ERASMUS UNIVERSITY ROTTERDAM

WINKELSTRAAT.NL

Econometrics & Management Science Master Thesis

Business Analytics & Quantatative Marketing

Modelling Customer Lifetime Value in a Continuous, Non-Contractual Time Setting

Author:	*Supervisor:*	*Second assessor:*
J.R. Bernat (415592)	Dr. A.J. Koning	Dr. D. Fok

29 October 2018

Abstract

The need of online retailers to maintain a competitive advantage in the today's booming online retail industry has led to an increased focus on customer relationship management (CRM). The aim of CRM is to increase a company's profits by creating long-term relationships with their profitable customers. However, before this can be accomplished, these profitable customers first need to be identified. The profitability of a customer is often expressed in terms of customer lifetime value (CLV), which is the net present value of all future purchases by a customer. The goal of this research is to compare the predictive power of several different classes of prediction models with respect to predicting CLV. These classes include probability models that are specifically designed to model customer purchase behaviour, duration models that model the general time until a customer's next purchase, and machine learning techniques. This research shows that, for Winkelstraat.nl's database of customer activity, probability models are most suitable for predicting CLV.

Keywords: customer lifetime value, probability models, duration models, machine learning

Contents

1 Introduction

In the Information Age, customers often prefer purchasing online to purchasing at traditional brick-and-mortar shops. This change in customer behaviour has led to a large growth in revenue for online retailers, along with a large increase in the number of online retailers. This has caused the online shopping environment to become a highly competitive business environment in which online retailers need to maintain a competitive advantage. Customer Relationship Management (CRM) has become an important business strategy for maintaining this advantage. CRM assumes that companies can increase their profits by identifying the most profitable customers and allocating disproportionate marketing resources to them to create strong, long-term relationships. The profitability of a customer is often expressed in terms of customer lifetime value (CLV). CLV is the net present value of the sum of all future revenues of a customer, minus all costs associated with that customer. Note that CLV can be negative, as the costs of attracting, selling, and servicing customers can exceed their revenues over time. Therefore, online retailers should not aim to create long-term relationships with all of their customers. Besides retaining the most profitable customers, CLV can also be used to enhance the customer acquisition strategy and to improve the output from customer support.

The computation of CLV requires detailed data on customers' purchase behaviour. Online retailers benefit from the fact that all activity on their website is tracked and stored. Therefore, they have data on each customer's order history, as well as on other activities, such as a customer's return history and click-through behaviour. Furthermore, data on personal characteristics are also often available. These data can be used to create models that capture the general purchase behaviour of customers, which can then be used to predict future cash flows and associated costs of customers, or in other words, predict CLV.

To my knowledge, no studies exist that provide an empirical comparison between the predictive performance of machine learning techniques that are used to predict CLV directly, and more traditional CLV models. Here, direct prediction of CLV by machine learning techniques refers to machine learning techniques that do not model individual

components of CLV separately, but rather directly predict future customer spend. This topic is explained in further detail in the literature review (Section 2). Therefore, the aim of this research is to compare the predictive power of several different classes of prediction models with respect to predicting CLV. Three classes are considered in this research, namely probability models, duration models, and machine learning models. For each class I have chosen a model which I consider to be representative of the corresponding class. The probability models are represented by the Pareto/NBD model, which is specifically designed to model customers' purchase behaviour. The duration models are represented by the Cox proportional hazard model, which models the general time until customers purchase again. Finally, the machine learning techniques are represented by a technique called gradient tree boosting. Although machine learning techniques exist that might be better known than gradient tree boosting, it should represent the field of machine learning well due to its rapidly increasing popularity among data scientists and its high predictive performance. In addition to these models, several extensions to or different implementations of these models are studied. For the probability model, the extended Pareto/NBD model by Abe (2009) is also considered. Furthermore, the Cox proportional hazard model is used both with and without the inclusion of covariates in the model, and the gradient tree boosting model is trained using different loss functions. Note that this research focuses primarily on comparing the predictive power of models and less on the degree of interpretability of the models' results. As machine learning models lack interpretability, they are unable to compete in this area with the other models considered, which that do not lack interpretability. Of course, when one is interested in how the predictions come about, one should take the degree of interpretability into account when deciding what model to use for predicting CLV.

The aforementioned models are applied to the data of Winkelstraat.nl, who have provided me with their complete database of six years' customer activity. Winkelstraat.nl is a Dutch online retailer of luxury designer clothing, bags, and shoes. By collaborating with hundreds of exclusive boutiques from all over the Netherlands and Belgium, it is able to offer over 500 different premium and luxury brands. Its brand awareness has grown rapidly in recent years to approximately 180,000 unique customers as of September 2018.

The report is structured as follows. First, a review of scientific literature on CLV prediction is given. Next, the methodology is discussed, which contains a discussion on or description of the data, the research outline, the data pre-processing, the data stationarity, the estimation of each model applied, and the software that was used to implement the models. Thereafter, the results of the models are presented and discussed. The report concludes with the main findings of the research, along with possible future work.

2 Literature Review

CLV prediction has been extensively researched in the literature, and numerous different models are available. However, before discussing these models, it is important to make a distinction between different kinds of customer-company relationships (Reinartz and Kumar, 2000). These relationships can either be contractual or non-contractual. Contractual relationships imply that there is a legal relationship between the customer and the company (e.g. in case of a subscription or membership). In this setting, the company knows exactly when a customer becomes inactive. Conversely, in a non-contractual setting, there is no legal relationship between the customer and the company, and the company does not observe the time when a customer becomes inactive (e.g. in the case of department store purchases). Furthermore, one can distinguish between discrete- and continuous-time purchases. Discrete-time purchases can only occur at a certain time (e.g. charity fund drives), and continuous-time purchases can occur at any point in time. The customer-company relationship of Winkelstraat.nl is non-contractual and continuous in time, and therefore only CLV models that fall within this context are discussed.

A basic model for computing CLV at the individual level is given by

$$CLV_i = \sum_{t=1}^{T} \frac{(p_{it} - c_{it})\, r_{it}}{(1+d)^t} - AC_i, \tag{1}$$

where i is the customer index, p_{it} is the price paid by customer i at time t, c_{it} is the direct cost of servicing customer i at time t, d is a pre-determined discount rate, r_{it} is the

probability of customer i being active at time t, AC_i is the acquisition cost of customer i, and T is the forecast horizon for estimating CLV. The term $(p_{it} - c_{it})$ is often referred to as the customer margin at time t of customer i, and r_{it} is often referred to the customer retention rate at time t of customer i. The customer acquisition, retention, and margin can be modelled separately and thereafter combined to compute CLV. Several different approaches to model customer acquisition, retention, and margin have been used in the literature, some of which are reviewed in this section.

The difference between CLV at the aggregate and individual level is discussed in the next subsection. This is followed by a review of research on CLV involving the usage of probability models, econometric models, and machine learning techniques. Finally, different measures of performance for CLV models are discussed.

