapter 2).

2.1.2 Contractual and non-contractual customer churn

Customer churn is most easily studied in companies with subscription or recurring payment business models (Gold, 2020, Chapter 1). These companies sell products or services which are contractually based, meaning the customer has some kind of continuous relationship established through a contract with the company that they can either renew or cancel when the contract comes to an end (Ahn et al., 2020; Gold, 2020, Chapter 1). Many modern subscription-based products or services are characterised by being offered online, by there being similar, alternative services for a customer to choose from, and that they are often services people can do without (Gold, 2020, Chapter 1).

For non-subscription, non-contractual products, churn may be defined through user inactivity (Gold, 2020, Chapter 1). Ahn et al. (2020) also state that churn may be described as an extended period of inactivity. For a definition of non-contractual churn, there are no time constraints on when the customer may leave the service, since there is no contract binding the customer to the service for a given amount of time (Ahn et al., 2020). For non-contractual products, one must define "a time window in which a user must engage with the service", and then if the customer is inactive for that specific time window, they will be defined as having churned. The time window should consist of a period where inactivity is noted, but the customer is not yet considered to be at risk of churning (Gold, 2020, Chapter 1).

After this, the time window contains a period where the customer is considered to be at risk of becoming a churned customer, a so-called churn determination window. Then, after this period is over, the customer will be considered to have churned. One may also include a period after the churn determination window, where one can observe whether the customer returned or not, which may provide valuable information about the accuracy of the definition and analysis in the long run (Ahn et al., 2020).

2.2 Predictive machine learning

Machine learning models are frequently used to combat customer churn (Huang et al, 2012), but the research field is currently being explored within several industries and is far from new. In an article titled *Computing Machinery and Intelligence* published in 1950, Alan Turing (1950) posed the question "Can machines think?". He proposed a test, named *The Imitation Game*, in which one person would communicate with two entities; a human and a machine. If the person could not determine which of them was the machine, the machine had won the game and should be regarded as intelligent (Turing, 1950). Nine years later, machine learning was defined by Arthur Samuel as "a field of study that gives computers the ability to learn without being explicitly programmed" (Bell, 2015, Chapter 1).

Machine learning is a branch of artificial intelligence in which algorithms that help machines learn over time are designed and developed. The models exhibit some human-like intelligent behaviour by being able to automatically improve through experience (Mellouk, 2010, p.IX) and can predict outcomes based on learning during its lifetime (Bell, 2015, Chapter 1).

By identifying behavioural patterns in customer data, a model can be trained to predict whether or not a customer is about to churn, allowing the company to take action to retain that customer. Preventive actions can be targeted specifically at the at-risk customers identified in the prediction, making the retention process more cost efficient than if actions were put in place for all potential churners (Berry & Linoff, 2004, p 6).

2.2.1 Supervised learning

A machine learning model is constructed in different steps. The underlying data is split into subsets, one used for training the model and one used for testing it (James et al., 2013, p.166). During training, the model is fitted to the training data and validation is then performed using subsections of the training data. Validation is thus a way of estimating the test error using the training error (James et al., p.24-36), allowing for optimisation of the model (Murphy, 2022, p.125). Further explanation of the validation process is provided in Chapter *2.4.1 K-fold cross-validation.* After a model has been selected based on its performance during validation (Murphy, 2022, p.13), the model is tested on the unseen test data set to determine how well the model generalises to new unseen data (James et al., 2013, p.30).

Machine learning algorithms can be trained by using labelled or unlabelled data (James et al., 2013, p.26). The labels are the outputs that the model should be able to predict, also called the target or the dependent variable (Deisenroth et al., 2020, p.253). In supervised learning, the labels inform us which is the correct outcome and the performance of the model is measured by how frequently the outcome it produces is the same as the label (Bell, 2015, Chapter 1). After training, the algorithm should be able to generalise its findings onto new, previously unseen data and make accurate predictions of the outcome without having access to a label (Marsland, 2014, Chapter 1).

A machine learning model is based on a predictive function *f*, also called a *predictor*. In supervised learning, the predictor is given a data set of *independent variables*, also called *features*, and tasked to predict the value of each corresponding dependent variable, also called label. The data set is provided in the form of a table, where each column is a feature *d*, and every row is an *instance x*, also called a *data point*. The features describe the instances, which are later used by the predictor to predict the value of the associated label *y*. For a table of data with *M* number of columns (features) and *N* number of rows (instances), every row creates a vector $\boldsymbol{x_n} \in \mathbb{R}^M$ of inputs, for $n = 1, 2, \dots, N$. The associated label is a scalar $y_n \in \mathbb{R}$. Aggregating all row vectors $\boldsymbol{x_n}$ creates a matrix of all instances called $\boldsymbol{X} \in \mathbb{R}^{NxM}$ and aggregating all outputs create a corresponding label vector $\boldsymbol{Y} \in \mathbb{R}^N$ (Deisenroth et al., 2020, p.253-260).

The predictor *f* is parameterised by a vector of parameters $\boldsymbol{\theta}$. The main task of model design is to estimate a vector of parameters $\boldsymbol{\theta*}$ that are tuned to optimise the performance of the predictor and decrease the difference between the predicted and actual label. The predictor is thus described as

$$\mathrm{f}(\mathbf{x_n}, \boldsymbol{\theta}^*) \approx \mathrm{y_n} \qquad \text{(Eq. 2)}$$

and the predicted label is given by

$$\hat{\mathrm{y}}_\mathrm{n} = \mathrm{f}(\mathbf{x_n}, \ \boldsymbol{\theta}^*) \qquad \text{(Eq. 3)}$$

for all instances *n = 1, 2, …, N* (Deisenroth et al., 2020, p.259).

Machine learning models can be both *parametric* and *non-parametric*. In a parametric model, the number of parameters are fixed and the set of training data is used for learning the parameters in a process where the model is fitted to the data and then discarded. In non-parametric models, the number of parameters vary and the training data is not discarded after fitting, which enables a non-parametric model to learn as many parameters as there are data points in the data set (Murphy, 2022, p.545).

2.2.2 Classification

Machine learning algorithms can be applied to solve either *regression* or *classification* problems. In regression, the outcome is continuous and quantitative. The outcomes can be ordered in relation to each other, where some outcomes are closer together while others are further apart. In classification, the outcome is discrete and qualitative, meaning that although the outcomes are different, the distance between them cannot be measured. In a classification problem the outcomes are called *classes* and each instance is assigned to a class. Collectively, the classes span the whole outcome space (Hastie et al., 2009, p30). A part of the classification problem is to find the *decision boundaries*, which mark where one class ends and another begins (Marsland, 2014, Chapter 1). The outputs of a classification problem are typically represented numerically for ease of use. The easiest classification case is a case with two classes, called *binary classification*. Examples of binary outcomes are “survived” and “died”, or “retained” and “churned”. The numerical representations are usually 1 and 0. If there are more than two outcomes, any numerical value can be used to represent the different classes (Hastie et al., 2009, p.31).

2.2.3 Data treatment

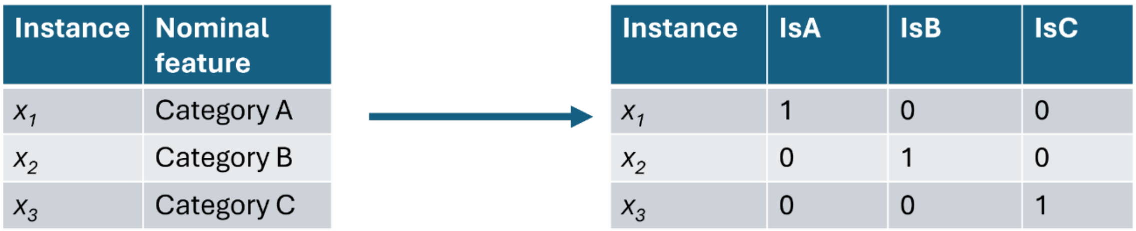
To increase the quality of the predictions made by a classifier, the data used in its construction often has to be pre-processed. Data in its original form is often inconsistent, noisy, and incomplete, issues that can be remedied through pre-processing. Pre-processing can be performed in different ways, some steps include *data cleaning*, which removes inconsistent data points and noise, *data transformation*, which allows for more accurate and efficient algorithms, and *data reduction*, which removes redundant and irrelevant data (Han & Kamber, 2006, p.47-48).

Several actions are taken when data cleaning is performed. Missing values can be handled using *listwise deletion* in a process where instances with missing values are directly removed from the data set without consideration for which features the value is missing from or the features' importance for the final prediction. Listwise deletion is suitable for pre-processing when the amount of data is large enough to not lose important patterns with the removal of instances (Cheng et al., 2021).

Noise is a random error in data which makes the value of a measured data point incorrect. It is smoothened during data cleaning to ensure that it does not skew the results of any data mining algorithm applied to the data, thus making the output of the algorithm unreliable (Han & Kamber, 2006, p 62). Similarly, *outliers* are identified and removed. Outliers are data points that significantly differ from the other data points, do not behave as expected, and are rare in comparison to other "normal" data points (Boukerche et al., 2020). Outliers can be identified using *z-score*. Z-score is a parametric outlier detection which normalises all data points (Anagnostou, 2021). E.g. a z-score of 1.25 informs that the data point lies 1.25 standard deviations away from the mean (Berry & Linoff, 2004, p.162). By setting a threshold of the maximum z-score that a data point can have without being identified as an outlier, outliers can be detected and removed (Anagnostou, 2021).

Data transformation may be necessary if the original form of the data is not appropriate for data mining. *Normalisation* is a transformation that involves rescaling the data to ensure that all values fall within an interval, usually 0-1. Normalisation is useful when working with distance based algorithms, such as K-nearest neighbour. The goal of normalisation is to reduce the risk that values which are naturally higher, for example *cost*, do not outweigh features with naturally lower values, such as *age*. *Min-max normalisation* uses the existing minimum and maximum value of a feature and the desired new minimum and maximum value to perform linear transformations which rescales data to fall within the desired interval (Han and Kamber, 2006, p 70-71).

A feature may need to be split into multiple features to enable the use of certain machine learning models due to its data type. For example, a nominal feature must be transformed into numerical features before being used in a K-nearest neighbour classifier. If the nominal feature contains three categories, each category can be represented with its own numerical feature, resulting in the nominal feature being replaced with three numerical features (Witten & Frank, 2005, Chapter 7). For example, if the nominal feature contained values "A", "B", and "C", it can be transformed into three binary features titled IsA, IsB, and IsC. The feature that corresponds to the value that an instance had in the nominal feature would get the value "1" or "True" and the other features would be "0" or "False". Further explanation is provided in Figure 2.1. The new features are called dummy variables (McKinney, 2012, Chapter 7).



Instance	Nominal feature
x_1	Category A
x_2	Category B
x_3	Category C

Instance	IsA	IsB	IsC
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1

Figure 2.1: Feature transformation using dummy variables (Liljestam & Lindell, 2024)

Data transformation may also be necessary when one of the two classes in the target variable is overrepresented in the dataset, as is common in binary classification. If the imbalance is too great, a classifier could simply assign all instances to the majority class and still obtain a high accuracy. However, always predicting the majority label may be problematic, despite the high accuracy obtained (Witten et al., 2017, Chapter 2). An example of this is during mammography screenings. Not being able to classify instances to the minority class would have significant negative consequences, as that would entail failing to identify cancer cells. One of the most common ways of dealing with class imbalance is to re-sample the original data set, either by *over-sampling* the minority class or *under-sampling* the majority class. Over-sampling means creating new synthetic instances to increase the number of instances included and under-sampling means removing instances to decrease the number of instances included (Chawla et al., 2002).

Synthetic Minority Over-sampling Technique (SMOTE) is an over-sampling technique where synthetic instances are created by taking samples from the minority class and identifying a chosen number of nearest neighbours that also belong to the minority class. New samples are introduced along the line segments which joins the original sample and its neighbours, and then added to the data set (Chawla et al., 2002).

Sometimes, data reduction has to be performed. Performing complex analysis on a large data set is computationally expensive, making the analysis either impractical or almost impossible. Data reduction can lower the computational cost by reducing the data set to exclude irrelevant features, thus enabling the performance of the analysis (Han & Kamber, 2006, p.73). One method to identify which attributes are irrelevant and can be removed is to use *SHapley Additive exPlanations* (SHAP), which is a unified framework for feature reduction. SHAP computes feature importance by combining *shapley sampling values* from game theory with weighted linear regression to determine the impact of each feature on the final prediction. Shapley sampling values approximate how great the impact would be on the model if a feature was removed, by integrating over the data set used for training the model (Lundberg & Lee, 2017).

2.3 Models used for prediction

There are several classification algorithms which can be used to design a predictive customer churn model. Previous studies have shown that the optimal algorithm varies greatly between cases, not only dependent on what industry the company operates within, but also between companies (Ahn et al., 2020). This project explores the classifiers *Decision tree, Random forest, XGBoost, Logistic regression,* and *K-nearest neighbour*.

2.3.1 Decision tree

A Decision tree is a non-parametric model that can be used to solve both regression and classification problems. The algorithm behind a decision tree divides the feature space into subspaces. For classification trees, the subspaces are associated with different classes or values depending on what type of decision tree it is (Rokach & Maimon, 2007, p.8).

Decision trees are frequently used within machine learning due to their conceptual simplicity resulting in high interpretability (Coussement & Van der Poel, 2008). The tree consists of hierarchically ordered *nodes* that are connected through incoming and outgoing *edges*. At every node that has an outgoing edge, a decision is made to determine which path an instance should traverse. The initial decision is made in the *root node.* The root is the only node without an incoming edge, all other nodes have exactly one incoming edge. A node with both an outgoing and incoming edge is called an *internal node* and a node with only an incoming edge is called a *leaf*. A leaf is a node at the bottom of the tree. Each instance is classified by navigating from the root to a leaf, which is associated with a specific class. The number of hierarchical layers in the tree determines its *depth* and the number of nodes determines its *size* (Rokach & Maimon, 2007, p.8-9). For visualisations of a decision tree, see Figure 2.2.

Hastie et al. (2009, p.305) explain the algorithm behind constructing a decision tree, denoted as $f(\boldsymbol{x_n}, \boldsymbol{\theta})$, where f is the predictive function, $\boldsymbol{x_n}$ the vector of instances that are to be classified, and $\boldsymbol{\theta}$ a vector of parameters learned by the model during training. To enable visualisation of the model, the simplest case of recursive binary partitions is used, including only two dimensions. However, decision trees can be used for more complex problems of higher dimensions as well. For example, consider the target variable y being dependent on two features: d_1 and d_2. At the first split, the feature space is divided into two regions, R_1 and R_2, by creating a partition in either d_1 or d_2 at a value s, where the value for the first split is called s_1. The space is partitioned at the variable and value that leads to the best fit of the response variable. The feature space is continuously partitioned into different regions $R_1, R_2, \ldots, R_T$ by creating a split in either d_1 or d_2, where the feature equals threshold value s_{t-1}, resulting in a new region R_t. This continues until the tree has reached a criterion to stop the division, such as a desired depth or size (Hastie et al., 2009, p.305-306).

Two different ways of visualising a decision tree that has been split four times, where *t = 4,* are shown in Figure 2.2.

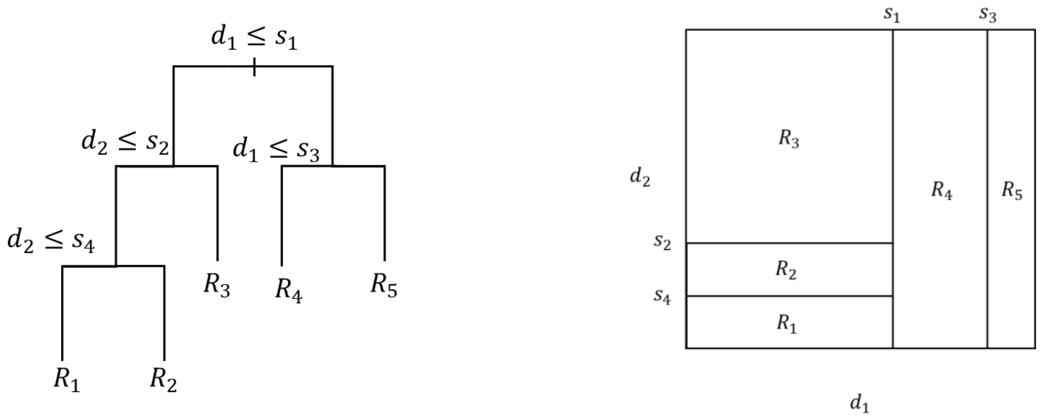


Figure 2.2: Visualisations of a decision tree with four splits (Liljestam & Lindell, 2024).

A decision tree that is used to solve classification problems is called a *classification tree* and its subspaces are associated with different classes. When deciding where to partition the feature space in a classification tree, *node impurity* is considered. The impurity of the leaf node *t* which corresponds to region R_t in tree $f(\boldsymbol{x}_n, \boldsymbol{\theta})$ is described by the variable $Q_t(f)$. Two such impurity measures are *GINI impurity* and *cross-entropy*. GINI impurity is calculated in following manner (Hastie et al., 2009, p309):

$$Q_t(f) = \sum_{i=1}^{I} p_i(1 - p_i) \tag{Eq. 4}$$

and cross-entropy is given by

$$\mathrm{Q_t(f)} = -\sum_{\mathrm{i=1}}^{\mathrm{I}} \mathrm{p_i log p_i} \tag{Eq. 5}$$

where *i = 1, 2, …., I* is the number of classes and p_i is the probability that an instance *x* in node *t* belongs to class *i.* The instances *x* that belong to node *t* are assigned to the majority class of the node. The perk of using GINI impurity or cross-entropy instead of simply using the misclassification error, is that the former prefers and rewards pure nodes by being more sensitive to changes in node probability, which the classification error does not (Hastie et al, 2009, p.309).