2.1 Aggregate CLV

CLV can be predicted at both the aggregate and individual level. At the aggregate level, CLV is computed for segments of customers. For example, Sohrabi and Khanlari (2007) used K-Mean clustering to segment customers according to their lifetime expressed in terms of recency, frequency, and monetary value (RFM) measures. Recency is a measure of how recently a customer made a purchase, frequency is a measure of how often the customer purchases, and monetary value is a measure of how much the customer spends. Similarly, Shih and Liu (2003), Liu and Shih (2005) and Khajvand *et al.* (2011) used K-Mean clustering based on RFM measures to group customers together. However, they argue that recency, frequency, and monetary value are not equally important measures when it comes to predicting CLV. Therefore, they first applied an analytic hierarchy process (AHP) to determine the relative importance of RFM variables in evaluating CLV and subsequently segmented the customers based on their weighted RFM values. Liu and Shih (2005) conclude that applying AHP proved important in predicting CLV. Clustering customers into different groups helps decision-makers identify market segments more clearly and thus develop more effective strategies. However, it has limited use as a measure for allocating resources across customers because it does not account for customer level variations in CLV and is therefore often used as a surrogate

measure. Because this research is concerned with predicting CLV at the individual level, the remaining part of the literature review only includes CLV models that predict at the individual level.

2.2 Probability models

Well known models to predict CLV are probability models. These models take the past purchase behaviour of the entire customer base into account in order to compute the probability that a customer will still be active in the next period, and to predict the number of purchases a customer will make in the next period. Note that the probability that a customer will be active in the next period corresponds to the the retention rate r_{it} in Equation 1. Furthermore, the predicted number of future purchases can be seen as a combination of customer retention and margin as its computation intrinsically contains the condition of being active. However, the number of future purchases does not equal CLV as one does not know how much a customer will spend per purchase. Therefore, Schmittlein and Peterson (1994) created a submodel that predicts the average future purchase value per customer. Under the assumption that the number of future purchases and the average future purchase value are independent, these values can be multiplied to obtain a prediction of a customer's total future spend. The assumption of Schmittlein and Peterson (1994) that purchase values can be described by a normal distribution was dropped and replaced with a gamma-gamma model by Fader *et al.* (2005), who argued that the distribution of purchase values is too skewed to be characterised by a normal distribution.

The NBD model (Ehrenberg, 1959) is the first probability model for customer base analysis. This model assumes that customers purchase randomly around an individual-specific, time-invariant purchase rate, and that this purchase rate is different per customer. These two assumptions are captured by a Poisson distribution with gamma-distributed purchase rate, which is also known as the negative binomial distribution (NBD). However, the assumption of time-invariant purchase rates usually does not hold. Therefore, Schmittlein *et al.* (1987) introduced the Pareto/NBD model, which extends the previous model by allowing for time-variant purchase rates. The

Pareto/NBD model assumes that customers are first 'alive' (actively making purchases) for an unobserved period of time before they 'die' (become permanently inactive). While a customer is alive their purchase pattern is captured by the NBD model, and the time a customer stays alive is captured by an exponential distribution with a gamma-distributed dropout rate, which is also known as the Pareto distribution. A property of the Pareto/NBD model is that the recency, frequency, and observation length of customers are the only statistics required to make predictions of their future number of purchases. It therefore does not require other information such as the exact time purchases were made. Despite its limited use of purchase information, Fader and Hardie (2009) claim that the model has good predictive performance.

The Pareto/NBD model is modified by, among others, Glady *et al.* (2009), who relaxed the assumption of independence between the number of purchases and the average purchase amount and show that a dependency between these values can be exploited to increase the accuracy of CLV predictions. Other modifications of the Pareto/NBD model are proposed by Fader and Hardie (2007) and Abe (2009), who developed, respectively, a frequentist and a Bayesian method which incorporate time-invariant covariates in the model; this can especially be useful for rich data sets that include various characteristics on customers and their purchases.

2.3 Econometric models

This subsection contains a brief overview of studies that used econometric models to predict CLV. These models are often used to model customer acquisition, retention, and margin separately, and are ultimately combined (see Equation 1) to obtain CLV predictions. Therefore, customer acquisition, retention, and margin are also reviewed separately in this subsection.

2.3.1 Customer acquisition

Customer acquisition refers to the first purchase by new customers. Research in this field often focuses on factors that influence the acquisition of customers. Additionally, they

attempt to link customer acquisition with retention and CLV. However, since this research focuses on predicting CLV for existing customers and not on analysing how certain factors influence customer acquisition, this topic is not reviewed further.

2.3.2 Customer retention

Customer retention refers to the probability of a customer being active at some future point in time t. There are two broad classes of retention models. The first class is called the 'lost for good' class and assumes that customer defection is permanent. The second class is called the 'always a share' class and assumes that customers can switch between vendors. The 'lost for good' retention models usually are duration models. Allenby *et al.* (1999), Lewis (2006), and Venkatesan and Kumar (2004) used an accelerated failure time duration model to model relationship duration. Furthermore, Bolton (1998), Gönül *et al.* (2000), Knott *et al.* (2002), and Levinthal and Fichman (1988) used a proportional hazard model to model customer retention. The 'always a share' retention models usually are migration or Markov models. Markov models estimate the transition probabilities of a customer being in a certain state. Bitran and Mondschein (1996) used a Markov model and define transition states based on RFM measures, and Pfeifer and Carraway (2000) only used recency to define them, as well as an additional state for new or former customers.

Customer retention can also be modelled by using machine learning techniques. For example, Datta *et al.* (2000), Buckinx and Van den Poel (2005), Hung *et al.* (2006), and Koh and Gerry (2002) used neural networks and decision trees, Bae *et al.* (2005) and Song *et al.* (2004) used self-organising maps, and Cheung *et al.* (2003) used support vector machines to model customer retention.

2.3.3 Customer margin

Customer margin refers to the margin generated by a customer at time t. A simple method is used by Reinartz and Kumar (2003) and Gupta *et al.* (2004), who assumed constant margins across time and use the average margin of a customer's past purchases. Gupta

and Lehmann (2005) show that in many cases this assumption is likely to hold. Venkatesan and Kumar (2004) relaxed this assumption and used a simple regression model to capture time-variant customer margin. A more complex method involving Markov chain models was used by Etzion *et al.* (2005). Machine learning techniques can also be applied to model customer margin. For example, neural networks were used by Drew *et al.* (2001), and Bayesian network classifiers by Baesens *et al.* (2004).

2.4 Machine learning techniques

Machine learning techniques are universal approximators and generally have good predictive performance. The main advantage of these techniques is that they are very flexible as they make no assumptions on underlying relationships in the data. However, the disadvantages of these techniques are that they usually require a lot of parameter tuning, may suffer from overfitting, and can be computationally expensive. Machine learning techniques can be used to model either customer acquisition, retention, or margin, as discussed before. However, in this research a different approach to the use of machine learning techniques to predict CLV is considered. Instead of combining customer acquisition, retention, and margin using Equation 1, machine learning is used to directly predict customers' total future spend one year ahead. The choice of a one-year prediction horizon is an implication of the distinctive nature of machine learning techniques, which will be clarified in Section 3.2.1. To predict customer spend one year ahead, the machine learning technique infers a function from labelled training data, where the previous year's aggregated customer order history, along with additional covariates, serve as input into the model, and the following year's customer spend serves as desired output value.