Using a single decision tree on high-dimensional data can produce a model that does not generalise well to unseen data, as the tree tends to be overly flexible. One way of decreasing flexibility is to use ensemble methods such as *bagging* or *boosting* to combine multiple trees into one model (Wu et al., 2022). Ensemble methods are combinations of simpler models, in which the strengths of the combined models are used to create a more accurate and reliable model (Hastie et al., 2009, p.607). Bagging is used in the classifier Random forest (Breiman, 2001) and boosting is used in the classifier XGBoost (Chen & Guestrin, 2016), which are explained in the following subchapters.

2.3.2 Random forest

The Random forest algorithm was first introduced by Leo Breiman (2001), as a solution to the high flexibility of decision trees (Wu et al., 2022). It is a non-parametric ensemble method that creates multiple tree-based classifiers to create a classifier.

Each tree in the Random forest model is trained using a unique data set, but the distribution for the trees is the same throughout the entire forest. Following the *Law of Large Numbers*, Breiman (2001) showed that the generalisation error converges as the number of trees increases, resulting in a Random forest classifier not *overfitting* the data as more trees are added. Overfitting is when the model is too flexible and adapts to the training data too well (Murphy et al., 2022, p.13), reducing its ability to generalise to unseen data (Murphy et al., 2022, p.121). A model of high flexibility typically has low *bias* and high *variance*. Bias is the error arising from having to simplify a complex reality using a more simple model and variance explains how much the estimation would differ if another training set was used (James et al., 2013, p.34-35). *The generalisation error* is the difference between the predicted and actual output, also referred to as the prediction error, for an independent test sample where both the inputs and target are randomly selected from their joint distribution (Hastie et al. p.219-220). The generalisation error is dependent on the strength of and the correlation between the individual trees. Breiman (2001) showed that this algorithm performed favourably, in part because the algorithm is relatively resistant to noise and outliers (Breiman, 2001).

The Random forest algorithm uses *bootstrap aggregation*, also known as bagging, to create subsets of data of the same size as the original data set to train its trees, DS_{train}, to increase model variance and solve the inadaptability problem of decision trees (Ho, 1998). The accuracy of a random forest used in binary classification is determined by classifying the *out-of-bag data sets* for each tree, consisting of the data points that were not used for its training. By using bagging, accuracy is increased and internal estimates of strength, correlation, and the generalisation error are given (Breiman, 2001).

A Random forest is a collection of $h = 1, 2, ..., H$ number of tree-based classifiers f, trained on a subset of data generated through bagging, DS_{train_h}. For every unique tree-based classifier f_h, a new data set DS_{train_h} and a random vector $\boldsymbol{\theta}_h$ are generated. Not every feature in the training dataset DS_{train} is considered when splitting the different tree-based classifiers f and the random vector $\boldsymbol{\theta}_h$ is used to select which features are used for splitting in the classifier f_h. The random vectors $\boldsymbol{\theta}$ are generated independently of all previously generated random vectors, meaning that $\boldsymbol{\theta}_h$ is generated independently of the vectors $\boldsymbol{\theta}_1$ through $\boldsymbol{\theta}_{h-1}$, but they all have the same distribution (Breiman, 2001).

To create a random forest, the following algorithm is given (Breiman, 2001):

1) Given a training subset of the original data set, DS_{train_h}, created using bagging, construct tree-based classifiers $f(\boldsymbol{x_n}, \boldsymbol{\Theta}_h)$ on the bootstrapped subsets.
2) Let each classifier $f(\boldsymbol{x_n}, \boldsymbol{\Theta}_h)$ vote on the class of every instance $(\boldsymbol{x_n}, y_n)$ that is included in the training subset, but not in the bootstrapped data set that was used for training the tree DS_{train_h}.
3) Aggregate the votes of all classifiers and assign every instance $(\boldsymbol{x_n}, y_n)$ the majority class of their respective predictions.

In this process, the estimate of the out-of-bag data, meaning the data that was not used in training the classifier, is used for testing the classifier. The approach eliminates the need to set aside a test set, making the estimate as accurate as if using a test set the same size as the training set. Out-of-bag estimates are also unbiased, enabling random forests to limit overfitting without increasing the error that stems from bias, making them a powerful tool for classification (Breiman, 2001). A visualisation of the creation of a random forest with *H* number of trees is given in Figure 2.3.

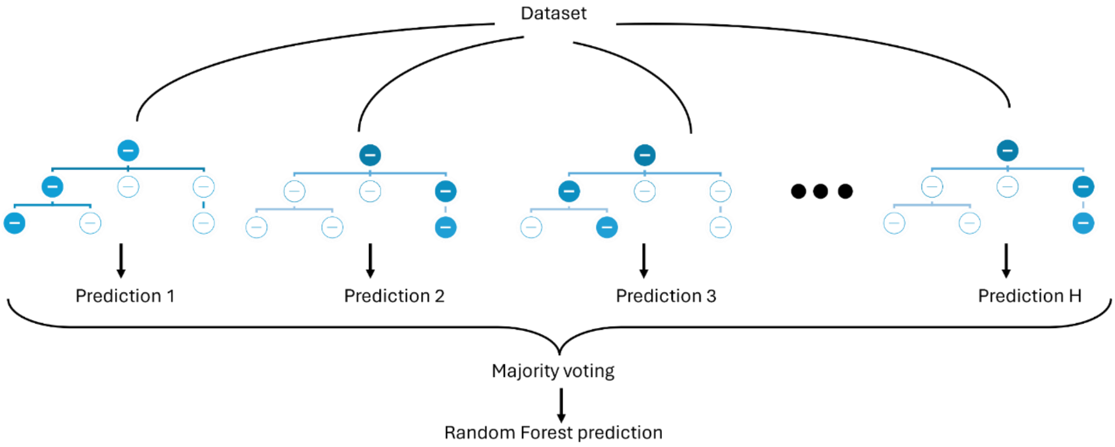


Figure 2.3: Visualisation of a random forest classifier (Liljestam & Lindell, 2024).

2.3.3 XGBoost

A Gradient Boosted Decision Tree (GBDT) is a non-parametric model that utilises boosting to resolve the issue of the high flexibility of single decision trees. Boosting is a way of sequentially constructing classifiers by emphasising the misclassifications of the previous classifier in the training of the following classifiers through the calculation of *residuals* (James et al., 2013, p.321). The residuals are the difference between the actual value and the predicted value (James et al., 2013, p.62). They act as weights for the data points and can thus force the next classifier to prioritise a misclassified data point by increasing its weight. This methodology allows classifiers to learn from the mistakes of its predecessor (Wu et al., 2022) and is realised through the use of a loss function that penalises incorrect predictions (Hastie et al., 2016, p.18).

Extreme Gradient Boosting (XGBoost) was introduced by Chen and Guestrin (2016) as a highly scalable end-to-end system used to implement GBDT:s. As of 2022, it was the superior model of choice in data science competitions (Wu et al., 2022). The objective function, meaning the function we wish to minimise or maximise (Goodfellow et al., 2016, p.95), in XGBoost contains a *regularisation* term to control complexity and reduce the risk of overfitting. Regularisation is a way of expressing a preference for one solution over another, with the goal of reducing the generalisation error (Goodfellow et al., 2016, p.119, 128). The regularisation term is used to prune trees in a process that optimises the trade-off between simplicity, adaptability, and accuracy (Chen & Guestrin, 2016). In XGBoost, tree-based classifiers are created sequentially by first calculating the *gradient* of the loss function of the classifier and fitting the following classifier to the negative gradient (Chen & Guestrin, 2016). The gradient of a function is a vector in the parameter space containing all its partial derivatives, which points in the steepest uphill direction (Hastie et al., 2013, p.12; Goodfellow et al., 2016, p.97).

The optimization function of XGBoost is the sum of convex and differentiable loss functions *l* and a regularisation term **Ω** that penalises the complexity of the model. The loss function is a differentiable convex function, which measures the difference between the correct and assigned class. Letting y_n be the correct class of the *n*th data point, $\widehat{y_n}$ be the model's classification of the *n*th data point, *f* being the classifier that assigned the data point its class, and *H* the number of classifiers included, the regularised objective function for XGBoost is

$$\mathrm{L}(\Theta) \ = \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{l}(\,\hat{\mathrm{y}}_{\mathrm{n}}, \mathrm{y}_{\mathrm{n}}) + \sum_{\mathrm{h}=1}^{\mathrm{H}} \Omega(\mathrm{f}_{\mathrm{h}}), \qquad \text{(Eq. 6)}$$

for *h = 1, 2,….H* and *n = 1, 2…, N.*

Because the objective function consists of other functions, the parameters cannot be optimised using traditional models in the Euclidean space. Instead, the optimization is done additively (Chen & Guestrin, 2016). First candidate splitting points are proposed according to percentiles of the feature distribution. Then the best outcome following the proposed splitting points are identified, based on aggregated statistics of the features affected by the split (Wu et al., 2022).

At every iteration *q*, a classifier f_k is added to the objective function of the XGBoost classifier in a greedy manner, starting from a leaf and adding branches in every iteration. The formula for evaluating split candidates are

$$\mathrm{L}_{\mathrm{split}} \ = \ \frac{1}{2}\left[\frac{\left(\sum_{\mathrm{n}\in\mathrm{N_L}} \mathrm{g_n}\right)^2}{\sum_{\mathrm{n}\in\mathrm{N_L}} \mathrm{n}+\lambda} + \frac{\left(\sum_{\mathrm{n}\in\mathrm{N_R}} \mathrm{g_n}\right)^2}{\sum_{\mathrm{n}\in\mathrm{N_R}} \mathrm{g'}_{\mathrm{n}}+\lambda} - \frac{\left(\sum_{\mathrm{n}\in\mathrm{N}} \mathrm{g_n}\right)^2}{\sum_{\mathrm{n}\in\mathrm{N}} \mathrm{g'}_{\mathrm{n}}+\lambda}\right] - \gamma, \qquad \text{(Eq. 7)}$$

where

$$g_n \ = \partial_{\hat{\mathrm{y}}_{\mathrm{n}}(q-1)} l(y_i, \hat{\mathrm{y}}_{\mathrm{n}}{}^{(q-1)}) \qquad \text{(Eq. 8)}$$

and

$$g'_n = \partial^2_{\hat{\mathrm{y}}_\mathrm{n}(q-1)} \, l(y_i, \hat{\mathrm{y}}_\mathrm{n}{}^{(q-1)}) \quad \text{(Eq. 9)}$$

are the first and second order gradient statistics on the loss function, and λ and γ are parameters used for tuning the model (Chen & Guestrin, 2016).

2.3.4 Logistic regression

Logistic regression is a parametric algorithm used for classification. It does not model the target variable directly, but it rather calculates the probability that an instance belongs to a particular class (James et al., 2013, p.133). Logistic regression was introduced as a way of dealing with binary classification in 1958 by Cox, who argued that the probability can not be given by a linear relation, since the dependent variable is confined to the soft interval (0, 1) (Cox, 1958). Instead, the probability that an instance belongs to a class is calculated using a *logistic* function. Considering the case where the dependent variable is binary, taking the value of either 0 or 1, and the probability that an instance x_n is associated to the label $y_n = 1$ is given by

$$\text{logit}\,\Pr(\mathrm{y_n} = 1 |\, \mathbf{x_n}, \boldsymbol{\Theta}\,) = \log\{\frac{\Pr(\mathrm{y_n}=1)}{1-\Pr(\mathrm{y_n}=1)}\} = \Theta_0 + \Theta_1^\mathrm{T}\,\mathrm{x_n}, \quad \text{(Eq. 10)}$$

where Θ_0 and Θ_1 are parameters that can be tuned to increase the model's performance (Cox, 1958). The first parameter Θ_0 is called the bias and the second parameter Θ_1 is a vector containing the weights of the instances (Murphy, 2022, p.337). The *logit function* gives the odds that the label y_n equals 1. As seen in Eq. 10 above, the logistic regression model's logit, or odds, are linear in $\boldsymbol{x}$. The linearity of the Logistic regression model is dependent on the decision boundary it creates between classes, which in turn is determined by the weight vector Θ_1 and the bias Θ_0. The vector defines the direction of the decision boundary and the magnitude of the vector controls the confidence of the predictions (Cox, 1958).

The probability that an instance belongs to class 1 and 0 are given by the logistic functions (Cox, 1958)

$$\Pr(\mathrm{y_n} = 1 |\, \mathbf{x_n}, \boldsymbol{\Theta}) = \frac{\mathrm{e}^{\Theta^\mathrm{T}\mathrm{x_n}+\varepsilon}}{1+\mathrm{e}^{\Theta^\mathrm{T}\mathrm{x_n}+\varepsilon}} \quad \text{(Eq. 11)}$$

and

$$\Pr(\mathrm{y_n} = 0 |\, \mathbf{x_n}, \boldsymbol{\Theta}) = 1 - \Pr(\mathrm{y_n} = 1) = \frac{1}{1+\mathrm{e}^{\Theta^\mathrm{T}\mathrm{x_n}+\varepsilon}}, \quad \text{(Eq. 12)}$$

where ε is a nuisance parameter.

The performance of the model is determined by the selection of Θ_0 and Θ_1. Typically, they are calculated using maximum likelihood, by selecting the values of Θ_0 and Θ_1 that maximise the likelihood function

$$L(\Theta_0, \Theta_1) = \prod_{n:\, y_n=1} \Pr(x_n) \prod_{n: y_n=0} (1 - \Pr(x_n)) \,. \qquad \text{(Eq. 13)}$$

The probability that the predicted label is the same as the observed label is maximised in this process (James et al., 2013, p.135).

2.3.5 K-nearest neighbour

K-nearest neighbour, or K-NN, is a non-parametric model known to be one of the simplest machine learning algorithms. It was first introduced by Fix and Hodges in 1951. The idea is that, within a defined neighbourhood close to an instance *x*, the ratio of the average value of the data points close to *x* decides how the instance should be classified. The *k* data points closest to the instance *x* are the *k* nearest neighbours of *x* (Fix & Hodges, 1951). These *k* nearest neighbours are denoted as $NN_k(\boldsymbol{x_n})$ and they make up a region around *x*, for which we can derive a distribution by computing

$$\Pr(y_n = i |\, \mathbf{x_n}) = \frac{1}{k} \sum_{n \in NN_k(x_n)} I(y_n = i), \qquad \text{(Eq. 14)}$$

where *i* are the different labels. The most common label is returned by this distribution and is known as the majority label. This becomes the model's output, *y,* while *i* is the classification we believe the model will settle on based on the labels of the *k* nearest neighbours. E.g., if the five nearest neighbours of the new input *x* have the labels (0, 0, 1, 1, 1), then the prediction can be shown as (Murphy, 2022, p.545-546):

$$\Pr(y_n = 1 |\, \mathbf{x_n}) = \frac{3}{5}. \qquad \text{(Eq. 15)}$$

The two main inputs to the K-nearest neighbour algorithm are the size of the neighbourhood *k* and the function used to calculate the distance between two instances x_i and x_j. A common distance measure is the Mahalanobis distance (Murphy, 2022, p.545):

$$\text{dist}_M(x_i, x_j) = \sqrt{(x_i - x_j)^T \mathbf{M} (x_i - x_j)}. \qquad \text{(Eq. 16)}$$

The constant ***M*** in Eq. 16 is a positive definite matrix. The Euclidean distance is obtained here by letting ***M*** be equal to the identity matrix ***I***.

2.4 Choosing the right model

A central part of any machine learning problem is to identify which model is the most suitable for solving a given problem. Model performance cannot be evaluated using the training data, as that will result in overfitting. As previously explained, a model is overfitting when it is too flexible and too complex, i.e. has a high variance and a low bias. A model suffering from this issue has been fitted to the training data to such a high degree that it fails to identify general patterns in the data set, thus rendering it useless when applied to unseen data (Murphy, 2022, p.12).

The opposite case, when a model is not flexible enough, is called underfitting. The differences between an underfitted (left), an overfitted (right) and a well fitting (centre) model is visualised in Figure 2.4.

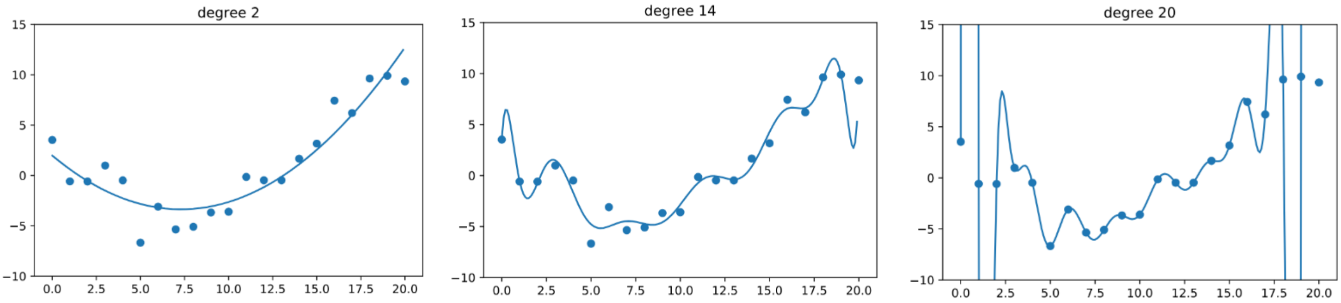


Figure 2.4: Visualisation of an underfitted, a well fitting, and an overfitted model (Murphy, 2022, p.12)

To avoid overfitting, the evaluation of model performance is done on unseen data. Evaluation can be done on training data, using k-fold cross-validation, or using a separate test data set (James et al., 2013, p.30-33).

2.4.1 K-fold cross-validation

During validation, the classifier's ability to adapt to unseen data is evaluated by estimating the test error using the training error. Resampling methods can be used to draw samples from the training data and create new data subsets. The model is then refitted to the subsets iteratively to obtain information from the fitted model. Bootstrap aggregating, which was explained in *Chapter 2.3.2 Random forest*, is one of the most common resampling methods. Another is K-fold cross-validation (James et al., 2013, p.175).

During K-fold cross-validation, the training data is divided into *k* subsets, called folds, of approximately the same size. A model is fitted on *k*-1 of the folds. The last fold, called the *hold-out fold*, is saved for evaluation. From the evaluation, using the hold-out fold, a training error is calculated. The classifier is then retrained on *k*-1 folds, but this time another fold is chosen as the hold-out fold. The process is repeated *k* times until every fold has been used as a hold-out fold. The estimated train errors are then averaged to provide the estimation of the test error. The process can be described through (James et al., 2013, p.181)

$$\mathrm{CV}_{\mathrm{k}} = \frac{1}{\mathrm{k}}\sum_{\mathrm{b}=1}^{\mathrm{k}} \mathrm{train\_error}_{\mathrm{b}}. \qquad \text{(Eq. 17)}$$

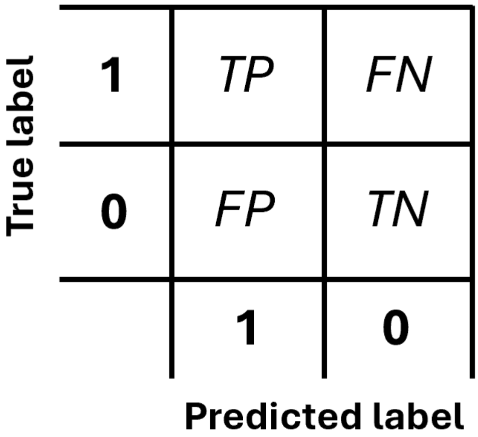
2.4.2 Evaluation

Theodoridis and Tsadiras (2021) performed an extensive review of the previous publications within the customer churn field in 2021. One of the conclusions they reached was that the dominant evaluation metric used to determine the performance of a machine learning model in predictive customer churn modelling was *AUC* (Theodoridis & Tsadiras, 2021). The abbreviation stands for *Area Under the Curve* and it measures the area under the *Receiver Operating Characteristic* (ROC) curve (Mellouk, 2010, p 200). The measurement has been used in medical diagnostics since the 1970’s (Huang & Jing, 2005).

When evaluating the performance of a model, the proportion of correctly classified data must be determined. When performance for a model is evaluated, the classified instances are grouped into four categories. In this example, performance is measured for a model that tries to identify if a customer will churn. The model will classify a customer as either churned or retained. The categories are (Huang & Jing, 2005):

1) True positive (TP) – the customer has correctly been classified as churned.
2) True negative (TN) – the customer has correctly been classified as retained.
3) False positive (FP) – the customer has incorrectly been classified as churned.
4) False negative (FN) – the customer has incorrectly been classified as retained.

The categories are typically displayed in a *confusion matrix* (Murphy, 2022, p. 171), as shown in Figure 2.5.



True label	1	*TP*	*FN*
	0	*FP*	*TN*
		1	0
		Predicted label	

Figure 2.5: Confusion matrix (Liljestam & Lindell, 2024)

Let N_C be the number of churned customers, N_R the number of retained customers, $n_{c\to C}$ the number of customers correctly classified as churned (TP), $n_{r\to C}$ the number of customers incorrectly classified as churned (FP), $n_{r\to R}$ the number of customers correctly classified as retained (TN) and $n_{c\to R}$ the number of customers incorrectly classified as retained (FN). The true positive rate (TPR) is the proportion of correctly classified churned customers (Mellouk, 2010, p 201):

$$\mathrm{TPR} = \frac{\mathrm{n}_{\mathrm{c}\to\mathrm{C}}}{\mathrm{N}_{\mathrm{C}}}. \qquad \text{(Eq. 18)}$$

The true negative rate (TNR) is the proportion of correctly classified non-churned customers (Mellouk, 2010, p.200-201):

$$\mathrm{TNR} = \frac{\mathrm{n}_{\mathrm{r}\to\mathrm{R}}}{\mathrm{N}_{\mathrm{R}}}. \qquad \text{(Eq. 19)}$$

The false positive rate (FPR) is the proportion of customers who have been incorrectly classified as churned (Mellouk, 2010, p.201):

$$\mathrm{FPR} = \frac{\mathrm{n}_{\mathrm{r}\to\mathrm{C}}}{\mathrm{N}_{\mathrm{R}}}. \qquad \text{(Eq. 20)}$$

The false negative rate is the proportion of customers who have been incorrectly classified as non-churned (Mellouk, 2010, p.201):

$$\mathrm{FNR} = \frac{\mathrm{n}_{\mathrm{c}\to\mathrm{R}}}{\mathrm{N}_{\mathrm{C}}}. \qquad \text{(Eq. 21)}$$

As the goal of a classifier is to accurately predict the label associated with each instance, the true rates, TPR and TNR, should be high, and the false rates, FPR and FNR, should be low (Bell, 2015, Chapter 1).

Although AUC proved to be the most useful evaluation metric, it is not the most common. The performances of classifiers are typically measured using *accuracy* (Huang & Ling, 2005). The predictive accuracy is calculated through (Mellouk, 2010, p.201):

$$\mathrm{accuracy} = \frac{\mathrm{n}_{\mathrm{r}\to\mathrm{R}}+\mathrm{n}_{\mathrm{c}\to\mathrm{C}}}{\mathrm{N}_{\mathrm{R}}+\mathrm{N}_{\mathrm{C}}}. \qquad \text{(Eq. 22)}$$

A weight can be introduced in the numerator to increase the impact of $n_{c\to C}$ or $n_{r\to R}$ on the final accuracy. When Huang and Ling (2005) made a formal comparison of AUC and accuracy, they concluded performance is better captured by AUC than accuracy, since the ROC curve, which decides the size of the AUC, compares the model's performance across all class distributions and range of error costs. Accuracy, on the other hand, fails to capture this information (Huang & Ling, 2005).

The TPR is plotted on the Y-axis and the FPR on the X-axis to construct the ROC curve, as depicted in Figure 2.5. Figure 2.5 illustrates four different ROC curves. In this example, the ROC curves of classifiers A and B are to the left of classifier D:s curve evaluated at every instance, meaning that the classifiers for A and B have a lower expected cost than D for all possible class distributions and error costs. It is thus instantly evident that classifiers A and B outperform classifier D. In situations where one curve does not dominate another, performance can be measured by comparing the respective AUC:s (Huang & Ling, 2005). Such analysis is necessary when comparing the curves of classifiers A, B, and C. The AUC falls within the interval [0, 1], with 1 being the best possible classifier and 0 being the worst (Nahm, 2022). Obtaining an AUC of 0.5 is similar to making a classification by chance and implies that the classifier has no information that it can use to distinguish between the two labels (Holzinger, 2022, p.105). The ROC curve for which the AUC is 0.5 follows the left diagonal (Nahm, 2022), visualised as a dotted line in Figure 2.5. By comparing the dotted line and the ROC curve of classifier D, a conclusion can be drawn that classifier D performs worse than chance for most class distributions.

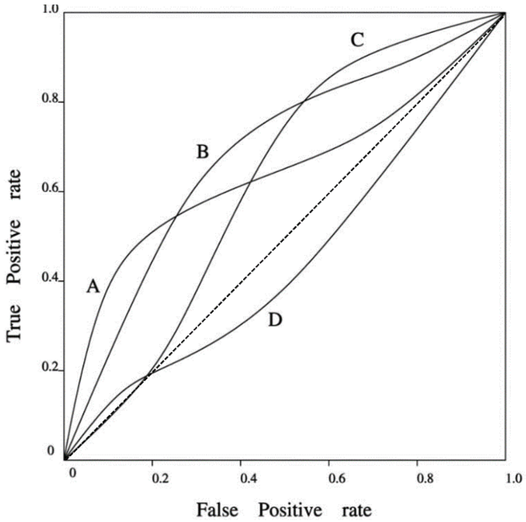


Figure 2.5: Examples of ROC curves (Huang & Ling, 2005 (dotted line added by Liljestam & Lindell, 2024)).

Mandrekar (2010) and Nahm (2022) explain how to interpret an AUC performance. They both agree that performance above 0.7 is to be considered a positive result, but use different adjectives to describe it. Their descriptions are compared in Table 2.1.

Table 2.1: Adjectives describing different intervals of AUC

AUC	**Mandrekar (2010)**	**Nahm (2022)**
0.7 - 0.8	Acceptable	Fair
0.8 - 0.9	Excellent	Good
>0.9	Outstanding	Excellent

3. Method

In this chapter, we describe how the research project outlined in this thesis was executed. The project consisted of three main parts. Initially, we defined customer churn for the research case; a company which offers its customers non-contractual products within the winter tourism industry. Secondly, we describe the data management, which took up a significant part of the allocated project time as it required a lot of reflection and discussion with the company to understand and select relevant features. Lastly, the machine learning models were designed, trained, validated, and evaluated on the data. Additionally, the models were tuned to increase performance.

3.1 Project execution

We executed the project in two environments: *SQL Server Management Studio (SSMS)* and *Jupyter Notebook*. SSMS allows for interaction with SQL databases and was thus used to access the company's database during the data selection phase. Using SQL programming, the existing data was collected, reorganised, and transformed into tables that were later used for model construction. We performed some data pre-processing in SSMS, but the majority of pre-processing was done in Jupyter Notebook after having exported the tables from SSMS. Jupyter Notebook is an open-source web-based application designed to be compatible with different languages and provide interactive services for data science (Silaparasetty, 2020). Although we used some SQL programming in Jupyter Notebook, most commands were executed using Python. Python is commonly used to perform data analysis since it supports many powerful open source libraries (McKinney, 2012, Chapter 1). Jupyter Notebook hosted the design, training, validation, and evaluation of the five machine learning models.

The four main libraries used in Jupyter Notebook to enable the execution of the project were *JupySQL, NumPy, pandas,* and *scikit-learn.*

JupySQL is a library which allows for SQL programming in Jupyter applications such as Jupyter Notebook. It utilises so-called magics to enable the execution of SQL commands. Magics are a feature of the IPython kernel that powers a Notebook. Line magics are denoted by a percent sign '%' and using the specific line magic "%sql" allows for one line of SQL commands to be executed. Cell magics are denoted by double percent signs "%%" and "%%sql" allows for an entire cell of SQL code.

NumPy is a Python library that provides, among other things, efficient multidimensional array objects and other derived objects, which enables fast and efficient computations of data. The name is short for Numerical Python and as it suggests, the library is used for scientific computing. The library also provides methods for performing operations with arrays, element-wise operations within arrays, linear algebra operations, and generation of random numbers, to name a few. NumPy arrays are commonly used to contain data when moving it between algorithms and libraries, due to its efficiency compared to other libraries supported in Python (McKinney, 2012, Chapter 1).

The pandas library is the result of the open-source pandas project which is developed by a community of contributors. The library is used to perform data analysis in Python and includes tools for reading and writing data in different formats, aligning data, and handling missing data. Additionally, data sets can be pivoted and reshaped, sliced, indexed, merged, joined, and resized using the DataFrame object and functions the pandas library provides. The idea behind pandas is to combine the philosophy of computing with multidimensional arrays from Numpy with data manipulation capabilities from relational databases, such as SQL databases. The name comes from "panel data", which McKinney (2012) describes as an econometrics term used for multidimensional data sets and is a reference to Python data analysis.

Scikit-learn is a machine learning toolbox in Python used for implementing machine learning algorithms. It is tightly integrated in the Python ecosystem and therefore compatible with other applications outside the typical range of statistical data analysis. Scikit-learn supports both supervised and unsupervised learning. The tool kit was written in a high-level language for ease of use, which combined with a relatively simple interface makes it accessible to laymen and experts in fields outside of computer science, according to Pedregosa et al. (2011). The toolkit is built on existing libraries in Python; NumPy, SciPy, and Cython. NumPy is the base structure for data and model parameters, SciPy provides efficient algorithms for matrices, statistical functions, and linear algebra, and Cython is used to combine C in Python and makes it possible for scikit-learn to increase performance to the level of compiled languages, despite being a high-level language (Pedregosa et al., 2011). An overview of the libraries used in the project is provided in Table 3.1.

Table 3.1: Libraries used in the project

Library	**Language**	**Description of use**
JupySQL	SQL	Used to allow for SQL programming in Jupyter Notebook.
NumPy	Python	Used to make numerical calculations.
pandas	Python	Used to create and manipulate DataFrames, which contained the data used in the project.
Scikit-learn	Python	Used for the design, training, validation, and evaluation of the machine learning models.

3.2 Definition of customer churn

It is difficult to intuitively define at what exact point a customer churns in a non-contractual research case (Reichheld & Sasser, 1990). To define churn in this project, we needed to understand the literature on the subject, but also what would be the most fitting definition for our specific research case. Thus, our research strategy for defining churn included both conducting a literature review and conducting brainstorming sessions and meetings with the company. In this subchapter, we describe the part of the literature review that concerns the definition of customer churn.

3.2.1 Research strategy: literature review and brainstorming sessions

In order to understand the research field we are contributing to with this thesis, the project started with a literature review. A literature review is made by reviewing the current literature in the research area that one is investigating. Initially, it helps to gain an overview of the field and to generate and refine ideas. The next part of the literature review is called the critical literature review and it is a deep dive into the research field. The relevant information found during the critical review becomes part of the research paper in the end. The critical review is also used to discover what is known and what is not known within the research field (Saunders et al., 2012, p.70-73). According to Funck and Karlsson (2021, p.9-10), when working with a research field that has existed for some time, but where few academic studies have been made, it is necessary to compile what research has been conducted in the field thus far. On the other hand, when many academic studies have been made in a field, it is necessary to review large quantities of studies in order to find patterns and themes in the research (Funck & Karlsson, 2021, p.9-10). In our case, customer churn is a well-researched field, but research on predicting customer churn in a non-contractual setting is scarcer.

When performing a literature review, relevant sources can be primary, secondary, and tertiary. Primary sources include reports, theses, company reports, conference proceedings, and the like, while secondary sources include journals, books, newspapers, and government publications. Tertiary sources include different kinds of search tools for finding primary and secondary sources, such as databases, indexes, catalogues, and dictionaries. Refereed academic journals are the most useful types of sources in literature reviews, according to Saunders et al. (2012, p.82-83). We mainly sought academic research papers and published books within the research field to see what research had been made in published previous work thus far. We mainly made use of secondary sources in our literature review and used tertiary sources to locate them.

The literature review on customer churn was conducted in parallel with brainstorming sessions and informal meetings with the company. The information provided by the company during these sessions served in this project as a primary source. Brainstorming as a technique is used to generate and refine ideas, either in a group setting or alone. The different steps of a brainstorming session are; *finding a definition for the problem*, *asking others for suggestions that relate to the problem*, *recording those suggestions*, *reviewing the recorded suggestions*, and finally *analysing the suggestions to decide which ideas are the most appealing* (Saunders et al., 2012, p.36).

The brainstorming sessions were performed in a group setting and contained all the different steps above, although the steps overlapped in practice and the process was not linear. The purpose of the sessions and the additional informal meetings was to answer the question; *How can we define customer churn for SkiStar?*, which in turn would yield an answer to the research question; *How can non-contractual customer churn be defined to enable prediction of future churn for a company in the winter tourism industry?*

For such complex products as the ones offered by the company that this research case centres around, we found that there are several possible definitions of customer churn. The different aspects we had to consider when formulating our definition is further explained in the *Result* chapter and in Subchapter *3.3.1 Data selection* of this thesis. We sought to encapsulate these different aspects into our definition, while at the same time trying to not make the definition overly complex, since the case was already complex enough to begin with. Considering this, and the unintuitive nature of when non-contractual churn occurs, we decided to explore two different definitions instead of one. Testing two definitions would allow us the opportunity to compare our results and show that customer churn can be defined in several ways when dealing with a complex non-contractual research case. It would further allow us to test whether a more complex definition or a less complex definition would make our models perform better. Additionally, it would enable us to balance the importance of the industry knowledge acquired during the research project with the theoretical background. The two final definitions are stated in our *Result* chapter and discussed in our *Discussion* chapter.

3.2.2 Critique of the churn definition

Our research question – *How can non-contractual customer churn be defined to enable prediction of future churn for a company in the winter tourism industry?* – indicates several different definitions should be investigated for the question to be answered in a satisfactory way. Had the question lacked the initial *how*, and instead been phrased as *Can customer churn be defined for a company in the winter tourism industry offering non-contractual products?*, the answer would have been a simple *yes* and we could have showcased this by including our one definition as proof. However, including *how* offers the opportunity to test different definitions of how customer churn can be defined for this research case. Investigating one more complex and one less complex definition allowed us to test how the complexity of the definition impacted the results of the models. Comparing definitions also allowed us to showcase that crafting the best possible definition takes work and careful consideration, as well as a deep understanding of the research case that one is studying.

Still, within the scope of our research project (time-wise and width-wise), it was not possible to test every possible definition that was considered and discussed during the research project. The definitions that were finally chosen were decided to be the most ideal ones, but it would have been interesting to test more definitions of varying complexity and consider different aspects, had the scope allowed it.

3.3 Data

Data used in the construction of the predictive models in this project was collected from the company's internal database. Not all the data in the company's database was considered relevant for the research case of predicting customer churn, and the features that ended up in the final tables were handpicked from the database by us. To comply with confidentiality requirements set by the company, some feature names have been changed and some will not be revealed or discussed in this thesis. Further explanations are provided in Chapter *3.3.1. Data selection*. In this *Data* chapter we explain how we handled the data to make it clean enough to insert into our machine learning algorithms.

3.3.1 Data selection

The goal of the data selection was to put together a table of relevant features that could be used to predict customer churn. We needed to select data that would allow us to determine which examined customers had already fulfilled our criteria for churn and which customers have not. For the latter type of customer, a prediction would be relevant, while for the former, it would not be necessary. We only used pre-existing data from the company and we have not produced any data ourselves. As instructed by the company upon the start of the project, we only considered data gathered by the company over a time period of five years.

In order to select the data that would be the most useful for our case, we examined the company's database in its full scale, but with the aim to find relevant features to perform our churn prediction. Which features would be relevant was decided by us through exploration and in accordance with the company's suggestions. We also consider features that previous studies have found interesting, as described in Chapter *1.1 Previous studies on predicting customer churn* (Durun-Cengizci & Caber, 2024). We also imposed conditions on what type of customers should be included in the data set. We balanced the desire to be able to predict churn for the largest subset of customers possible with an understanding that a too wide basis for the model might decrease performance due to increased variance in the data. The selection was made in discussions with company representatives.