Surprisingly, there are few studies that have considered machine learning techniques for modelling CLV directly. Malthouse and Blattberg (2005) used, among other things, a neural network to predict CLV. In a slightly different context, Chamberlain *et al.* (2017) explain the CLV model for a global online fashion retailer, and show that the use of neural nets on a rich source of data can significantly improve prediction performance. Moro *et al.* (2015) present, in a contractual setting, a CLV data-driven approach using neural

networks to predict possible bank deposit subscriptions. Machine learning techniques can also be used to combine predictions from different models. These predictions can serve as feature vectors in, for example, random forest or neural networks, which may result in better predictions compared to individual models.

2.5 Performance measures

There are multiple ways to measure the performance of CLV models. Glady *et al.* (2009) split the data set into a training set of three years and a test set of two years of data. They predicted CLV using data from the training set and compared their predictions with the actual CLV extracted from the test set by computing the root mean squared error and the mean absolute error. In order to improve robustness to possible outliers in the data set, they discarded the largest 1% of the prediction errors. Additionally, they ranked each individual by sorting them based on predicted CLV and compared them with their true ranking by computing the Spearman's correlation coefficient, which measures the strength of a monotonic relationship between two variables. Furthermore, Venkatesan and Kumar (2004) split the data into a 2.5 year training set and a 1.5 year test set. They ranked the customers from best to worst according to each of their models and then compared the actual sales, costs, and profits from the predicted top 5%, 10%, and 15% of customers. Similarly, Malthouse and Blattberg (2005) split their data into a training and test set of roughly the same size. They ranked the customers from best to worst and looked at whether customers in the predicted top 20% are part of the actual top 20%. They then computed the accuracy, false positive rate, and false negative rate of the predictions.

3 Methodology

This section contains a detailed description of the set-up of this research and the application of the models used in it. First of all, the Winkelstraat.nl data set is described, whereafter a general research outline is given which explains how the data and models are used to obtain CLV predictions. Next, a description is given of how the

data are pre-processed to ensure that they are in the correct format for use in the models, followed by an evaluation of the data stationarity. Afterwards, it is explained how the Pareto/NBD model, the duration model, and the gradient tree boosting model are applied, in that order. The section ends with a brief overview of the software that was used to implement the models.

3.1 Data

The data are obtained from Winkelstraat.nl, an online Dutch retailer that specialises in designer clothing, and consists of two data sets. The first data set, called *products*, consists of all 422,724 products that were purchased between 3 October 2012 and 28 March 2018. The second data set, called *customers*, contains personal information on all 170,556 customers who made purchases between 3 October 2012 and 28 March 2018.

For each product in the *products* data set, the following attributes are available:

- **Product ID:** The unique ID of the product.
- **Order ID:** The unique ID of the order. An order consists of either a single product or a group of products that were purchased together. Hence, multiple products can have the same *Order ID*.
- **Customer ID:** The unique ID of the customer who purchased the product.
- **Purchase Date:** The purchase date of the product.
- **Price:** The price of the product.
- **Returned:** Whether the product was returned.
- **Brand:** The brand of the product. Winkelstraat.nl offers approximately 500 different brands.

The *customers* data set contains information on personal characteristics of all customers and has the following attributes:

- **Customer ID:** The customer's unique ID.
- **Account:** Whether or not the customer created an account on Winkelstraat.nl.

- **Email:** The customer's email address.
- **Date of Birth:** The customer's date of birth.
- **Sex:** The customer's sex.
- **Subscriber Start Date:** The start date of a possible subscription to the monthly newsletter.
- **Subscriber End Date:** The end date of a possible subscription to the monthly newsletter.

Since Winkelstraat.nl was founded in October 2012 and has been growing ever since, the number of orders is not evenly distributed across the years. Figure 1 shows that the first year (3/10/2012 to 2/10/2013) only accounts for a small proportion of the total number of orders, and that each year the total number of orders increased. The first and second year contain 3.1% and 13.8% of the total number of orders, respectively. In addition, in order to increase our understanding of the purchase behaviour of customers, Figure 2 shows the number of customers per total number of orders by customers. The majority of customers, approximately 120,000 (70%), only made a single purchase. For these customers, especially if they purchased recently, it may be difficult to accurately predict their CLV because little is known about their purchase behaviour.

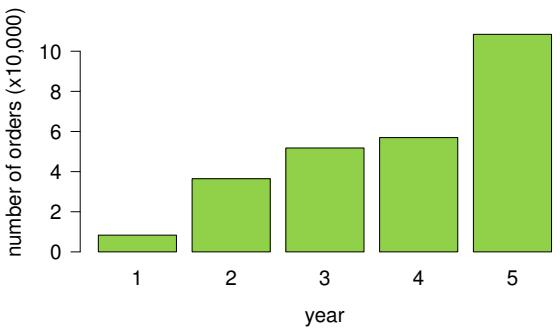

Figure 1: Total number of orders per year for the first five years.

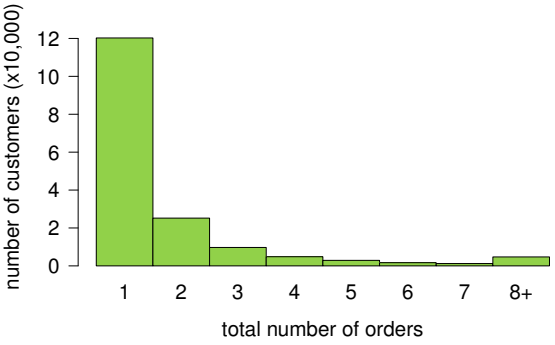

Figure 2: Number of customers per total number of orders by customers.

3.2 Research outline

This subsection first discusses the time horizon of prediction, followed by a description of the performance measures that are used to evaluate the predictive performance of the models.

3.2.1 Time horizon of prediction

The Winkelstraat.nl data set consists of roughly 5.5 years of transaction data. To account for seasonal trends, only data collected from whole years should be used in the models. Therefore, the data from approximately the first six months are discarded from the data set so that it consists of exactly five years of data. Note that a large part of the data is preserved since the first half year of data only contains 0.6% of all orders (see also Figure 1). From now on, I will refer to the period 29/03/2013 to 28/03/2014 as year 1, the period 29/03/2014 to 28/03/2015 as year 2, and so on, until year 5.

Although, theoretically, CLV predictions concern an infinite time horizon, it is common practice to use a finite time horizon because customers usually do not stay customers for their entire lifetime, and furthermore, it simplifies statistical models. To enable the performance of the CLV models to be measured, the data is split into a training and test set. This way, each model can be fitted to the data using the training set, whereafter predictions can be made for a finite future period, which can be compared with the actual values in the test set.