We used the SSMS environment and SQL programming to explore the company's database and construct new tables. The existing tables from which we utilised features contained information on customers, reservations, activities and purchases. Features were connected to a customer ID and could thus be combined using join-commands in SQL to form our table.

The target variable would state if that specific customer would churn or not, which made it a requirement to only have one row per customer in our table of features. In the existing tables of reservations, activities, and purchases, this posed a problem, as several rows were connected to the same customer ID. We tackled this issue in different ways. For purchases, we created a variable that simply counted if a customer had performed a purchase in the company’s shop. For reservations, we included information only about the latest reservations. Activities were used to categorise customers into different groups based on patterns in their visits to the ski resorts, so called visit cycles. These visit cycle features are described in more detail in the *Result* chapter of this thesis.

We considered a list of requirements that each customer had to fulfil in order to be included in the data set. These requirements were made to clear out potentially noisy customers that might disturb the model performance and to avoid making the model too complex. The requirements were decided in consensus with experts from the company and their knowledge regarding their own data. When constructing the tables, we considered the following requirements:

1) A customer cannot be a corporate customer, they must be a private customer.
2) A customer cannot only have visited a ski resort during the summer season. By this, we mean we only looked at customer churn related to the winter season. The company also offers activities at their locations in the summer, but these activities were excluded from the data set, since the company mainly offers winter related activities. The customer having previously made a visit to one of the locations during the summer season was, however, included as a feature in the data set, to see if it has an effect on customers churning from the winter related part of the business.
3) A customer has to have visited one of the company’s locations at least once. A visit is defined as having performed an on-site winter activity. We did not consider customers who only had a future reservation without having visited. We did this because we decided the customer needs to have actually experienced the product to be able to churn. Only having made a purchase in the company’s shop also does not count as having visited, but we consider having previously made a purchase as a feature to predict churn instead.
4) A customer cannot previously have fulfilled the churn criteria. Meaning, a customer cannot already have churned when we predict whether they will churn or not. It is not interesting to make a prediction for a customer that by definition has already left the company, since that customer is not the right customer to target with retention strategies.
5) A customer cannot permanently reside on or too close to the ski resorts, meaning the customer’s address cannot share the same zip code as any of the resorts. This condition is enforced as the behaviour of customers who live close to the destinations, according to company experience, differ significantly from those who live further away.
6) A customer cannot own their own accommodation on the resort. Similar to the previous condition, this is enforced as the behaviour of customers who own their own accommodation differ from those who do not.

3.3.2 Dividing data into two data sets

An important aspect of the table construction was the confidentiality requirements requested by the company. Discussions were held with the company about which features could be included in the thesis, which feature names had to be changed, and which features had to be completely anonymised. We managed this by creating two tables. The first table we created was the *CompleteDataSet* (CDS). The CDS contained all the features that were selected from the company database to be used during the project, even the ones that had to be anonymised. Thus, the contents of this data set cannot be disclosed or discussed in this thesis.

The second data set we created was the *ReducedDataSet* (RDS), a subset of the data in the CDS. This table only contained the selected features that did not have to be anonymised, which allows us to discuss them in detail. Table 3.2 lists the features included in the RDS. The excluded features from the CDS are addressed in the final row of the table under the *Feature name* called *Anonymous features*.

Table 3.2: Features included in the project

Feature name	Data type	Description
IsEmailAddressCorrect	Boolean	Is true if it is confirmed that the email address the customer uses when making a reservation is correct.
IsMarketingCommunicationOk	Boolean	Is true if the customer has approved to receive marketing communication.
IsSmsMarketingCommunicationOk	Boolean	Is true if the customer has approved to receive marketing communication, specifically via SMS.
HasValidMobilePhoneNumber	Boolean	Is true if the phone number that the customer gave when they made a reservation is correct.
HasShopPurchase	Boolean	Is true if the customer has purchased physical products, such as ski equipment, from the company shop.
HasCancellation	Boolean	Is true if the customer has previously cancelled a reservation.
NoOfReservationPurchaseOccations	Integer	The number of times a customer has made a reservation with the company.
TotalReservationSpend	Float	The total cost in SEK of all reservations that a customer has made with the company.
HasVisistedInSummer	Boolean	Is true if a customer has visited a ski resort during the summer season, when activities such as biking and swimming are offered.
ProductsIncludedInLastReservation	Boolean	A collection of features that state whether or not a certain product was included in the last reservation made by the customer.
MunicipalityType	Boolean	A collection of features that describe what type of municipality the customer resides in.
Origin	Boolean	A collection of features that state in what country or region of the world the customer resides in.
HadVisit	Boolean	A collection of features that state what year the customer performed an on-site winter activity at a ski resort owned by the company.
Churn1	Boolean	The target variable calculated from the first definition of customer churn.
Churn2	Boolean	The target variable calculated from the second definition of customer churn.
Anonymous features	-	A collection of anonymous features. No description provided due to confidentiality requirements. These features are only included in the CDS.

3.3.3 Data pre-processing

We pre-processed the selected data in various ways. Some of the processing steps were performed in Jupyter Notebook on our finalised tables, while other processing steps were made during the construction of the tables in SSMS. Data can be processed in many ways in order to prepare it for insertion into machine learning algorithms, but which steps are actually necessary for certain projects may vary. We researched which pre-processing steps had been used in previous work to predict customer churn, where the most important ones with a clear theoretical motivation have been described already in Chapter *2.2.3 Data treatment*. Other pre-processing steps have been performed based on expert advice given by the company.

The steps of processing the data in this research case are visualised in Table 3.3. The table illustrates the different steps, a description of what they entail, if they were done in SSMS or Jupyter Notebook, and the reason why we chose to perform this specific step.

Table 3.3: Pre-processing steps and their respective descriptions

Pre-processing step	**Description**	**Program used**	**Motivation**
Removal of customer groups	We removed some customers from our data set completely, e.g. company customers and people living on site of the alpine mountain resorts.	SSMS	Some customers had to be removed since there was a risk of them being drastically different from other customers in their behaviours, meaning they could confuse our models unless they were removed.
Feature manipulation	Some features were manipulated by combining them with information from other features. Other features were remade from categorical to boolean, where only the fact that the feature had a value was relevant rather than what the value actually was.	SSMS	Some features were not represented the way we needed them to be and hence they were manipulated.
Feature creation	Some features had to be created by writing extensive Python code and using already existing features in new ways, in order to get the features we wanted.	Jupyter Notebook	The feature *VisitCycle*, further explained in the *Result* chapter, had to be created like this. It was important to create for the sake of our churn definition.
Creation of new categories	For some features, we created broader categorical groups that we sorted detailed categorical values into.	SSMS	The values of some features could assume more detailed categories than we were actually interested in, hence we aggregated these features by condensing the amount of categories these features had.
Dropping columns	Some features were removed completely (i.e. dropped) from our data sets.	Jupyter Notebook	Some features, like different ID:s, were added in SSMS to combine features from different tables. These features had to be dropped before inserting our data sets into our machine learning algorithms, since they would not follow any real patterns and would only confuse the models if they were kept.
Outliers	Outliers that were 3 standard deviations away from the mean were removed in Jupyter Notebook using the z-score. The number 3 was chosen after visualising the data.	Jupyter Notebook	Some features contained outliers, i.e. values that drastically differed from the mean. These values were so extreme that they could skew our models if kept around. Since there were so few of them, removing the customers related to these outliers did not noticeably reduce the amount of data we had at our disposal.

Conversions	Some features were converted from objects to numerics, or vice versa. Some of the numerics were then turned into booleans.	Jupyter Notebook	Data had to be either boolean for categorical values, or numerical for values where the specific amount was relevant for the evaluation. If the data type of a feature was not suitable, it was converted.
Dummy variables	Dummy variables were created for features with categorical values.	Jupyter Notebook	For the same reason we did conversions, we also created dummy variables for some of our features. This created new features where each categorical value of the original feature became a boolean variable. This was necessary for our models to interpret the information given in the feature correctly.
Normalisation	Normalisation was made in Jupyter Notebook and used for all our machine learning models. Normalisation was performed by min-max rescaling the values of numerical features.	Jupyter Notebook	Some features with numerical values varied greatly in size. Rescaling these features through normalisation would make the performance of our machine learning models better.
Balancing data	We balanced our data using the Synthetic Minority Oversampling Technique (SMOTE).	Jupyter Notebook	We balanced our data to avoid issues that could arise from our target variable being imbalanced. A heavy imbalance between the number of churners and non-churners could otherwise lead to bias in the models.
Correlation	We checked for correlation between our features and the target variable. We visualised the correlation using a bar plot.	Jupyter Notebook	We checked to see if there was any correlation reaching 1.0 or suspiciously close to it, meaning we might have included customers for which prediction was unnecessary.
Feature reduction	We performed feature reduction by looking at SHAP values and removed features that were not important for the prediction.	Jupyter Notebook	We visualised the SHAP values of the data sets to see which features were the most and least important to the prediction.
Missing values	We removed instances with missing values through listwise deletion.	Jupyter Notebook	We removed instances that contained missing values for some of its features. Since the data sets were sufficiently large, this deletion did not have any significant impact on the overall size of the data sets.

3.3.4 Critique of the data management

A major part of the data management was identifying which features could be included without creating unwanted scenarios where prediction was not necessary, as there would be a 100% correlation between a feature and churn for certain customers. This would result in an unreasonably high performance and a model which did not make actual predictions. In other words, the outcome of the whole project would be compromised. Hence, we spent much time discussing the case, different types of customers and how their behaviour affected features to avoid this.

Even with expert advice from the company considered, the final feature selection was still performed by us; individuals with limited knowledge of the winter tourism industry and the behaviour of its customers. Because of this, it is possible that we were unsuccessful in selecting the most relevant features. This might have resulted in a decrease in model performance, which could have been avoided if we had more time to try different constellations of features.

There are many different pre-processing steps one might choose to perform. Performing them might sometimes increase the performance, but one can only know for certain the steps have helped by comparing the precision of the models when both including and excluding the steps. For this project, we have included many pre-processing steps based on research, logic, and their theoretical relevance to our project. Several of them we have included without comparison to how the models performed when the steps were excluded instead. The decision to not make several of these comparisons was made since it would be too time consuming to run all of our models with all the different versions of the data that would arise. Instead, we researched which pre-processing steps would likely improve the precision of our models the most and trusted this choice.

The fact that we only had access to five years of historical data is something that might have affected the results of our classifiers, specifically when considering the visit cycles. Five years can be considered too short of a time to observe customer patterns and behaviours for infrequently purchased products that depend on seasonality, such as the products in this research case. We acknowledge that this might have impacted the results of our machine learning classifiers and that having more historical data to work with would have been preferable, had it been possible.

3.4 Creation of the machine learning models

After the data had been pre-processed, machine learning models were constructed. We performed a literature review to select which models were to be used and constructed them in Jupyter Notebook. We labelled the data ourselves and utilised supervised learning. The model selection and construction is explained below.

3.4.1 Model selection

The selection of the machine learning algorithms that we used during this project was based on the results of a literature review, as previously mentioned. The review investigated how frequently the algorithms occurred and their performance in previous studies. The studies were identified through the keywords *machine learning*, *customer churn,* and *predictive customer churn* using the library of Uppsala university and the database *Web of Science*. The latter visualises the number of citations, which enabled us to select articles that had many citations, which we used as an indicator that they were significant in their field.

We found supervised learning to be the most appropriate approach to address this particular case, because we were able to create labels for our data as soon as we had a definition for customer churn. In its essence, our problem is a classification problem, where we want to use the different features of the customers in our data set to predict the outcome of our target variable, which is either *churn* or *non-churn*. Thus, we utilise models that are suitable for classification. We found that the machine learning models that have had the highest performance were Decision tree classifiers and ensemble methods based on the Decision tree classifiers. Specifically, Random forest stood out as an effective bagging method and XGBoost as a high-performing boosting method (Wu et al. 2022, Rajendran et al., 2023, Ahmad et al., 2019, Liu et al., 2023, Lalwani et al., 2022, Coussement & Van den Poel, 2008, Yizhe et al., 2017, Theodoridis & Tsadiras, 2022, Imani & Arabnia, 2023, Geiler & Nadif, 2022, Lee et al., 2019).

As simpler algorithms than tree-based ones had also been frequently occurring in previous studies (Buckinx et al., 2005; Huang et al, 2012; Imani & Arabnia, 2023; Lalwani et al., 2022; Liu et al., 2023; Li et al., 2021; Yang et al., 2019), we decided that we would include two additional classifiers; Logistic regression and K-nearest neighbour.

We also explored a simple baseline classifier to make sure that the performance achieved by the other classifiers were sufficiently high. We reason that a machine learning model that performs similar to or worse than a baseline classifier is not interesting to use, since the baseline classifier is much more efficient. To warrant the extra cost of using machine learning, the performance must increase. The baseline classifier used in this project is a majority vote classifier which assigns all customers to the majority class. The accuracy is dependent on how many customers belong to the majority class. If 70% of all customers churn, then predicting churn for all customers will result in an accuracy of 70%. In this example, the other classifiers must achieve an accuracy score higher than 70% in order to be deemed valuable.

3.4.2 Model construction

The selected models were trained and tested on the pre-processed data. Model construction was performed in a similar structure for all models. We created two classifiers per algorithm, one for each definition of customer churn. The classifiers were created and fitted to the balanced training data. They were then validated using 10-fold cross-validation to determine their performance using the train data. Hyperparameters were also optimised and in almost all cases we utilised the *GridSearch* algorithm for hyperparameter optimisation. In some cases, the algorithm was not supported for the particular classifier and instead the hyperparameters were optimised using a more manual approach. When the classifiers had been optimised, their performance was evaluated by using the test data to calculate the respective AUC and accuracy. In accordance with the findings of Huang and Ling (2005), we use AUC as our main evaluation metric and accuracy as a complimentary metric. A more detailed description of the construction of each model can be found in Table 3.4. It includes the machine learning algorithms, the libraries that were used, the validation techniques, and the tuned hyperparameters and the optimisation that was made.

Table 3.4: Construction of the machine learning models

Machine learning algorithm	**Libraries**	**Validation technique**	**Hyperparameters and optimization**
Decision tree	*sklearn.tree* *sklearn.model_selection* *sklearn.metrics*	10-fold cross-validation	GridSearch optimisation for: *Criterion, splitter, max_depth, min_samples_leaf, min_sample_split*
Logistic regression	*sklearn.linear_model* *sklearn.model_selection* *sklearn.metrics*	10-fold cross-validation	GridSearch optimisation for: *Penalty, dual, solver, max_iter*
Random forest	*sklearn.ensemble* *sklearn.model_selection* *sklearn.metrics*	10-fold cross-validation	GridSearch optimisation for: *N_estimator, criterion, min_samples_leaf, max_features*
XGBoost	*xgboost* *sklearn.model_selection* *sklearn.metrics*	Built-in validation	Optimised for: *Tree_method, early_stopping_rounds*
K-nearest neighbour	*sklearn.neighbors* *sklearn.model_selection* *sklearn.metrics*	10-fold cross-validation	GridSearch optimisation for: *Weights, metric* Optimised hyperparameter: *n_neighbors*

3.4.3 Critique of the creation of the machine learning models

The basis of the model selection was an exploration of previous work performed to predict customer churn. The extensiveness of the literature study reflected the importance of the model selection for this particular case. The argument can be made in any literature study that it is not vast enough to capture variations in existing literature. We do not find this critique to apply in this case, since any limitation in scope of the literature study reflected the fact that finding the optimal classification algorithm for non-contractual churn was not the main focus of the project.

The creation of machine learning models follow a fairly standardised procedure. We used libraries for classifiers that are commonly used in the field and thus do not warrant much scrutiny. The selection of validation and hyperparameter optimisation techniques are also not unusual and thus we do not see a high risk of misjudgement when selecting these techniques. The optimisation of the models is where we are most critical of our own work. Since it is not feasible to explore every possible combination of parameters to achieve optimal results, there is a risk that we did not explore the optimal combination of parameters, leading to us being unable to determine peak performance. By using an optimisation algorithm, this risk is somewhat decreased, even though it is still existing.

4. Result

In the following chapter, the results derived from this project are presented. The results are closely connected to our research questions; *How can non-contractual customer churn be defined to enable prediction of future churn for a company in the winter tourism industry?* and *How can supervised machine learning be used to predict non-contractual customer churn in the winter tourism industry?* Subchapter *4.1 Definition of the churn criteria* presents our results connected to the first question, while Subchapter *4.2 Results of the machine learning models* is connected to the second research question.

4.1 Definition of the churn criteria

To answer the research question; *How can non-contractual customer churn be defined to enable prediction of future churn for a company in the winter tourism industry?*, we produced two definitions of customer churn specific to our research case. Other definitions would have been possible, but these two definitions were chosen as one captures the complexity of the non-contractual product by making use of industry knowledge and the other follows the trend of previous studies of customer churn. The definitions differ in their inherent complexity, as the first definition is more complex in its phrasing than the second. For both definitions, we consider a customer having performed an on-site winter activity at one of the company's ski resorts as the customer having made a visit. In the following subchapters, we explain the definitions, their construction, showcase their differences, and explain how the same customer can churn according to one definition, while it might not churn according to the other.

4.1.1 Definition 1 – built on cyclic behaviour

The first definition of customer churn that was made for this research case was developed using industry knowledge from the company about their products and the complex behaviours of their customers. The goal with this definition was to incorporate industry knowledge when crafting the definition. We did this to evaluate whether a non-contractual churn definition might require more than just looking at inactivity, which is the recommended approach to define non-contractual churn in the theoretical background. By incorporating industry knowledge, we believed we would better capture the complexity of the research case. Our first developed definition of customer churn for this specific research case, from now on referred to as Definition 1, is defined as:

A customer can churn in the following two ways:

1) **A customer who is expected to perform** an on-site winter activity[6] on any of the company's alpine mountain resorts before the end of next year's winter season churns if they do not perform such an activity, as long as they do not have an upcoming reservation for an on-site winter activity on any of the company's alpine mountain resorts.
2) **A customer who is not expected to perform** an on-site winter activity on any of the company's alpine mountain resorts before the end of next year's winter season churns after two winter seasons of inactivity, as long as they do not have an upcoming reservation for a winter activity on any of the company's alpine mountain resorts.

Definition 1 is built on verifying three aspects:

1) Checking if the customer has an upcoming reservation or not. If they have one, they cannot churn. If they do not, they are at risk of churning.
2) Checking for visit cycles in the customer's previous travel-related behaviour to determine whether they are expected to perform any on-site winter activities during the next winter season or not. If they are not expected to perform a winter activity based on previous cyclic behaviour, they cannot churn. If they are expected to perform a winter activity and they do not, they churn.
3) Checking if a customer performs any on-site winter-related activities during the upcoming season, whether they were expected to perform one or not. If they perform a winter activity, they cannot churn. If they do not perform a winter activity, their previous cyclic behaviour impacts whether they churn or not, based on if they were expected to perform a winter activity or not.

The most complex aspect of Definition 1 is the consideration of patterns and *visit cycles* in the customer's behaviour. This consideration was made based on the advice of the company. The company had taken notice that their customers displayed certain cyclic behaviours, meaning patterns, when interacting with the company's products and services. Including these visit cycles in Definition 1 would capture some of the complexity in how customers churn or do not churn from the company. The purpose of the visit cycles was to enable Definition 1 to more closely mimic reality, by allowing for different expectations for customers with different visit cycles. As previously mentioned in Chapter *3.3.1 Data selection*, we have looked at data gathered by the company over a time period of five consecutive years. When customers did not exhibit a consistent cyclic behaviour during the entire five year period, we instead checked to see if they exhibited a cyclic behaviour in the last three years of the period. If the customer did, they were assigned a cycle based on the three year period instead. This is the case for the customer "Klara" in Figure 4.1, visualised further down in this subchapter. In Table 4.1, the possible visit cycles that a customer may be assigned are listed.

[6] Performing an on-site winter activity is what we consider to be a visit to a ski resort.

Table 4.1: Customer visit cycles and their respective meanings

Visit cycle	Description
C0	C0 means that the customer has not exhibited a cyclic behaviour. A customer is assigned the visit cycle C0 when they have performed an on-site winter activity once in the last three years. For a C0 customer to churn, two winter seasons must have passed since their last visit. Customers who have been assigned C0 earlier than three years ago are labelled as REMOVE instead, since they have already churned in a prior year than the year we are observing.
C1	When a customer is assigned C1 as a visit cycle, they are expected to return every year. A customer is assigned the visit cycle C1 when they have performed an on-site winter activity for at least two consecutive years, thus a customer must have visited at least twice to be assigned a C1 cycle. If the customer does not return every year, they are considered to have churned, since they have broken their cyclic behaviour.
C2	When a customer is assigned C2 as a visit cycle, they are expected to return every other year. A customer is assigned the visit cycle C2 when they have performed an on-site winter activity at least twice, with one year between the visits to a destination. A customer must have visited at least twice to be assigned a C2 cycle. If they break this pattern by visiting two years in a row instead, they are reassigned as having a C1 visit cycle. If they break the pattern by arriving less often than every other year, they are considered to have churned, since they have broken their cyclic behaviour and they have not achieved a new cycle.

The number of customers having visit cycles of every three years (C3) and every four years (C4) were also calculated. These customers were found to be very few in numbers. They were also not compatible with the decision that it should take two years of inactivity at most (for C0 and C2 customers) to churn. Had we considered C3 and C4 visit cycles, it would have taken three and four years respectively before we could determine whether those customers had churned or not. This was also not compatible with the limit of only observing customers over a five year time span in total. The decision was hence made to remove C3 and C4 customers. Customers who had churned during previous years were also removed from the data set, since they would automatically be categorised as having churned and are not of interest to make a prediction for.

In the following Figure 4.1, we visualise how different visit cycles work in practice on five example customers, showing how customers with different behaviours are predicted as churned. We observe if the customers churn at the end of the winter season during Year 5, which is the year of prediction. Customers having made a visit to a ski resort during a specific year are marked in turquoise, while years when the customers did not make a visit are white. How the visit cycles change over the years, as the customers exhibit new behaviours, is also visualised by the cycle names changing for the same customer. A dash indicates that the customer was not a customer during that year, either by not yet having become a customer (by not having made a first visit), or by the customer having churned during a previous year. For some customers in Definition 1 with visit cycles C0 or C2, prediction is rendered unnecessary, as it is not possible for them to churn. The example customer "Viktor" is such a customer for this definition, and he is thus removed before the prediction. Prediction is also rendered unnecessary for customers with a future reservation, and they are also removed from the data set.

	Tilde	Ida	Kerstin	Pelle	Klara	Viktor
Year 1 (historical data)	C0	–	–	–	C0	–
Year 2 (historical data)	C0	C0	–	C0	C0	–
Year 3 (historical data)	C2	C1	C0	C0	C2	–
Year 4 (historical data)	C2	C1	C0	CHURN	C1	C0
Year 5 (year of prediction)	Predicted CHURN	Predicted CHURN	Predicted CHURN	–	Predicted CHURN	(No need for prediction)
Explanation of classification in Year 5	**Churn** - is expected to visit and is thus at risk of churning. The customer is predicted to not visit, resulting in being labelled as churned.	**Churn** - is expected to visit and is thus at risk of churning. The customer is predicted to not visit, resulting in being labelled as churned.	**Churn** - is predicted to not visit, resulting in being labelled as churned, as they will have two winter seasons of inactivity.	The customer has previously met the criteria for churn and has been removed from the data set. **No prediction is made for this customer.**	**Churn** - went from a C0, to a C2, to finally exhibiting a C1 cycle in the last three years of observation. Because of their C1 visit cycle, they are expected to visit but predicted as absent, hence they churn.	The customer is a C0 customer that made a visit in Year 4, meaning they cannot churn in Year 5 according to the definition. **No prediction is made for this customer.**

Figure 4.1: Visualisation of customer churn using Definition 1

As can be seen in Figure 4.1, Definition 1 allows us to consider the cyclic behaviours of the customers when predicting if the customers churn or not.

4.1.2 Definition 2 – built on inactivity

The second definition of customer churn that we constructed for this specific research case was built entirely on customer inactivity, without any consideration of the customers’ cyclic behaviour. The goal with this definition was to follow the trend in previous studies of predicting churn and the recommended approach explained in the theoretical background. Our second definition of customer churn for this specific research case – from now on referred to as Definition 2 – is defined through the following criteria:

***A customer has churned** if they have not performed an on-site winter activity on any of the company's alpine mountain resorts for two consecutive winter seasons, and if they do not have an upcoming reservation for an on-site winter activity on any of the company's alpine mountain resorts.*

Definition 2 is built on verifying two aspects:

1) Checking if the customer has an upcoming reservation or not. If they have one, they cannot churn. If they do not, they are at risk of churning.
2) Checking if a customer has had any on-site winter related activities during the year of prediction, Year 5. If they have, they cannot churn. If they have not, we checked to see if the customer performed any on-site winter related activities during the previous year instead; Year 4. If they have, they cannot churn. If they had not performed a winter activity for either of these years, the customer would be considered to have churned at the end of Year 5.

Definition 2 is simply dependent on a customer being active or not. For this definition, the decision was made that it should take two seasons to churn from the company, since not interacting with an infrequently purchased product for just one season was deemed a time too short to rightfully classify that customer as churned. In Definition 2, all customers churn or do not churn on the same principles, unlike in Definition 1, where the visit cycles affect when a customer can churn or not.

In the following Figure 4.2, we visualise how five example customers can be predicted as churn according to Definition 2 and three cases where there is no need for prediction (example customers "Ida", "Klara", and "Viktor"). Customers visiting a ski resort during a specific year are marked in turquoise, while a white box indicates a customer did not visit during that year. A dash indicates that the customer was not a customer during that year, either by the customer not yet having become a customer (by not having made a first visit), or by the customer having churned during a previous year. When using Definition 2, prediction is rendered unnecessary for some customers in the data set, as it is not possible for them to churn. This customer group contains all customers who made a visit during Year 4. It is impossible for them to churn, since churn requires two years of inactivity. If they had activity during Year 4, it does not matter if they are absent during Year 5, which is the year we are predicting for, as they can only have one year of absence at the most at this point in time. "Ida", "Klara", and "Viktor" belong to this customer group in Figure 4.2. Similar to Definition 1, customers who have a future reservation are also removed before the prediction, since making a prediction for these customers is unnecessary.

	Tilde	Ida	Kerstin	Pelle	Klara	Viktor
Year 1 (historical data)	■	–	–	–	■	–
Year 2 (historical data)		■	–	■		–
Year 3 (historical data)	■	■	■		■	–
Year 4 (historical data)		■		CHURN	■	■
Year 5 (year of prediction)	Predicted CHURN	(No need for prediction)	Predicted CHURN	–	(No need for prediction)	(No need for prediction)
Explanation of classification in Year 5	**Churn** - is predicted to not visit, resulting in being labelled as churned, as they will have two winter seasons of inactivity.	The customer visited in Year 4, meaning they cannot churn in Year 5 according to the definition. **No prediction is made for this customer.**	**Churn** - is predicted to not visit, resulting in being labelled as churned, as they will have two winter seasons of inactivity.	The customer has previously met the criteria for churn and has been removed from the data set. **No prediction is made for this customer.**	The customer visited in Year 4, meaning they cannot churn in Year 5 according to the definition. **No prediction is made for this customer.**	The customer visited in Year 4, meaning they cannot churn in Year 5 according to the definition. **No prediction is made for this customer.**

Figure 4.2: Visualisation of customer churn using Definition 2

Comparing Figure 4.2 to Figure 4.1 in Chapter *4.1.1 Definition 1 – built on cyclic behaviour*, shows that two fewer predictions of customer churn (here for customers “Ida” and “Klara”) is made when utilising this definition.

4.2 Results of the machine learning models

Below are the results of the machine learning classifiers that were evaluated during this project, as well as the Baseline classifier that the performances are measured against. All plots, figures, and prediction rates are visualisations of the classifiers trained on the *ReducedDataSet* (RDS). Due to confidentiality requirements, we do not include plots, figures, and prediction rates for the classifiers trained on the *CompleteDataSet* (CDS). We do, however, include the performances of both data sets, as visualised in tables.

4.2.1 Overview of the classifiers' performances

Table 4.2 contains an overview of the results of the models constructed in this project, listing the AUC and accuracy of each classifier based on definition and data set, as well as which features were most impactful on the prediction made from the RDS, evaluated using SHAP values. SHAP values were not investigated for the K-nearest neighbour algorithm because of an unmanageable runtime. Hence, no feature reduction was made for this algorithm and no features are listed in the right most column in Table 4.2. Additionally, feature importance was not considered for the Baseline classifier, since this classification is not made based on features. A higher score (approaching 100%) is better than a lower score (approaching 0%) for both the AUC and accuracy. The highest AUC scores and accuracies for the respective data sets – CDS and RDS – and the two definitions – Definition 1 and Definition 2 – are underlined in the table.

Table 4.2: Performance of every classifier and their most important features

Classifier	**Definition**	**Performance CDS**	**Performance RDS**	**The five most important features, in descending order**
Decision tree	Definition 1 of churn	AUC: 84.57% Accuracy: 78.40%	AUC: 83.63% Accuracy: 77.94%	*HadVisit_Year4, TotalReservationSpend, NoOfReservationPurchaseOccasions, HadVisit_Year3, ProductsIncludedInLastReservation_ProductA*
Decision tree	Definition 2 of churn	AUC: 73.94% Accuracy: 70.96%	AUC: 70.44% Accuracy: 65.03%	*HadVisit_Year2, NoOfReservationPurchaseOccasions, ProductsIncludedInLastReservation_ProductA, TotalReservationSpend, ProductsIncludedInLastReservation_ProductB*
Random forest	Definition 1 of churn	AUC: 84.70% Accuracy: 78.28%	AUC: 81.96% Accuracy: 78.28%	*HadVisit_Year4, HadVisit_Year2, TotalReservationSpend, HadVisit_Year3, NoOfReservationPurchaseOccasions*
Random forest	Definition 2 of churn	AUC: 74.39% Accuracy: 78.27%	AUC: 69.52% Accuracy: 70.62%	*TotalReservationSpend, NoOfReservationPurchaseOccasions, HadVisit_Year2, ProductsIncludedInLastReservation_ProductA, HadVisit_Year3*
XGBoost	Definition 1 of churn	AUC: 85.56% Accuracy: 78.67%	AUC: 84.18% Accuracy: 78.34%	*HadVisit_Year4, TotalReservationSpend NoOfReservationPurchaseOccasions HadVisit_Year2, HadVisit_Year3*
XGBoost	Definition 2 of churn	AUC: 74.59% Accuracy: 80.17%	AUC: 71.24% Accuracy: 66.77%	*NoOfReservationPurchaseOccasions, ProductsIncludedInLastReservation_ProductA, TotalReservationSpend, ProductsIncludedInLastReservation_ProductC, ProductsIncludedInLastReservation_ProductD*
Logistic regression	Definition 1 of churn	AUC: 84.78% Accuracy: 78.19%	AUC: 83.65% Accuracy: 78.10%	*HadVisit_Year4, HadVisit_Year2, NoOfReservationPurchaseOccasions, HadVisit_Year3, IsMarketingCommunicaitonOk*

Logistic regression	Definition 2 of churn	AUC: 76.03% Accuracy: 66.83%	AUC: 72.48% Accuracy: 57.81%	*HadVisit_Year2, ProductsIncludedInLastReservation _ProductA, NoOfReservationPurchaseOccasions, IsMarketingCommunicationOk, HadVisit_Year3*
K-nearest neighbour	Definition 1 of churn	AUC: 84.38% Accuracy: 77.95%	AUC: 83.26% Accuracy: 77.92%	–
K-nearest neighbour	Definition 2 of churn	AUC: 72.57% Accuracy: 61.52%	AUC: 69.77% Accuracy: 60.79%	–
Baseline	Definition 1 of churn	AUC: 50% Accuracy: 57.27%	AUC: 50% Accuracy: 57.07%	–
Baseline	Definition 2 of churn	AUC: 50% Accuracy: 82.03%	AUC: 50% Accuracy: 81.79%	–

4.2.2 Baseline classifier

The accuracies of the Baseline classifiers are dependent on the sizes of the majority classes relative to the full data set. To determine the accuracy of the majority classifier, the test data subset was used. For every other classifier explored, the data set was balanced using SMOTE to include 50% churning customers and 50% non-churning customers, but this was not the case for the Baseline classifier.

The accuracy differs slightly between the two data sets, which can be explained by the fact that accuracy is measured on the test data, and not on the full dataset. The customers included in the test set vary slightly between the data sets, making the balance of churners and non-churners – and thus the accuracy – different. For Definition 1, the accuracies – and hence also the sizes of the majority classes (here churn) – are 57.27% for the CDS and 57.07% for the RDS. For Definition 2, the accuracies are 82.03% for the CDS and 81.79% for the RDS.

The accuracies differ significantly between the definitions. This is in part explained by the fact that customers receive different labels depending on the definition. Additionally, some customers who are included in the prediction using one definition are excluded from the prediction using the other. When producing the result, we considered whether the phrasing of the definitions rendered prediction unnecessary for certain types of customers, since these customers would not be able to churn. One such case was identified for both definitions; customers who visited a ski resort during Year 4 without having a visit cycle. For Definition 1, this type of customer is someone who visited during Year 4 and had visit cycle C0 or C2. According to Definition 1, it takes two winter seasons for a customer with visit cycle C0 to churn, meaning a customer who visited during Year 4 cannot churn. Customers with visit cycle C2 that had a visit during Year 4 can also not churn during Year 5, since they were not expected to visit that year. For Definition 2, customers that cannot churn are all customers who visited a ski resort during Year 4, since Definition 2 states that it takes two winter seasons of inactivity to churn. These customers were hence removed, since making a prediction for them was not necessary. The exclusion of these customers had a direct impact on the data sets and the accuracies of the Baseline classifiers. As the number of customers that had to be removed from the data sets varied between the definitions, the data sets used for classifying customers according to Definition 1 contained different amounts of customers than the data set used to classify customers according to Definition 2, resulting in the accuracies of the Baseline classifiers yielding different results for the different definitions.

The accuracy of a majority vote classifier who assigns every customer as churned, as our baseline classifiers do, would normally equal the company’s churn rate, as described by Eq. 1. However, the removal of customers resulted in this not being the case for the Baseline classifier, as the number of churners are overly represented for both definitions due to many non-churning customers being removed. Having to remove customers did not create a lack of data, as the data set that we were initially provided was large enough.

For the AUC scores, the Baseline classifiers have scores of 50% for both data sets and both definitions. This result will be further discussed in the Discussion chapter.

4.2.3 Classifying according to Definition 1

The following visualisations are derived from the RDS. Figures 4.3 - 4.7 show the confusion matrices for the Decision tree (Figure 4.3), Random forest (Figure 4.4), XGBoost (Figure 4.5), Logistic regression (Figure 4.6), and K-nearest neighbour (Figure 4.7) classifiers using Definition 1 to create the target variable. The confusion matrices visualise the predicted class of customers in relation to their actual class. In this case, *churn* is considered to be the positive class and *non-churn* the negative. The darker the hue of each quadrant in the matrix, the more data points fall within that category. The number in each quadrant describes the fraction of customers predicted to the category in relation to the full dataset.