The different natures of the three classes of models have an important implication for the choice of the length of the training and test set. The probability and duration models are flexible methods in the way that the training set length is independent of the prediction horizon. In other words, given a training set of any length, these models can make predictions for a future period of any length. However, in the case of time-series prediction using machine learning techniques, the training set length is dependent on the prediction horizon because these techniques require labelled training data, the labels being the values of interest in the next period. Therefore, the training set is divided into two periods for

the machine learning technique: the data of the first period are used to model the labels (next year's spend) of the second period. Next, the data of the second period are inserted into the trained model to obtain predictions for a future period. Note that the length of the test set should equal the length of this second period, as one trains a model to predict a fixed period ahead. Because of these restrictions, I decided to use the third and fourth year as the training set and the fifth year as the test set, thus predicting the 'next year's CLV' using the previous two years of purchase history. Finally, note that customer acquisition costs are omitted from the CLV prediction. This is because the goal of this research is to compare the performance of different models, and since customer acquisition costs are equal for every method, they do not affect the relative performance of the models.

3.2.2 Performance measures

The models are evaluated by multiple performance measures which can be grouped in three different domains. The first domain is concerned with the prediction of individual CLV. The prediction of individual CLV is relevant to companies who want to target customers with a specific CLV. For example, a company can set up a marketing campaign and target only customers with a CLV above a certain level, as the marketing campaign is only profitable for these customers. The second domain is concerned with predicting customer ordering based on CLV. This may be interesting for companies that want to target their most profitable customers without being interested in their exact level of CLV. Lastly, the third domain is concerned with valuing the total customer base, which can be used for company valuation purposes. Before the performance measures are computed, the true CLV and CLV predictions are discounted to the present by using an annual discount rate of 10%.

The performance of the models with respect to the first domain is measured by the root mean squared error (RMSE), which is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{CLV}_{i,5} - CLV_{i,5})^2}, \tag{2}$$

where N is the number of customers, $\widehat{CLV}_{i,5}$ is the predicted CLV of customer i for year 5, and $CLV_{i,5}$ is the actual CLV of customer i for year 5. Because CLV tends to be right-skewed (the distribution is truncated at zero and usually has a fat right tail as a small number of customers often spend relatively much), the RMSE can easily be inflated by some extreme spenders. Therefore, the mean absolute error (MAE) is also considered, which is a more robust performance measure, and is given by

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\widehat{CLV}_{i,5} - CLV_{i,5}|. \tag{3}$$

An even more robust performance measure would be the median absolute error (MedAE). However, since the majority (80.3%) of the customers who made purchases in the training set do not do so in the test set, predicting a CLV of zero for all customers results in a MedAE of 0. This means that this measure would greatly favour models that underestimate the total customer base, and therefore it is, in this case, inappropriate for comparing models.

The performance of the models with respect to the second domain is measured by evaluating the ordering of the top customers with the highest CLV. The customers are ranked from high to low CLV by both their predicted CLV and true CLV. One then computes the percentage of top 10% of customers according to the predicted CLV who are in the actual top 10% of customers. For future reference, this percentage will be referred to as the ranking percentage (RP). Note that 80.3% of the customers who made purchases in the training set do not do so in the test set and thus have an actual CLV of 0. Therefore, the choice of top percentage customers must be lower than or equal to 19.7%, as one cannot rank customers with equal CLV.

The performance of the models with respect to the third domain is measured by the percentage deviation of the predicted total customer base from the true value of the total customer base. In formula, the customer base percentage deviation (CBPD) becomes

$$CBPD = \frac{(\sum_{i=1}^{N}\widehat{CLV}_{i,5} - \sum_{i=1}^{N}CLV_{i,5})}{\sum_{i=1}^{N}CLV_{i,5}} * 100\%. \tag{4}$$

3.3 Data pre-processing

Approximately 54% of the orders were placed by customers who created an account on Winkelstraat.nl. These customers are identified and are assigned a customer ID. The remaining 46% of the orders were placed by customers who checked out as guests and have no customer ID. Therefore, to identify each customer, their email addresses are used to assign them unique customer IDs. An incidental advantage of this method is that it correctly identifies customers who created an account but still made at least one purchase as a guest. Furthermore, the original customer IDs are taken into account and preserved to account for customers who use multiple email addresses. These email addresses then all correspond to the same customer ID.

Since our goal is to predict CLV using two years of transaction data, all customers who made no purchases in the training set are dropped from the data set. Note that it would be impossible to predict the CLV of these dropped customers as no data are available on them in the training set. Moreover, customers who made purchases in the training set but whose first purchase occurred before the training set period are also dropped from the data set, as the probability model requires a customer's complete order history in order to model their future behaviour. As a result, the number of unique customers is reduced to 76,844 and the number of purchased products to 235,735.

The three different models require the data to be in different forms, and therefore the data are pre-processed differently for each model. However, the following pre-processing is performed for all models:

- A new variable called *Revenue* is added to the *products* data set, which equals the price of the product if the product is not returned, and 0 if returned.
- All products purchased on the same day by a customer are considered as a single order, regardless of their *Order ID*. These products are therefore assigned the same unique *Order ID*.
- A new data set called *orders* is created, which consists of all unique orders in the *products* data set. The variable *Revenue* now equals the sum of the revenues from

all individual products in the order. The variable *Returned* is transformed to equal the percentage of returned products in the order. The variables *Product ID*, *Price*, *Category* and *Brand* are omitted from this data set.

- The variable *Date of Birth* in the *customers* data set is transformed to the age of a customer as of the end date of the training set period, and renamed *Age*.
- A new variable called *Subscriber* is added to the *customers* data set, indicating whether the customer was subscribed to the newsletter at the end of the training set period.
- Since the variables *Sex* and *Age* contain missing values, two new variables called *Sex Missing* and *Age Missing* are created, which indicate whether the sex or age of a customer is missing at the end of the training period, respectively. Furthermore, the missing values in *Sex* are imputed by the mode of available values in *Sex*, and the missing values in *Age* are imputed by the mean of available values in *Age*.

The following data manipulations are performed in order to be able to use the Pareto/NBD model:

- Since the Pareto/NBD model only requires the recency, frequency, and monetary value of all customers, all variables from the *orders* data set are discarded besides *Purchase Date*, *Revenue*, and *Customer ID*.
- The recency, frequency, and monetary value are extracted for each customer based on their orders in the training set. Recency is measured as the day of a customer's last purchase. In this case, a recency of 1 corresponds to customers who made a purchase on the first day of the observation period, and a recency of 730 corresponds to customers who made a purchase on the last day of the observation period. The frequency is the number of times a customer made a purchase in the observation period, and the monetary value is the mean purchase value of a customer's orders disregarding returned items.