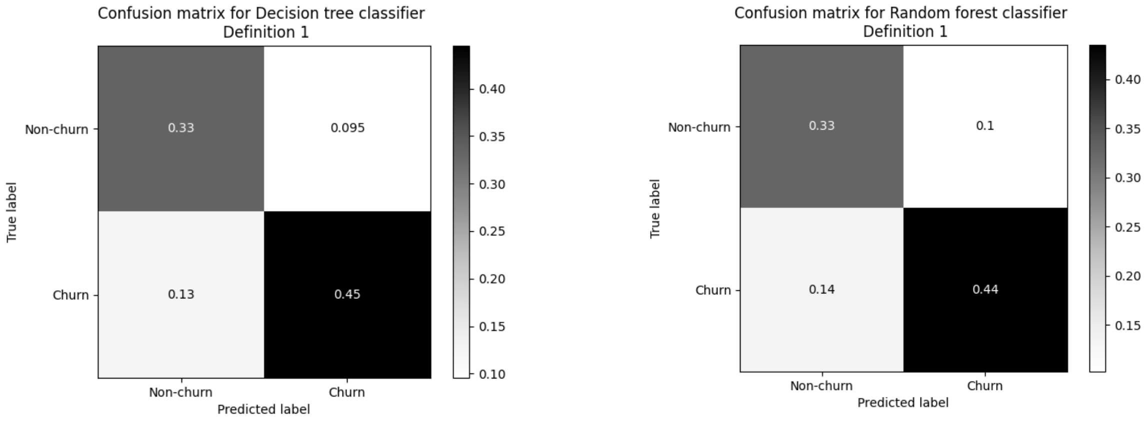


Figure 4.3 and 4.4: Confusion matrix for Decision tree and Random forest using Definition 1.

In Figures 4.3 - 4.7 the quadrants of the left diagonal, going from the upper left corner to the bottom right, are significantly darker than the right diagonal. The bottom right quadrant are the *true positive (TP)* predictions and the top left quadrant are the *true negative (TN)* predictions, which indicates that most customers are correctly classified and the classifiers perform well. The classifiers have a higher number of TP predictions, around 45%, than TN predictions, around 34%, meaning they correctly classify churning customers to a higher degree than customers who do not churn. However, they make TN predictions much more frequently than *false negative* (FN) and *false positive* (FP) predictions. A similarity can also be identified in the false prediction cases, as the bottom left quadrant, containing the FP predictions, is slightly darker at around 13%, than the top right quadrant at around 9.5%, containing the FN predictions. We note that the balance in all quadrants are similar for all classifiers.

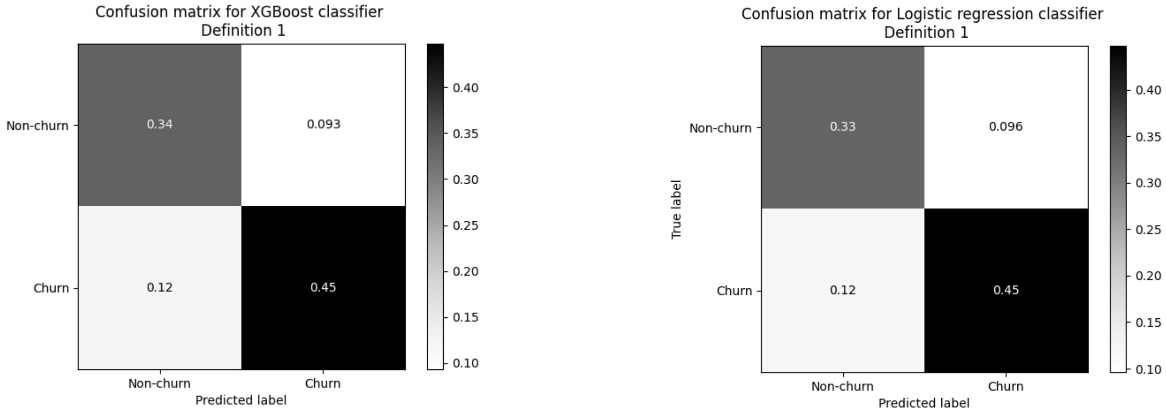


Figure 4.5 and 4.6: Confusion matrix for XGBoost and Logistic Regression using Definition 1.

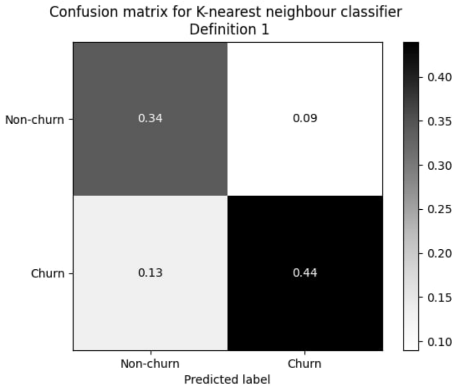


Figure 4.7 Confusion matrix for K-nearest neighbour using Definition 1.

Using the confusion matrices, the prediction rates can be calculated to explain the probability that a certain prediction will be made. The *true positive rate (TPR), false positive rate (FPR), true negative rate (TNR),* and *false negative rate (FNR)* for all classifiers are listed in Table 4.3. The true rates, TPR and TNR, should be high and the false rates, FPR and FNR, should be low to indicate a high performance. The best values in each category are underlined. Discussions and comparisons are made in the fifth chapter of this thesis, *5 Discussion.*

Table 4.3: Prediction rates for classifiers using Definition 1

	Decision tree	**Random forest**	**XGBoost**	**Logistic regression**	**K-nearest neighbour**
TPR	78.02%	76.31%	78.37%	78.45%	77.02%
TNR	77.78%	76.20%	73.30%	77.75%	79.12%
FPR	22.22%	23.80	21.70%	22.25%	20.88%
FNR	21.94%	23.69%	21.63%	21.55%	22.98%

ROC curves can be calculated and visualised using the prediction rates. Figures 4.8 - 4.12 show the ROC curves for the Decision tree (Figure 4.8), Random forest (Figure 4.9), XGBoost (Figure 4.10), Logistic regression (Figure 4.11), and K-nearest neighbour (Figure 4.12) classifiers using Definition 1 as blue lines. The figures also include a black dotted line representing the performance obtained by making a prediction by chance, for example by flipping a coin. It is apparent that the blue line is to the right of the chance curve in all figures, meaning all classifiers outperform the chance classifier.

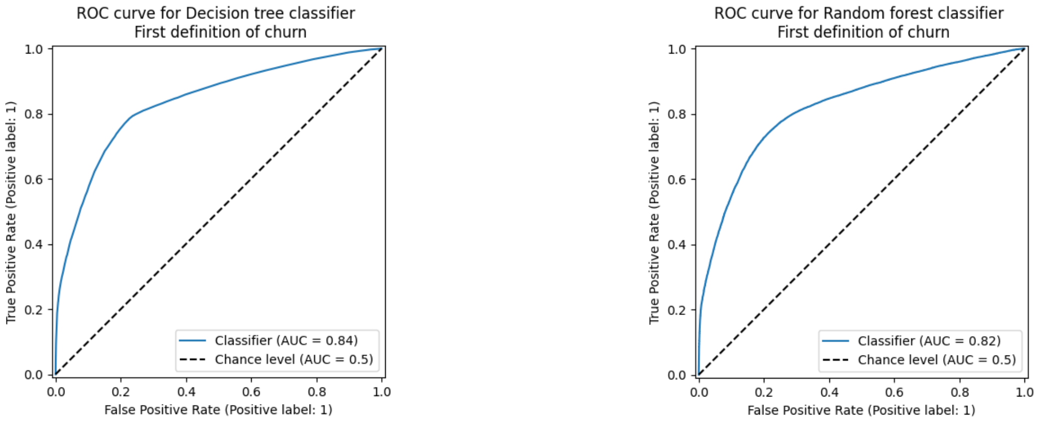


Figure 4.8 and 4.9: ROC curve of Decision tree and Random Forest using Definition 1.

The ROC curve should be as close to the top of the Y-axis as possible for a classifier to have a high performance. It is difficult to determine which classifier performed the best when only observing the ROC curve, as performances are similar. When comparing AUC:s listed in the graphs, it is evident that the best performing classifiers according to the AUC metric, which are rounded numbers, are the Decision tree, XGBoost, and Logistic regression classifiers. To determine which of these perform the best, decimals must be considered by observing the result in Table 4.1. The best classifier using the AUC metric is XGBoost.

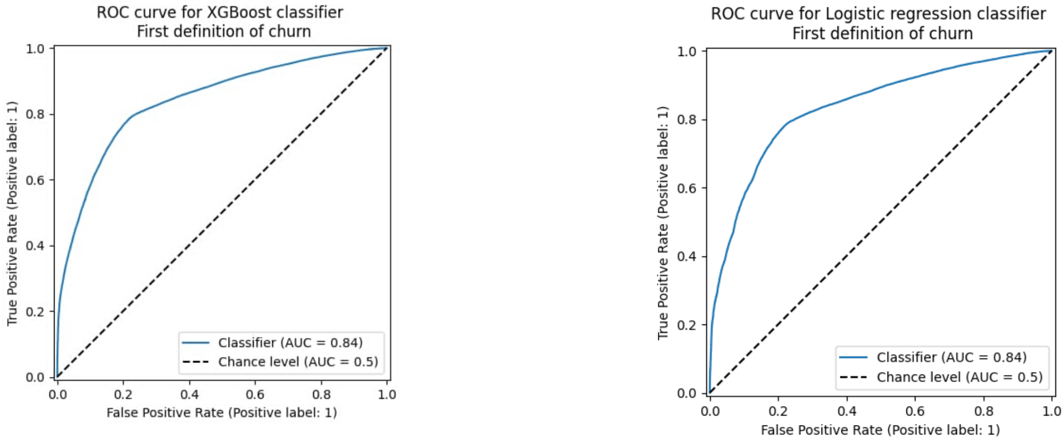


Figure 4.19 and 4.20: ROC curve of XGBoost and Logistic regression using Definition 1.

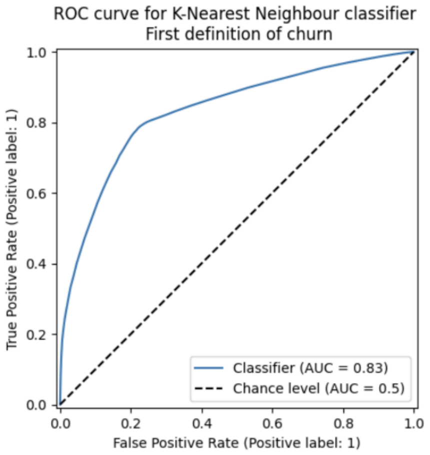


Figure 4.12: ROC curve of K-nearest neighbour using Definition 1.

The exact performances of each classifier are also listed in Table 4.2. Below in Table 4.4 is a ranking of the classifiers based on their AUC scores for the RDS and their accuracies for the RDS, in descending order.

Table 4.4: Ranking of classifiers according to AUC and accuracy using the RDS

Ranking	Classifiers ranked according to AUC (score)	Classifiers ranked according to accuracy (score)
1	XGBoost (84.18%)	XGBoost (78.34%)
2	Logistic regression (83.65%)	Random forest (78.28%)
3	Decision tree (83.63%)	Logistic regression (78.19%)
4	K-nearest neighbour (83.26%)	Decision tree (77.94%)
5	Random forest (81.96%)	K-nearest neighbour (77.92%)
6	Baseline (50.00%)	Baseline (57.07%)

According to Mandrekar (2010), the AUC performance of all classifiers can be described as *excellent*. All classifiers also outperform the Baseline classifier when looking at the AUC performance. Comparing the AUC performance with the typically used metric accuracy, Table 4.4 also shows that XGBoost is the classifier with the highest performance of all the classifiers and that all classifiers have a higher accuracy than the Baseline classifier. The other classifiers have different rankings when looking at AUC compared to accuracy.

Table 4.2 also includes a list of the features that had the highest impact on the prediction made by each classifier. By comparing the impactful features between classifiers, it is evident that there is little variation in which features are most important. The most impactful information was which years a customer had performed a winter activity at a ski resort, described by the *HadVisit* feature, and how often a customer had made a reservation at a ski resort, described by the feature *NoOfReservationPurchaseOccasions.* It was deemed important by all classifiers. Another frequently recurring feature was *TotalReservationSpend,* which describes how much a customer has spent on activities at the ski resorts.

4.2.4 Results from the *CompleteDataSet* for Definition 1

A ranking of the classifiers based on their AUC and their accuracies for the CDS are given in Table 4.5, in descending order.

Table 4.5: Ranking of classifiers according to AUC and accuracy using the CDS.

Ranking	Classifiers ranked according to AUC (score)	Classifiers ranked according to accuracy (score)
1	XGBoost (85.56%)	XGBoost (78.67%)
2	Logistic regression (84.78%)	Decision tree (78.40%)
3	Random forest (84.70%)	Random forest (78.28%)
4	Decision tree (84.57%)	Logistic regression (78.10%)
5	K-nearest neighbour (84.38%)	K-nearest neighbour (77.95%)
6	Baseline (50.00%)	Baseline (57.07%)

All classifiers trained on the CDS and Definition 1 can be described as having *excellent* AUC performances (Mandrekar 2010). This statement does not consider the Baseline classifier. XGBoost is the best performing classifier for this data set as well, both considering AUC and accuracy. The Baseline classifier has the lowest scores considering both AUC and accuracy. The other classifiers have different rankings when looking at AUC compared to accuracy.

4.2.5 Classifying according to Definition 2

Figures 4.13 - 4.17 show the confusion matrices for the Decision tree (Figure 4.13), Random forest (Figure 4.14), XGBoost (Figure 4.15), Logistic regression (Figure 4.16), and K-nearest neighbour (Figure 4.17) classifiers using Definition 2 to create the target variable. Both similarities and differences can be identified between the classifiers' confusion matrices.

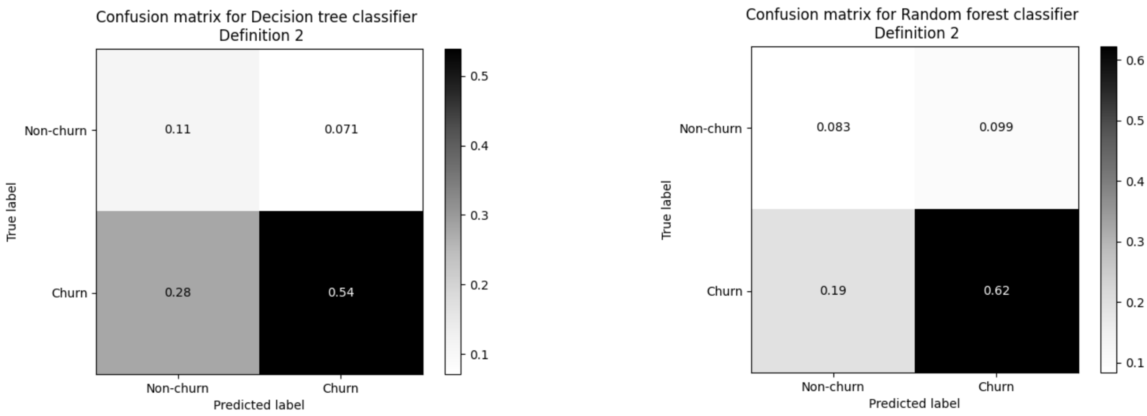


Figure 4.13 and 4.14: Confusion matrix for Decision tree and Random forest using Definition 2.

A trend that can be observed among the classifiers is the fact that the majority prediction class is TP, made evident by the dark hue of the bottom right quadrants. The most common prediction among the classifiers is thus to correctly classify a churner. Another similarity is that the FN prediction, found in the bottom left quadrant, is the second most common prediction class for the classifiers. Differences can be identified in how many customers fall within these categories. Random forest makes the most TP predictions at 62%, followed by XGBoost and Decision tree at around 54% and K-nearest neighbour and Logistic regression at around 45%. 35% of all predictions done by Logistic regression and K-nearest neighbour are FN and the tree based classifiers predict between 19 - 28% FN predictions. Having *true positive* predictions and *false positive* predictions be the most common prediction classes allows us to conclude that classifiers using Definition 2 to construct their target variables are more prone to classify a customer as churned than non-churned.

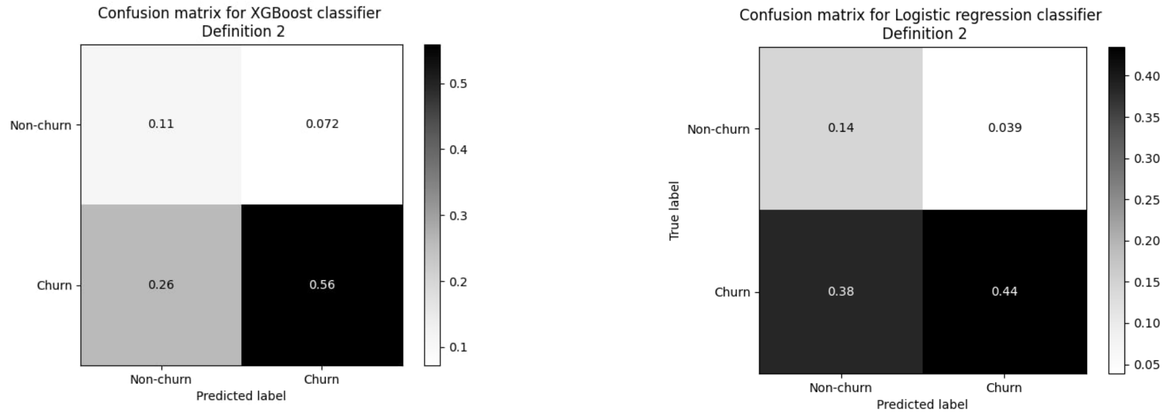


Figure 4.15 and 4.16: Confusion matrix for XGBoost and Logistic regression using Definition 2.

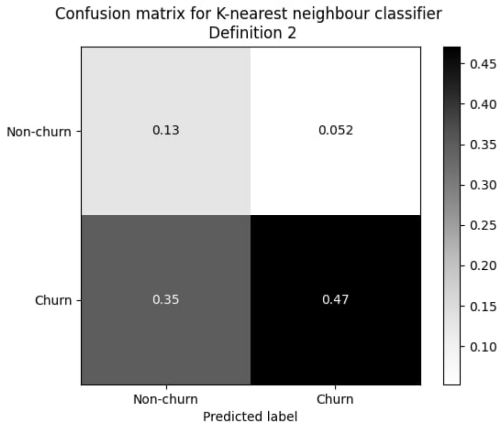


Figure 4.17: Confusion matrix for K-nearest neighbour using Definition 2.

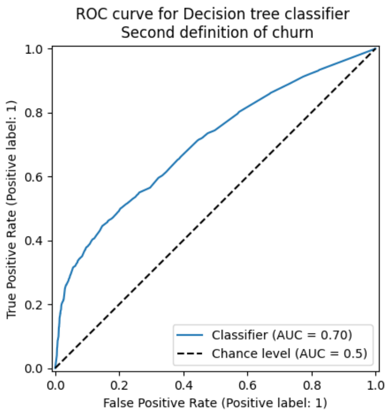
The number of TN and FN predictions do not vary as much between the classifiers as the positive predictions, suggesting that they correctly classify non-churners to a much more similar degree.

Prediction rates can be calculated from the confusion matrices to determine the probability that a customer will be predicted as belonging to a class. The prediction rates are presented in Table 4.6. The true rates, TPR and TNR, should be high and the false rates, FPR and FNR, should be low to indicate high performance. The best values in each category are underlined. Discussions and comparisons are made in the fifth chapter of this thesis, *5 Discussion*.

Table 4.6: Prediction rates for classifiers using Definition 2.

	Decision tree	**Random forest**	**XGBoost**	**Logistic regression**	**K-nearest neighbour**
TPR	65.94%	<u>75.34%</u>	68.27%	53.22%	57.49%
TNR	60.99%	46.90%	60.09%	<u>78.81%</u>	70.94%
FPR	39.01%	53.10%	39.91%	<u>21.19%</u>	29.06%
FNR	34.06%	<u>24.66%</u>	31.73%	46.78%	42.51%

ROC curves can be calculated and visualised using the prediction rates. Figures 4.18 - 4.22 show the ROC curves for the Decision tree (Figure 4.18), Random forest (Figure 4.19), XGBoost (Figure 4.20), Logistic regression (Figure 4.21), and K-nearest neighbour (Figure 4.22) classifiers as blue lines. All figures include a black dotted line representing a classification made by chance, in accordance with the previously mentioned ROC curves. All classifiers outperform the chance classification.



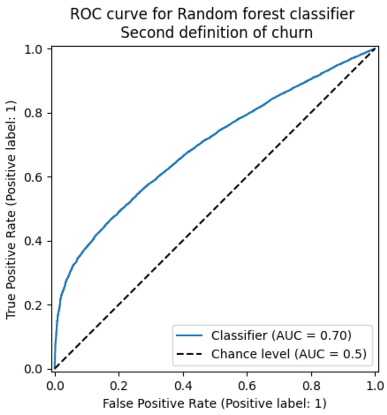
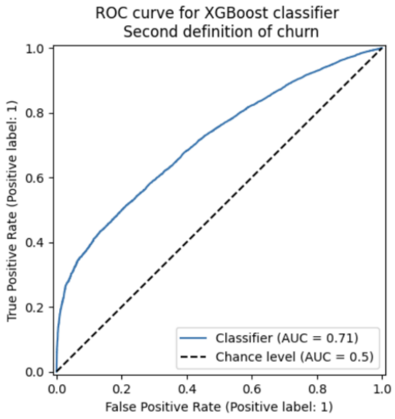


Figure 4.18 and 4.19: ROC curve of Decision tree and Random forest using Definition 2.

When comparing the AUC:s listed in the graphs, it is evident that the best performing classifier is Logistic regression.



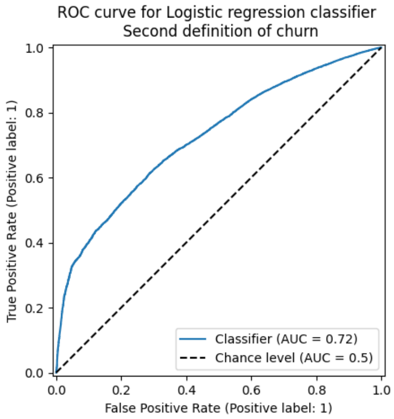


Figure 4.20 and 4.21: ROC curve of XGBoost and Logisitc regression using Definition 2.

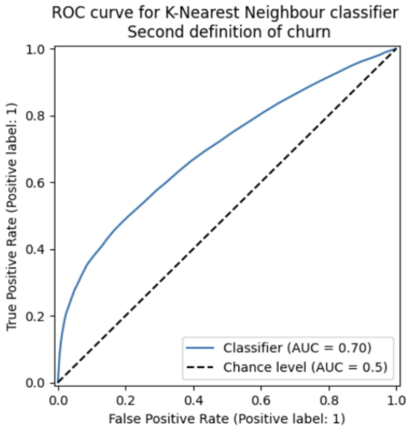


Figure 4.22: ROC curve of K-nearest neighbour using Definition 2.

The exact performances of each classifier are listed in Table 4.2. Below in Table 4.7 is a ranking of the classifiers based on their AUC scores for the RDS and their accuracies for the RDS, in descending order.

Table 4.7: Ranking of classifiers according to AUC and accuracy using the RDS

Ranking	Classifiers ranked according to AUC (score)	Classifiers ranked according to accuracy (score)
1	Logistic regression (72.48%)	Baseline (81.79%)
2	XGBoost (71.24%)	Random forest (70.62%)
3	Decision tree (70.44%)	XGBoost (66.77%)
4	K-nearest neighbour (69.77%)	Decision tree (65.03%)
5	Random forest (69.52%)	K-nearest neighbour (60.79%)
6	Baseline (50.00%)	Logistic regression (57.81%)

According to Mandrekar (2010), the AUC performances of the Logistic regression, XGBoost, and Decision tree classifiers can be described as *acceptable,* but the K-nearest neighbour and Random forest classifiers fall just below the acceptability threshold of 70%. All classifiers have a higher AUC performance than the Baseline classifier.

Comparing the AUC performance with the typically used metric accuracy, Table 4.7 shows that no classifier has a better accuracy than the Baseline classifier. The second best classifier, Random forest, has a significantly lower accuracy than the Baseline classifier. The classifiers have different rankings when looking at AUC compared to accuracy.

Table 4.2 also includes a list of the features that were most impactful on the prediction made by each classifier. By comparing these important features between classifiers, it is evident that there is little variation in which features are most important. How often a customer has made a reservation at a ski resort, described by the feature *NoOfReservationPurchaseOccasions,* and the type of product included in a reservation, described by *ProductsIncludedInLastReservation* were deemed important by all classifiers. Other information that was impactful on most of the predictions was which years a customer had performed a winter activity at a ski resort, described by the *HadVisit* feature, and how much a customer has spent on activities at the ski resorts; *TotalReservationSpend.*

4.2.6 Results from the *CompleteDataSet* for Definition 2

A ranking of the classifiers based on their AUC and their accuracies for the CDS are given in Table 4.8, in descending order.

Table 4.8: Ranking of classifiers according to AUC and accuracy using the CDS

Ranking	Classifiers ranked according to AUC (score)	Classifiers ranked according to accuracy (score)
1	Logistic regression (76.03%)	Baseline (82.03%)
2	XGBoost (74.59%)	XGBoost (80.17%)
3	Random forest (74.39%)	Random forest (78.27%)
4	Decision tree (73.94%)	Decision tree (70.96%)
5	K-nearest neighbour (72.57%)	Logistic regression (66.83%)
6	Baseline (50.00%)	K-nearest neighbour (61.52%)

All classifiers reach acceptable AUC performances (Mandrekar 2010). This statement does not consider the Baseline classifier. Logistic regression is the best performing classifier for this data set when considering AUC and all classifiers outperform the Baseline classifier. On the other hand, the Baseline classifier has the highest accuracy out of all the classifiers, with XGBoost being the second best. The other classifiers have different rankings when looking at AUC compared to accuracy.

5. Discussion

In the following chapter, the results of the project are discussed and compared with the theoretical background presented in the fourth and second chapter of this thesis, respectively. The first subchapter *5.1 The definitions of churn,* discusses the definitions produced during the project. The second subchapter *5.2 The machine learning models,* discusses the predictions made by the classifiers. In the third subchapter *5.3 Recommendations for future research,* we provide some suggestions on how to further expand our research in the future.

5.1 The definitions of churn

When comparing our research case to other studies of non-contractual churn (such as Larivière and Van den Poel (2004), Lee et al. (2019), and Yang et al. (2019)), we have found that other studies use inactivity to pinpoint when customers churn. However, for these other studies, the customer is interacting more often with the product or they have active accounts of some sort. Yang et al. (2019), who studied customer churn in the gaming industry, had determined that their time window would be three days, as they had observed that over 95% of customers did not return to their game after three days of not playing it. With the products offered by the company in our research case being infrequently purchased goods that are dependent on seasonality, the time during which we would have to observe a customer in order to determine if they will return is much longer. Unlike with e.g. a game, the product that the company in our research case offers is also very versatile, allowing the customer to interact with the product through different types of activities, in different combinations and at different locations.

Dealing with infrequently purchased goods makes it difficult to determine when a customer has churned. Another complicating factor is the fact that customers can interact with the products offered by the company in our research case in different ways, as there are many different types of winter activities that can be performed at the ski resorts. Some customers interact with several products and some with very few during a visit. Buckinx and Van den Poel (2005) explain that attempting to predict non-contractual churn is complex, as customers in the non-contractual setting can move back and forth between competitors without having churned, as they can move to a competitor to later return to the company. This applies to our research case, as a customer can decide one year to start visiting a ski resort owned by a different company. With all of the above reasons in mind and in comparison to previous work, we consider our research case to have a high complexity. This complexity in turn likely affects how well we can predict churn for this specific research case.

Creating an accurate definition of non-contractual churn can prove difficult, as there is no clear indicator of when a customer churns (Reichheld & Sasser, 1990). For this reason, we decided to partially rely on industry knowledge by consulting the company studied in the research case, allowing them to provide insight into customers' behaviour in this specific industry. Industry knowledge had a more significant impact on the first definition than the second, as seen in the inclusion of visit cycles. It also impacted how long a customer has to be inactive for them to be considered to have churned. Definition 2 is less dependent on industry knowledge, as it was only used to determine the length of the inactivity period.

Deciding how much time that we allow a customer to not visit without churning also affects the length of the time horizon over which we make our predictions. According to Gold (2020, Chapter 1), a time horizon has to be decided in order to predict non-contractual churn. Berry and Linoff (2004, p.10, 119) describe the time horizons as being fairly short (at most six months), but we predict churn over a full year. We argue that the length of the time horizon in non-contractual churn must depend on the frequency with which a customer can interact with the company's products or services. Winter activities at ski resorts are limited to the winter months, and customers typically visit, at most, once per season. During a time horizon of one year, customers can feasibly visit the ski resort once or twice, depending on the start and end date. With this perspective, a time horizon of one year can be considered to be short.

Inactivity is the only condition of churn in Definition 2, but churn according to Definition 1 is more complex, since it also considers visit cycles. Allowing customers to churn at different points in time depending on their visit cycles partly solves the issue of having to decide on a time window that fits all customers. In reality, customers interact with the product at different frequencies and do not all churn after the same amount of time. This also helps with the problem of deciding when a customer can be pinpointed to have churned, since having different expectations for different customers allows customers to churn on different terms.

The goal when working with customer retention strategies should not be to reduce the number of churners to zero (Blattberg et al. 2002), but instead, the company should focus on targeting the *right* customers (Gold, 2020, Chapter 1). We have considered this aspect by excluding some customers that are not necessary to make a prediction for. Furthermore, we also considered this aspect by investigating assigning each customer a percentage of churn rather than just a binary classification of 0 and 1. A percentage close to 0 would indicate a low churn risk, while a probability close to 1 would indicate a high risk customer. The practical implication of this would be for the company to receive an indication of which customers would be the right ones to target, depending on how they use the information. However, this removal of certain customers who do not churn makes the churn rate of the project – that is provided, as it is the same as the accuracy performance of the baseline classifier – different than the actual churn rate of the company.

Huang et al. (2012) explains that machine learning models are frequently used to combat customer churn. Producing a model that can accurately predict which customer is about to churn enables the company to take targeted action to retain that particular customer. Being able to anticipate churn may lead to an advantage for the company

(García et al., 2017) as reducing churn by 5% can increase the profitability of a company by up to 85% (Reichheld & Sasser, 1990). Our best performing model produced a result that Mandrekar (2010) would describe as *excellent*, meaning they are good at categorising which customer is going to churn. By providing the company with a list of customers that, most likely, will churn, they can ensure that their retention strategies are focused and efficient.

However, it is important to note that the performance of the models is only measured on their ability to predict churn according to the definitions constructed by us. The model is only of value if the definitions prove themselves to be accurate descriptions of the actual customers' behaviours, as being able to predict something that does not depict reality is, in this case, not of value. Ahn et al. (2020) highlights the importance of including a period after churn has been predicted to determine the accuracy of the definition, but the inability to do so in this project may result in a reduction of the value that can be obtained from the result. The practical implication of this is that the company has to evaluate the models over time and perform tests to see if the implementation of the models has been useful or not.

All the decisions we have made when crafting our two definitions of churn have implications on how well our models perform and how closely our models mimic reality. The phrasing of the definitions affects which customers get labelled as potential churn or as having a high risk of churning, which in turn has the practical implication of affecting which customers the company may target with strategies to reduce churn. With all the considerations we have made based on both previous work and expert advice, as well as the delimitations of this research project, we do however believe that at least the first definition we have created mimics the reality of this complex case of churn to an extent that we deem is acceptable. We deem Definition 1 acceptable since it yields AUC performances at an *excellent* level when used to produce a target variable, while at the same time being accepted by experts in the industry working at the company. Knowing for sure that the definition mimics reality can, however, only be achieved by testing our models in practice and evaluating them over time, especially if the company evaluates both definitions.

5.2 The machine learning models

For all of the machine learning models constructed in this research project, using Definition 1 to produce the target variable yielded a higher AUC performance than using Definition 2. This holds true when using both the *CompleteDataSet* (CDS) and the *ReducedDataSet* (RDS). When using Definition 1 to produce the target variable, the AUC performances of all the models could be described as *excellent* according to Mandrekar (2010), regardless of which data set was utilised. For Definition 2, all models utilising the CDS had AUC performances that could be described as *acceptable* (Mandekar, 2010), while some of the models using the RDS had performances below an *acceptable* level. For Definition 1, all models – regardless of the data set – had an AUC performance and an accuracy that was higher than the AUC:s and accuracies of their corresponding base classifier. For Definition 2, all classifiers over both data sets had AUC performances that were higher than their corresponding base classifier. However, when ranking the classifiers using Definition 2 according to their accuracies, all classifiers had lower accuracies than their corresponding baseline classifier.

In this thesis, we have evaluated our classifiers using both AUC and accuracy. Accuracy has been used since it is the most typically used performance metric (Huang & Ling, 2005). AUC was chosen as our main metric since, according to the conclusions reached by Theodoridis and Tsadiras (2021), the dominant evaluation metric that has been used to determine the performance of machine learning models when predicting customer churn has been AUC. Furthermore, when making a formal comparison of AUC and accuracy, performance is better captured by AUC than accuracy, according to Huang and Ling (2005). Based on these arguments, we will determine the overall value of the classifiers based on AUC scores, rather than accuracy. It should, however, be noted that when using the accuracy score, Definition 2 is not a useful definition of churn, since all the classifiers have accuracy scores below the base classifiers. Definition 1, on the other hand, is useful for both data sets when considering accuracy, since all the classifiers have accuracy scores above the base classifiers' accuracy scores.

As mentioned in Chapter *4.2.2 Baseline classifier*, our baseline classifiers have AUC scores of 50% for both data sets and both definitions. As stated by Holzinger (2022, p.105) obtaining an AUC of 0.5, or 50%, is similar to making a classification by chance, which has the implication that the classifier has no information that it can use to distinguish between the two labels. This is the case for our baseline classifiers. Since all our machine learning classifiers have higher AUC performances than their respective baseline classifiers, regardless of data set and definition, all our machine learning classifiers are useful compared to the baseline classifiers.

When further considering AUC, most classifiers achieve *excellent* or *acceptable* performances when using both definitions of churn (Mandekar, 2010). This means that most models have positive results by both Mandekar's (2010) and Nahm's (2022) standards. However, when measuring performance using AUC, we can clearly see that Definition 1 outperforms Definition 2. This also means that our more complex definition of churn, Definition 1, is a significantly more useful definition for predicting churn for this research case than our less complex definition, Definition 2.

Considering that Definition 1 is also useful when evaluating through the accuracy metric, since the classifiers using Definition 1 all outperform their baseline classifiers, further proves that Definition 1 is the superior definition of churn. The high accuracy metrics for the baseline classifiers of Definition 2 were caused by the imbalance in the data set when removing customers that were rendered useless for prediction, as explained in Chapter *4.2.2 Baseline classifier*. This removal also shows that Definition 1 was able to capture varying customer behaviours better than Definition 2, since prediction was not rendered useless for nearly as many customers when predicting churn using Definition 1. This means that the simplicity of Definition 2 caused many customers to not qualify for the prediction. Definition 1 allowed different customers to churn on different conditions, which made prediction possible for many more customers than for Definition 2. This, in addition with Definition 1 having higher AUC performances for all classifiers using both sets compared to Definition 2, shows that considering whether a company's different customers churn on the same conditions might impact both how many customers qualify for the churn prediction and how well the machine learning models perform. Definition 1, the more complex definition of the two, is thus clearly the better definition to use when predicting customer churn in this specific research case.

When it comes to the performances of the specific machine learning algorithms, XGBoost had the highest AUC performance for both of the data sets when using Definition 1 to produce the target variable, while Logistic regression had the highest AUC performance for both of the data sets when using Definition 2. As mentioned in Chapter *3.4.1 Model selection*, XGBoost has been one of the most frequently used algorithms to predict customer churn in previous work. XGBoost being our best performing model using Definition 1 to define churn hence aligns with previous work describing churn predictions, even for this complex non-contractual research case. As has already been mentioned, it also performs at a level that Mandrekar (2010) describes as *excellent*.