In order to be able to use the duration model, the following data manipulations are performed:

- All variables from the *orders* data set are discarded besides *Purchase Date* and *Customer ID*.
- A new variable called *Censor* is added to *orders*, which equals 1 if the corresponding order is the last observed order by a customer in the training set, and 0 otherwise.
- A new variable called *Time* is added to *orders*, which equals the time in days between a customer's current purchase and their next purchase if *Censor* equals 0, and equals the time in days between the current purchase and the end date of the training set period otherwise.

In order to be able to use the gradient tree boosting model, the data are pre-processed in the following way:

- The recency, frequency, and monetary value are extracted for both year 3 and 4 from the *orders* data set for all customers, and are called *Recency3*, *Frequency3*, and *Monetary3* for the third year, and *Recency4*, *Frequency4*, and *Monetary4* for the fourth year. In this case, a recency of 1 corresponds to customers who made a purchase on the first day of the observation period, and a recency of 365 corresponds to customers who made a purchase on the last day of the observation period.
- The variable *Subscriber* in the *customers* data set is split into the variables *Subscriber3* and *Subscriber4*, which denote whether or not a customer is subscribed to the newsletter at the end of the third and fourth year, respectively.
- The variable *Returned* in the *orders* data set is split into the variables *Returned3* and *Returned4*, which equal the percentage of returned items in the third and fourth year for each customer, respectively.
- The variable *Age* in the *customers* data set is split into the variables *Age3* and *Age4*, which equal the age of the customer at the end of year 3 and 4, respectively.
- A new variable called *Favourite Brand* is added to the *customers* data set, which is a categorical variable that denotes the brand that has been purchased most often by a customer in the training set. Since there are over 500 different brands in the data set, the five most popular brands are preserved, and the remaining brands are

gathered under the category 'Other'. This way, 24.9% of the customers are assigned a favourite brand. This variable is then split into six dummy variables. Furthermore, each of the dummy variables is split for the third and fourth year, such that they represent each customer's favourite brand in years 3 and 4 respectively. The top five brands are still based on all orders in both years 3 and 4.

Finally, since the extended Pareto/NBD model by Abe (2009), the Cox proportional hazard model, and the gradient tree boosting model make use of covariates, the *orders* and *customers* data set were combined by taking the natural join, where the variable *Customer ID* served as the key attribute.

3.4 Data stationarity

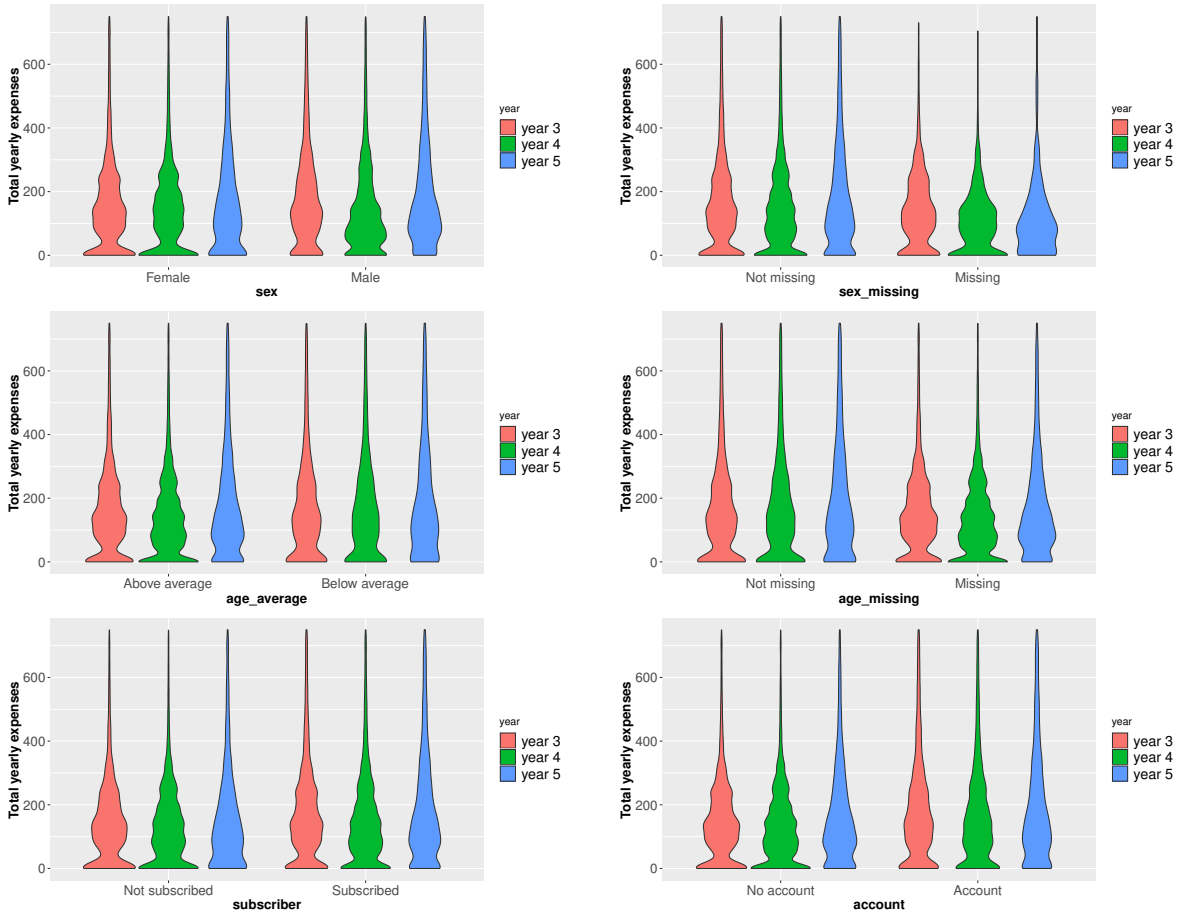

Figure 3: Violin plots of total yearly spends for each variable, where the 'violins' are split between years and between the unique values of the corresponding variable.

In this subsection the stationarity of the data is analysed with respect to years 3, 4, and 5. It is important to check whether the data behave constantly over time, as irregularities over time may indicate that the market has changed, or that the company's operating method has changed. This possible change may make past data inadequate for prediction purposes. As shown earlier in Figure 1, Winkelstraat.nl has been growing rapidly in recent years, meaning that there might be inconsistencies in the data across the years.

The stationarity of the data is analysed by looking at so-called violin plots. Violin plots are similar to box plots, except that they show the probability density of the data. Here, a kernel density estimator is used to estimate the probability density, which uses a Gaussian kernel and a bandwidth value according to the rule of thumb by Silverman (1986). Since the data contain outliers, a robust version of Silverman's rule of thumb is used, which is given by

$$\hat{h} = 0.9 \ \min\left(\hat{\sigma}, \frac{R}{1.34}\right) n^{-\frac{1}{5}}, \tag{5}$$

where $\hat{h}$ is the bandwidth, $\hat{\sigma}$ is the sample standard deviation, R is the sample interquartile range, and n is the sample size.

Figure 3 shows violin plots of total yearly spends for each variable for years 3, 4, and 5, whereby we distinguish between each unique value of a variable. For example, the leftmost violin in the upper left plot shows the density of total yearly spends in year 3 for females. One can see that relatively many of these customers spent nothing, indicating that these customers returned all their purchased products. Furthermore, one can see that there are relatively few customers with a total spend of roughly €20, most customers spent between €30 and €300, and a few customers spent more than €400. Note that the violins are cut off at a total yearly spend of €750 as otherwise the plot will become too small due to the existence of several extremely high-spending customers. Also, note that the variable *Age*, since it is a numerical variable, is split into two groups, indicating whether the age is below or above average.

To detect possible non-stationary data patterns, the violins across the three years are compared to each other for all twelve groups (there are two different values for each of the six variables). Since it is tedious to discuss the stationarity of the data with respect to each group individually, only several general patterns will be discussed. One can see that, for

all twelve groups, the probability density of the total yearly spends is quite similar across time. However, the number of customers with no spends is larger in years 3 and 4 than in year 5. This change is presumably caused by the introduction of a customer blacklist at the beginning of the fifth year that deprives customers who return relatively many products of certain payment options to discourage them from purchasing. Furthermore, the 'neck' of the violin generally seems to be larger in the fifth year than in the other two years, albeit with only a slight difference. Nevertheless, these differences are small, and there exist no major aberrations in probability densities across the years. Therefore, I believe that the data are adequate for predicting customer behaviour in the fifth year if one uses the previous two years of transaction data.

3.5 Pareto/NBD model estimation

To obtain predictions of CLV for each customer, both the Pareto/NBD model and the gamma-gamma submodel are applied to the transaction data. A detailed description of these two models is given in Appendix A. The Pareto/NBD model predicts the future number of purchases for each customer, and the gamma-gamma submodel predicts the future average spend per purchase for each customer. Under the assumption that the expected number of purchases and the expected average spend per purchase are independent, they can be multiplied to obtain CLV predictions.

In addition to the standard Pareto/NBD model, the extended Pareto/NBD model by Abe (2009) is applied to the transaction data. This extension replaces the analytical part of the Pareto/NBD model with a hierarchical Bayes framework and uses MCMC simulation to obtain parameter estimates. Furthermore, it relaxes the independence assumption of the purchase rate and death rate of the standard Pareto/NBD model and allows for the incorporation of time-invariant covariates. Similarly to the standard Pareto/NBD model, it can be combined with the gamma-gamma submodel to obtain predictions of CLV. A detailed description of the extended Pareto/NBD model by Abe (2009) is given in Appendix A.

The extended Pareto/NBD by Abe (2009) is applied both with and without the

inclusion of time-invariant covariates. In both cases, a single MCMC chain is constructed to generate draws from the posterior distribution of the model parameters. Each chain uses 50,000 steps, whereby the first 200,000 steps are used as burn-in steps and are discarded. Furthermore, in order to reduce the autocorrelation between the draws, a thinning value of 100 is used, meaning that only the draws of every 100^{th} step are returned. Whether the Markov chains are converged is assessed by looking at trace plots of the parameter draws.

The extended Pareto/NBD model that incorporates covariates into the model uses the following covariates: *Sex*, *Sex Missing*, *Age*, *Age Missing*, *Subscriber*, and *Account*. Note that the age of customers is technically not time-invariant. However, as the training set consists of only two years of transaction data, the customers' age will only change slightly over these two years. Therefore, the influence of age difference between customers on purchase behaviour is practically negligible, whereas the difference in age between customers is larger and might contain predictive power. Furthermore, the variable *Subscriber* is considered to be time-invariant, as most customers stay subscribed or unsubscribed during the complete training period. Finally, the variable *Account* is also considered as time-invariant, as the large majority of customers either set up an account during their first purchase or do not set up one at all. Moreover, the majority of customers only make a single purchase, meaning that *Account* is naturally time-invariant for these customers.

To summarise what data are used by the probability models, the recency, frequency, and monetary value from year 3 and 4 of all customers are used to train the models and to obtain CLV predictions. In addition, the extended Pareto/NBD model uses the following covariates: *Sex*, *Sex Missing*, *Age*, *Age Missing*, *Subscriber*, and *Account*.

3.6 Duration model estimation

Duration models represent a class of analytical methods that are appropriate for modelling data where the focus lies on the occurrence of a certain event. A characterising feature of these models is that they can deal with censored observations, which are observations of

an event of interest that has not yet occurred at the time the data are analysed. Given the waiting time until the occurrence of an event for each observation, and whether each observation is censored or not, duration models measure the general likelihood of an event occurring. A detailed description of duration models is given in Appendix B. In this research the Cox proportional hazard model is applied, both with and without the inclusion of covariates in the model. Note that in the latter case, the semi-parametric estimation of the Cox proportional hazard model reduces to ordinary non-parametric estimation (Rodrıguez, 2005).

In this research we are interested in the time when a customer will make their next purchase. Therefore, the input into the duration model is the time between a customer's previous purchase and their next purchase, for each purchase by a customer and for all customers. In addition, it is indicated whether each observation is censored or not. Note that each purchase by a customer is treated as an independent data instance in the model. Therefore, the index i in Appendix B refers to a single purchase by a customer and not to customers themselves. Furthermore, note that the time between a customer's very last observed purchase and their possible next purchase is censored, as we have not yet observed the customer's next purchase. For these observations, the waiting time until the occurrence of the next purchase is set to the time between the last purchase and the end of the training set period. All other observations are not censored. In addition, the following time-invariant covariates are included in the Cox proportional hazard model: *Sex*, *Sex Missing*, *Age*, *Age Missing*, *Subscriber*, and *Account*.

Once the duration model is trained, the probability that a customer will purchase at least once in the next year (year 5), or, in other words, the probability that a customer will still be 'alive' in the next year, is computed for each customer. CLV is then predicted by multiplying this probability by the customer's total spend in the previous period (year 4), assuming that, given that a customer is alive in the next period, his total yearly spend stays the same (see also Donkers *et al.* (2007)).

To summarise what data are used by the duration model, the time between each purchase by a customer, for all purchases by all customers in years 3 and 4 are used to train the model. In addition, the Cox proportional hazard model uses the following covariates: *Sex*,

Sex Missing, *Age*, *Age Missing*, *Subscriber*, and *Account*.

3.7 Gradient tree boosting estimation

Gradient tree boosting is a machine learning technique that uses an ensemble of 'weak' decision trees to obtain a 'strong' predictor, and can be used in both classification and regression settings. A detailed description of gradient tree boosting can be found in Appendix C. A characteristic of machine learning techniques is that they are able to learn without being explicitly programmed to do so, and they can therefore be applied to almost any prediction setting. Because of this, and the fact that machine learning techniques often have a high predictive performance, they have become popular among data scientists. However, the disadvantage of these models is that they lack interpretability, as the complex architecture of these models masks the effect of covariates on the variable of interest.