Apart from yielding a high AUC performance for Definition 1, XGBoost also produced high prediction rates for its *true positive rate* and *true negative rate* and low prediction rates for its *false positive rate* and *false negative rate*, raising its ROC Curve. However, examining the results of the classifiers at this more detailed level reveals that all of the algorithms had similar prediction rates that were all quite accurate. For example, examining the *true positive rates* of the RDS as shown in *Table 4.3: Prediction rates for classifiers for Definition 1*, reveals that Logistic regression classified the most customers correctly as churn, rather than XGBoost (which was close behind). If classifying customers correctly as being at risk of churning is deemed more important than a model having a generally high AUC performance, then studying the prediction rates might be more useful than relying entirely on the AUC performance when deciding which algorithm is the best one to utilise. However, since all the models are performing at an *excellent* AUC performance level for Definition 1, studying the algorithms at a degree this detailed might not be worth the effort.

When studying which features had the highest SHAP values between the definitions for the RDS, we found patterns in which features appeared to be the most important. The most important features for every model have been listed in *Table 4.2. HadVisit, NoOfReservationPurchaseOccasions,* and *TotalReservationSpend* are all features that have to do with a customer's previous reservation and purchase pattern. Additionally, *ProductsIncludedInLastReservation* was deemed important for all classifiers where the target variable was derived from Definition 2. This feature is also a part of explaining the customer's reservation and purchase pattern. Durun-Cengizci and Caber (2024) explain which factors affect customer churn in the tourism industry, and we can identify similarities with the categories Durun-Cengizci and Caber (2024) discuss and our results. We argue that *HadVisit* falls within the category recency or customer engagement which were both highlighted by Durun-Cengizci and Caber (2024). Both *NoOfReservationPurchaseOccasions* and *ProductsIncludedInLastReservation* fall in the customer engagement or monetary category, and *TotalReservationSpend* belongs to the monetary category. Thus, similarities with the findings of this project and previous studies are identified. This indicates that features relating to a customer's reservation and purchase pattern are worth observing in order to make accurate churn predictions.

Not all categories mentioned by Durun-Cengizci and Caber (2024) are reflected in our results. However, as we only disclose the five most important features, the lack of representation of some categories do not indicate that they are unimportant.

We see an increase in performance for the models using the CDS rather than the RDS. For Definition 1, all classifiers using the CDS had higher AUC performances (84.38% to 85.56%) than the top classifier using the RDS (XGBoost, score: 84.18%). For Definition 2, the same pattern may be observed, where all classifiers using the CDS had higher AUC performances (72.57% to 76.03%) than the top classifier using the RDS (Logistic regression, score: 72.48%). This indicates that making sure to study the *right* features has an impact on the quality of the predictive models. We notice that crafting a churn definition that mimics reality as closely as possible has a bigger impact on the AUC performance than having the perfect combination of features. This, of course, might have to do more with the specific features chosen than it has to do with the amount of features, meaning an *acceptable* churn prediction can be made even without the sharpest features.

5.3 Recommendations for future research

In this thesis, we define and predict churn in a context where it has not been previously studied. There is room for further exploration of the research field explored in this thesis and we have identified areas which would be suitable for further research.

Extending the scope of the case to include the churn determination window (Ahn et al., 2020) would enable a conclusion to be drawn about how well the definition of non-contractual churn in the winter tourism industry reflects reality. Such a study would be a key step into determining the monetary value of the results presented above for a company in the winter tourism industry.

Using data that has been collected during a longer time period than five years would allow for a more in-depth analysis of the customers' behavioural patterns. We visualised the patterns through visit cycles, but the relatively short time span available resulted in us only creating two visit cycles which describe the existence of a pattern; C1 and C2. Having the ability to better differentiate between cycles would introduce a higher level of complexity which could potentially increase performance. Generally, it would have been interesting to investigate an even more complex definition, which might have been able to capture the complex behaviours of customers to an even greater extent.

During the initial phase of the project, we labelled the data provided to us to enable the use of supervised learning. Future studies using unsupervised learning can be performed to determine if unsupervised learning is a better choice for a case such as this, when the definition behind the target variable churn is complex and labels have to be created. There is a possibility that machine learning algorithms that are suitable for unsupervised learning, such as clustering algorithms, would outperform the results obtained through supervised learning in this thesis, as our choice of using supervised learning was not based on the conclusion that unsupervised learning was unsuitable for this research case.

6. Conclusions

The purpose of this thesis, *to define and predict non-contractual customer churn in the winter tourism industry,* has been fulfilled by crafting definitions of customer churn and by developing machine learning models performing at an *excellent* level for our specific research case.

Non-contractual customer churn is challenging to define, as the moment of churn is not intuitively identifiable. When considering both the theoretical background and industry expertise, we were able to craft a definition capturing part of the complex churn behaviours of customers in the winter tourism industry. We also discovered that a definition that only considers inactivity, as the theoretical background suggests, is not enough to produce predictive models with an AUC performance higher than just *acceptable* levels for this research case. Based on these discoveries, we draw the conclusion that *if the context surrounding a case of non-contractual churn is complex, a simple definition using inactivity as the sole criterion for churn is not always sufficient*.

Based on industry knowledge, we identified and considered patterns in how customers have previously interacted with the company's products and services and incorporated these into one of our definitions in the form of *visit cycles.* This incorporation enabled us to capture the complexity of the research case by allowing customers to churn on different conditions, and using this definition produced the best performing machine learning models. This leads us to the conclusion that *a churn criteria for a non-contractual research case should take into consideration that different customers churn on different terms*. We furthermore draw the conclusion that *industry knowledge should be considered when constructing the definition*, in part due to the previous conclusion and in part due to the complexity of non-contractual churn requiring that a definition is tailored after the particular research case it concerns. As the existing literature on predicting non-contractual customer churn is limited, the knowledge required to construct a useful definition might in some cases only be possible to extract from industry actors. To enable this, sufficient time and research efforts have to be allocated to understanding the research case before crafting the churn definition.

Based on the conclusions drawn above, we answer our first research question by providing a definition of customer churn for a company in the winter tourism industry offering non-contractual products, as according to our Definition 1:

A customer who is expected to perform *an on-site winter activity on any of the company's alpine mountain resorts before the end of next year's winter season churns if they do not perform such an activity, as long as they do not have an upcoming reservation for an on-site winter activity on any of the company's alpine mountain resorts.* ***A customer who is not expected to perform*** *an on-site winter activity on any of the company's alpine mountain resorts before the end of next year's winter season churns after two winter seasons of inactivity, as long as they do not have an upcoming reservation for a winter activity on any of the company's alpine mountain resorts.*

Using the definition given above, it is possible to use supervised machine learning to predict non-contractual customer churn in the winter tourism industry. Predictions can be made using supervised learning and binary classification. While the AUC performances of all classifiers constructed in this project – Decision tree, Random forest, XGBoost, Logistic regression, and K-nearest neighbour – can be described as *excellent*, the XGBoost classifier performed the best. We draw the conclusion that, given that XGBoost had the highest AUC performance, *an XGBoost classifier can be used to predict non-contractual customer churn in the winter tourism industry*. This answers our second research question.

Important features to include when constructing the machine learning models relate to what year a customer has made a visit to a ski resort, how many times a customer made a reservation for a winter activity at the ski resorts, and how much money the customer has previously spent on winter activities at the ski resorts. We draw the conclusion that *features that are worth observing in order to make a useful non-contractual churn prediction relate to previous reservation and purchase patterns*.

Furthermore, we would like to mention once more that making a churn prediction is only the first step of fighting churn with machine learning. Evaluation of the usefulness of the predictive models and targeted action based on the information the models provide still has to follow in order to extract any value from the churn prediction. Worthy to note is also that a churn definition for a research case will always be unique for that particular case, unless the research cases are almost identical in the kinds of products they offer.

To conclude this Master's thesis, the final overall conclusion drawn from this research project is:

If the definition is sufficiently complex, non-contractual churn in the winter tourism industry can be predicted using data of previous reservation and purchase patterns to construct an XGBoost classifier that achieves an excellent AUC performance.

References

Articles:

Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019). “Customer churn prediction in telecom using machine learning in big data platform”. Journal of Big Data, Vol. 6, No.1, 1–24.

Ahn, J., Hwang, J., Kim, D., Choi, H., & Kang, S. (2020). “A Survey on Churn Analysis in Various Business Domains”. IEEE Access, Vol.8, 1–1.

Amin, A., Shah, B, Masood Khattak, A., Lopes Moreira, F. J., Ali, G., Rocha, A., Anwar, S (2019), “Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods”, International Journal of Information Management, Vol. 46, 304-319

Amin, A., Adnan, A., Anwar, S. (2023), “An adaptive learning approach for customer churn prediction in the telecommunication industry using evolutionary computation and Naïve Bayes”, Applied Soft Computing, Vol. 137, 110103.

Anagnostou, E., Dimopoulou, P., Sklavos, S., Zouvelou, V., & Zambelis, T. (2021), “Identifying jitter outliers in single fiber electromyography: Comparison of four methods”, Muscle & Nerve, Vol. 63, No. 2, 217–224.

Blattberg, R., Getz, G., Thomas, J., Steinauer, J.M. (2002), “Managing customer retention”, Incentive, 10425195, Apr2002, Vol. 176, Issue 4

Boukerche, A., Zheng, L., Alfandi, O. (2020), “Outlier Detection: Methods, Models, and Classification”, ACM Computing Surveys, Vol. 53, No. 3, 1–37.

Buckinx, W., Van den Poel, D. (2005), “Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting”, European Journal of Operational Research, Vol.164, No. 1, 252-268.

Breiman, L. (2001), “Random Forests”, Machine Learning, Vol. 45, No. 1, 5–32.

Chen, T., Guestrin, C. (2016), “XGBoost: A Scalable Tree Boosting System”, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

Cheng, C.H., Kao, Y.F., Lin, H.P. (2021), “A financial statement fraud model based on synthesized attribute selection and a dataset with missing values and imbalanced classes”, Applied Soft Computing, Vol.108, 107487-.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002), “SMOTE: Synthetic Minority Over-sampling Technique”, The Journal of Artificial Intelligence Research, Vol. 16, 321–357.

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, Vol. 34, No.1, 313–327.

Cox, D. R. (1958), “The Regression Analysis of Binary Sequences”, Journal of the Royal Statistical Society. Series B, Methodological, Vol. 20, No. 2, 215–242.

Dursun-Cengizci, A., Caber, M. (2024), "Using machine learning methods to predict future churners: an analysis of repeat hotel customers", International Journal of Contemporary Hospitality Management, Vol. ahead-of-print No. ahead-of-print.

Fix, E., Hodges, J. L. (1989). “Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties”, International Statistical Review, Vol. 57, No. 3, 238–247.

García, D. L., Nebot, À., Vellido, A. (2017). “Intelligent data analysis approaches to churn as a business problem: a survey”, Knowledge and Information Systems, Vol. 51, No. 3, 719–774.

Geiler, L., Affeldt, S., Nadif, M. (2022), “A survey on machine learning methods for churn prediction”, International Journal of Data Science and Analytics, Vol. 14, No. 3, 217–242.

Han, S. (2018), “A study on a predictive model of customer defection in a hotel reservation website”, MATEC Web of Conferences, Vol. 228.

Holzinger, A. (ed.) (2022) “Machine learning and knowledge extraction : 6th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2022, Vienna, Austria, August 23-26, 2022, proceedings.” Cham, Switzerland: Springer Nature Switzerland AG.

Huang, B., Kechadi, M. T., Buckley, B. (2012), “Customer churn prediction in telecommunications”, Expert Systems with Applications, Vol. 39, No. 1, 1414–1425.

Huang, J., Ling, C. X. (2005). “Using AUC and accuracy in evaluating learning algorithms”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 3, 299–310.

Ho, T. K. (1998), “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, 832–844.

Imani, M., Arabnia, H. R. (2023), “Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis”, Technologies (Basel), Vol. 11, No. 6, 167-.

Larivière, B., Van den Poel, D. (2004), “Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services”, Expert Systems with Applications, Vol. 27, No. 2, 277-285.

Lalwani, P., Mishra, M. K., Chadha, J. S., Sethi, P. (2022), “Customer churn prediction system: a machine learning approach”, Computing, Vol. 104, No. 2, 271–294.

Lee, E., Jang, Y., Yoon, D., Jeon, J., Yang, S., Lee, S., Kim, D., Chen, P., Guitart, A., Bertens, P., Perianez, A., Hadiji, F., Muller, M., Joo, Y., Lee, J., Hwang, I., Kim, K.

(2019), “Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data”, IEEE Transactions on Games, Vol. 11, No. 3, 215-226.

Li, Y., Hou, B., Wu, Y., Zhao, D., Xie, A., Zou, P. (2021), “Giant fight: Customer churn prediction in traditional broadcast industry”, Journal of Business Research, Vol. 131, 630–639.

Liu, Y., Fan, J., Zhang, J., Yin, X., Song, Z. (2023), “Research on telecom customer churn prediction based on ensemble learning”, Journal of Intelligent Information Systems, Vol. 60, No. 3, 759–775.

Lundberg, S., Lee, S.I. (2017), “A unified approach to interpreting model predictions”, In Proceedings of the 31st International Conference on Neural Information Processing Systems, 4765–4774.

Mandrekar, J. N. (2010), “Receiver Operating Characteristic Curve in Diagnostic Test Assessment”, Journal of Thoracic Oncology, Vol. 5, No. 9, 1315–1316.

Nahm, F. S. (2022), “Receiver operating characteristic curve: overview and practical use for clinicians”, Korean journal of anesthesiology, Vol. 75, No. 1, 25–36.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011), “Scikit-learn: Machine Learning in Python” Journal of Machine Learning Research.

Rajendran, S., Devarajan, R., Elangovan, G.. (2023), “Customer Churn Prediction Using Machine Learning Approaches”, 1-6.

Reichheld, F. F., Sasser, J. (1990), “Zero defections: quality comes to services”, Harvard business review, Vol. 68, No. 5, 105–105. Harvard Business School Press.

Taherkhani, L., Daneshvar, A., Amoozad Khalili, H., Sanaei, M.R. (2024), “Intelligent decision support system using nested ensemble approach for customer churn in the hotel industry”, Journal of Business Analytics, Vol. 7 No. 2, 83-93

Tamaddoni Jahromi, A., Sepehri, M. M., Teimourpour, B., Choobdar, S. (2010), “Modeling customer churn in a non-contractual setting: the case of telecommunications service providers”, Journal of Strategic Marketing, Vol. 18, No. 7, 587–598.

Theodoridis, G., & Tsadiras, A. (2022), “Applying machine learning techniques to predict and explain subscriber churn of an online drug information platform”, Neural Computing & Applications, Vol. 34, No. 22, 19501–19514.

Turing, A. M, (1950) Computing Machinery and Intelligence. Mind 49: 433-460.

Wu, X., Li, P., Zhao, M., Liu, Y., González Crespo, R., Herrera-Viedma, E. (2022), “Customer churn prediction for web browsers”, Expert Systems with Applications, Vol. 209, 118177.

Yang, W., Huang, T., Zeng, J., Yang, G., Cai, J., Chen, L., Mishra, S., Liu, Y.E. (2019), “Mining Player In-game Time Spending Regularity for Churn Prediction in Free Online Games”, IEEE Conference on Games (CoG), pp. 1-8

Yizhe Ge, Shan He, Jingyue Xiong, & Brown, D. E. (2017), “Customer churn analysis for a software-as-a-service company”, 2017 Systems and Information Engineering Design Symposium (SIEDS), 106–111.

Books:

Bell, J. (2015), Machine learning: hands-on for developers and technical professionals. 1st Edition. Wiley: Hoboken.

Berry, M. J. A. & Linoff, G. (2004), Data mining techniques for marketing, sales, and customer relationship management. 2nd Edition. Wiley: Hoboken.

Deisenroth, M. P., Faisal, A. A.m Ong, C. S. (2020), Mathematics for Machine Learning, Cambridge University Press: Cambridge.

Funck, E.K., Karlsson, T.S. (2021), Handbok för systematiska litteratur- och dokumentstudier inom samhällsvetenskapen, 1st Edition. Förvaltningshögskolans rapporter nummer 158

Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning, MIT Press: Cambridge

Gold, C. (2020), Fighting churn with data: the science and strategy of customer retention, Manning Publications Co: New York.

Han, J., Kamber, M. (2006) Data mining concepts and techniques, 2nd Edition. Elsevier: Amsterdam.

Hastie, T., Tibshirani, R., Friedman, J. (2009). Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer: New York.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An introduction to statistical learning: with applications in R, Springer: New York.

Kumar, V, Andrew Petersen, J (2012), Statistical Methods in Customer Relationship Management, Wiley: Hoboken

Marsland, S. (2014), Machine learning: an algorithmic perspective, 2nd Edition. Chapman and Hall/CRC, an imprint of Taylor and Francis.

Mellouk, A. (2010). Machine Learning, Y. Zhang Edition. IntechOpen.

McKinney, W. (2012). Python for data analysis, O’Reilly: Sebastopol

Murphy, K. P. (2022). Probabilistic machine learning: an introduction, The MIT Press: Cambridge.

Rokach, Lior., & Maimon, O. Z. (2008). Data mining with decision trees theory and applications, World Scientific: Singapore.

Saunders, M.N.K., Lewis, P., Thornhill, A. (2012), Research Methods for Business Students, 6th Edition. Pearson Education UK: London.

Silaparasetty, N. (2020). Machine learning concepts with Python and the Jupyter Notebook environment: using Tensorflow 2.0, 1st Edition. Apress: New York

Witten, I. H., & Frank, E. (2005). Data mining practical machine learning tools and techniques, 2nd Edition. Morgan Kaufman: Burlington.

Witten, I. H. Frank, E., Hall, M. A., & Pal, C. J. (2017). Data mining : practical machine learning tools and techniques, 4th Edition. Morgan Kaufmann: Burlington

Appendix

The project was executed by two master's students. Although we were both involved in all aspects of the project, a natural division of responsibilities and focus areas crystallised during the project. The difference in focus and main contribution between each student is listed in Table A.1. The workload of everything that is not included in the table was divided evenly between us.

Table A.1: Division of main responsibilities

Name	**Responsibility area**
Liljestam, H	Construction of the XGBoost and K-nearest neighbour classifiers, and pre-processing of data in Jupyter Notebook
Lindell, E	Construction of the Decision tree, Random forest, and Logistic regression classifiers