In the remainder of this subsection, a description of the model's set-up is given, followed by a description of the hyperparameter tuning process.

3.7.1 Model set-up

To obtain forecasts of CLV, the model is trained to predict customer spend one year ahead using the previous year's transaction data. First of all, the variables *Recency3*, *Frequency3*, *Monetary3*, *Returned3*, *Subscriber3*, *Age3*, *Sex*, *Account*, and the *Favourite Brand* dummy variables measured over year 3 are used to predict the total customer spend in the next year. After the model is trained, CLV predictions are obtained by inputting the same variables as those on which the model was trained, the difference being that the time variant variables (all but *Sex* and *Account*) are measured over year 4. Note that since there is no data available in year 3 on customers who made their first purchase in year 4, the model is only trained on customers who have made at least one purchase in year 3.

For all customers who made no purchases in year 4 (i.e. only purchased in year 3), their

recency4, *frequency4*, *monetary4*, and *returned4* are not available. Therefore, since the goal of this research is to make CLV predictions for all customers, I assume that customers who made no purchases in year 4 will have a low CLV, so that the values of these four variables can be set to values that lead to a low CLV. Therefore, their *recency4*, *frequency4*, and *monetary4* are set equal to the minimum value encountered in *recency3*, *frequency3*, and *monetary3*, which is 1 for *recency3* and *frequency3* and 0 for *monetary3*. Note that a low recency value corresponds to customers with no recent purchases. Furthermore, *returned4* is set equal to the maximum value encountered in *returned3*, which is 1. In addition, *Subscriber4* and all *Favourite Brand* dummy variables are set to 0.

To summarise what data are used by the model, the gradient boosting model is trained on the following variables for all customers who made at least a single purchase in year 3: *Recency3*, *Frequency3*, *Monetary3*, *Returned3*, *Subscriber3*, *Age3*, *Sex*, *Account*, and all *Favourite Brand* dummy variables. To obtain CLV predictions, the same variables as those on which the model was trained are used as input in the model, the difference being that the time-variant variables are measured over year 4.

Since the model's performance is measured, with respect to the first domain, by both the RMSE and MAE, the gradient tree boosting model is trained twice, once with the RMSE as a loss function and once with the MAE as a loss function. However, since the gradient boosting algorithm requires a derivation of the Hessian of the loss function, the MAE loss function cannot be used as it does not have a continuous second order derivative. Therefore, the Fair loss function is used instead of the MAE loss function, which approximates the MAE loss function and has a continuous second order derivative. The Fair loss function is given by

$$c^2\left(\frac{|x|}{c} - ln\left(\frac{|x|}{c} + 1\right)\right), \tag{6}$$

its first order derivative is given by

$$\frac{cx}{|x| + c}, \tag{7}$$

and its second order derivative is given by

$$\frac{c^2}{(|x| + c)^2}. \tag{8}$$

Here, x is the error and equals the prediction minus the actual value, and c is a control parameter which controls the smoothness of the function. Figure 4 shows a plot of the MAE vs the Fair loss function on both the interval $[-2, 2]$ and $[-50, 50]$. It shows that the Fair loss function is smooth around zero and that it approximates the MAE loss function well for larger values. In this research, the control parameter c is set to 1.

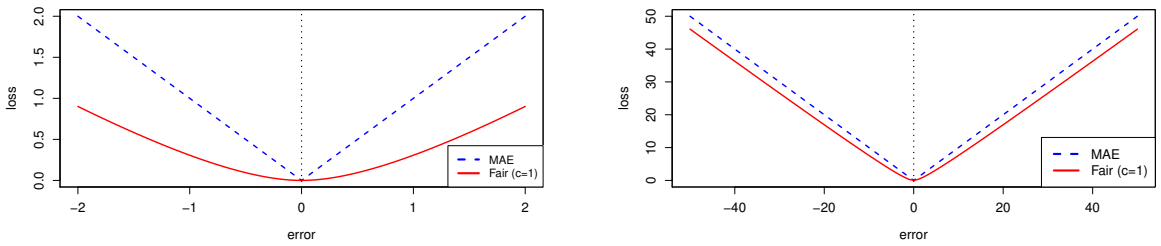

Figure 4: MAE and Fair loss function on the interval $[-2, 2]$ (left) and $[-50, 50]$ (right).

3.7.2 Hyperparameter tuning

Table 1: List of gradient tree boosting hyperparameters, along with a short description of each hyperparameter and their initialisation value.

Hyperparameter	**Description**	**Initialisation**
max_depth	Maximum depth[1] of a tree.	-
min_child_weight	The minimum number of instances required to be in nodes.	-
γ	Minimum loss reduction required to make a further partition on a leaf node of the tree.	0
subsample	Subsample ratio of the training instances.	0.7
colsample	Subsample ratio of columns when constructing trees.	0.7
α	L1 regularization term on weights[2].	0
λ	L2 regularization term on weights[2].	0
β	The learning rate.	0.1
rounds	The number of trees / boosting rounds.	-

Gradient tree boosting requires one to set values for the model's hyperparameters before it can be applied. These hyperparameters affect the performance of the model and should therefore be chosen carefully. To find their optimal values, 5-fold cross-validation is applied

[1] The depth of a decision tree is the length of the longest path from a root to a leaf.

[2] The software package *XGBoost* Chen *et al.* (2018) uses its own regularised model formulation. Each regression tree contains a continuous score (weight) on each of its leaves. The loss function can then be regularised to control the size of these scores. See Section 2.1 in Chen *et al.* (2018) for details.

to the training set. This way, the performance of the model can be investigated using different sets of hyperparameter values, whereby the best set of hyperparameter values corresponds to the values that lead to the lowest cross-validated training loss. Note that the performance of the model that minimises the Fair loss function will be evaluated by the MAE. A list of hyperparameters that need to be tuned, along with a short description of each hyperparameter, is given in Table 1. The first five hyperparameters in the table are tree-specific parameters, α and λ are regularisation parameters, and β and *rounds* are learning task parameters.

Since it is computationally infeasible to try out all parameter value combinations, some parameters are tuned first while the other parameters are kept fixed. The parameters are tuned in order from parameters that have the greatest impact on the model's performance to parameters that have the least impact on the model's performance. Therefore, the tree-specific parameters are tuned first, whereafter the regularisation parameters, and finally, the learning rate are tuned. For each trained model, the optimal number of boosting rounds should be re-tuned, as its optimal value is highly dependent on other parameters. Since gradient tree boosting builds its trees sequentially, instead of fixing the number of rounds at the beginning, the model's performance can be investigated at each round. Then, if its performance has not improved for ten rounds, the training of the model is stopped and the best number of rounds is used. A detailed road map of how the parameters are tuned is given below.

Step 1: **Initialisation**: The hyperparameters are initialised as shown in Table 1. The hyperparameters *max_depth* and *min_child_weight* are not initialised as they are the first parameters to be tuned (see next step). The parameter *rounds* is not initialised as this parameter is re-tuned for each model as described above. The parameter γ is set to 0 and will be tuned in Step 3. The parameters *subsample* and *colsample* are set to 0.7, as typical values for these parameters range between 0.5 and 0.9. The parameters α and λ are set to 0 as we do not want to control for overfitting yet. Finally, the parameter β is set to a relatively high value, as typical values range between 0.001 and 0.3. A value of 0.1 is a trade-off between computation speed and performance, as a lower learning rate makes the model more robust to overfitting, but increases the computation time significantly.

Step 2: **Tune *max_depth* and *min_child_weight***: The parameters *max_depth* and *min_child_weight* are tuned first as they have the greatest impact on the model's performance. It is important to tune them together in order to find a good trade-off between model bias and variance. A larger value of *max_depth* allows the model to capture more complex relationships, but since splits become less relevant and may be caused by noise, the model may overfit. A smaller *min_child_weight* allows the model to create child nodes that correspond to fewer instances, thus increasing the model's complexity and making the model prone to overfitting. To find the optimal values of *max_depth* and *min_child_weight*, a grid search is performed, in which *max_depth* is varied from 3 to 10 with steps of 1 and *min_child_weight* is varied between 1, 5, 10, 15, ..., 45, with the result that the grid search contains 80 different parameter combinations. After the grid search is completed, both parameters are fixed at their optimal value.

Step 3: **Tune γ**: Smaller values of γ will make the model more complex, and as a consequence, more likely to overfit. Typical values of γ range between 0 and 0.5, and therefore its value is varied from 0 to 0.5 with steps of 0.05. Afterwards, γ is fixed to its optimal value.

Step 4: **Tune *subsample* and *colsample***: The parameters *subsample* and *colsample* control the sampling of the data set for each tree by random selection, where *subsample* controls the rows of the data set on which a tree is built, and *colsample* controls the columns that are considered in each split. By selecting only a subset of the complete data set, each tree is built on slightly different data, making the model less prone to overfitting to a single data instance. Typical values of *subsample* and *colsample* range between 0.5 and 0.9, and therefore a grid search is performed in which the value of *subsample* is varied from 0.5 to 1 with steps of 0.05 and the value of *colsample* is varied from 0.5 to 1 with steps of 0.10. Note that we use larger steps for *colsample* as the model only uses sixteen variables, and therefore too small steps may not increase the number of variables that are considered in splits. After the grid search is completed, both parameters are fixed at their optimal value.

Step 5: **Tune α and λ**: The parameters α and λ add an absolute and a quadratic

penalty term for leaf weights to the loss function respectively in order to avoid overfitting. Although the parameter γ already provides a substantial way of controlling overfitting, it cannot harm to try to reduce overfitting further. Note that a value of 0 for both α and λ means that no regularisation is applied to the model. Furthermore, note that a positive value for both α and λ corresponds to elastic net regularisation. The optimal values of α and λ are found by performing a grid search in which both parameter values vary between 0, 0.001, 0.01, 0.1, 1, and 10. After the grid search is completed, both parameters are fixed at their optimal value.

Step 6: **Tune $\boldsymbol{\beta}$**: As mentioned above, a lower learning rate makes the model more robust to overfitting at the cost of increased computation time. A lower learning rate should therefore increase the model's performance, and for this reason, the learning rate is lowered and varied between 0.001, 0.005, 0.01, 0.05, and 0.1. Finally, β is fixed at its optimal value.

After the optimal parameter values are found, the model is trained once again with the whole training set (i.e. without using cross-validation). The value of the parameter *rounds* is set equal to the number of boosting rounds that led to the lowest cross-validated loss in Step 6. Once the model is trained, the data of year 4 are inserted into the model in order to obtain CLV predictions.

Since the gradient tree boosting algorithm is a randomised algorithm, a more accurate approach to tuning the model's hyperparameters would be to run each model multiple times and report the average RMSE or MAE. This reduces the likelihood of a hyperparameter value being found as optimal due to more favourable random results. However, this approach is not followed since this is computationally expensive.

3.8 Software

The programming language R (R Core Team, 2018) was used for all computations. The packages *BTYD* (Dziurzynski *et al.*, 2014) and *BTYDplus* (Platzer, 2016) were used to implement the Pareto/NBD model and its extension by Abe (2009), respectively.

Furthermore, the package *survival* (Therneau, 2015) was used to implement the duration models. Finally, the package *XGBoost* (Chen *et al.*, 2018) was used to implement the gradient tree boosting model. *XGBoost*, as opposed to other implementations of gradient tree boosting, uses a more regularised model formalisation to control overfitting, which enhances its performance.

4 Results

This section presents and discusses the results of the three models applied to the Winkelstraat.nl data set. First of all, the in- and out-of sample fit to the data of both the standard Pareto/NBD model and its extension are analysed. In addition, the independence assumption of frequency and average purchase value is assessed. Next, the duration model's in-sample fit to the training data is examined, and thereafter the hyperparameter tuning process of the gradient tree boosting model is presented. Finally, the performance of all models with respect to the test set is presented and discussed.

4.1 Pareto/NBD model

The in-sample fit of the Pareto/NBD model and its extension is analysed by comparing their predicted number of customers who made a certain number of repeat purchases in the training set with the actual number, which is shown in Figures 5, 6 and 7 for the standard Pareto/NBD model, its extension by Abe (2009), and its extension by Abe

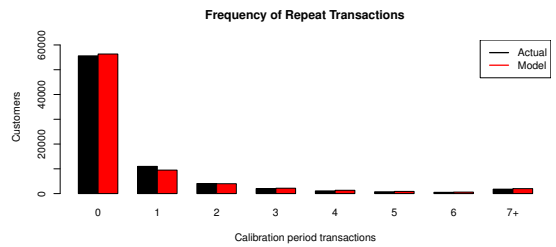

Figure 5: Frequency of repeat purchases according to the standard Pareto/NBD model.

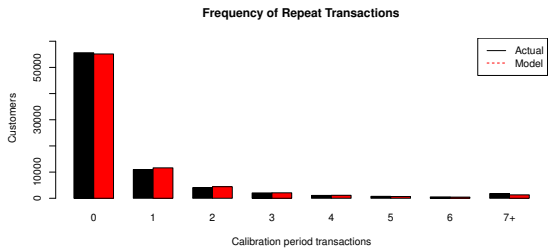

Figure 6: Frequency of repeat purchases according to the extended Pareto/NBD model by Abe (2009) without covariates.

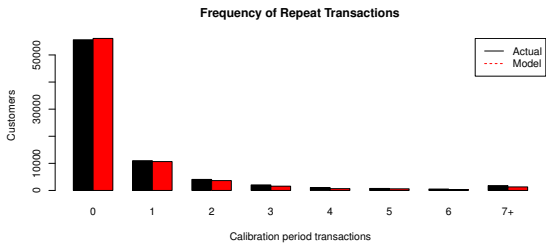

Figure 7: Frequency of repeat purchases according to the extended Pareto/NBD model by Abe (2009) with covariates.

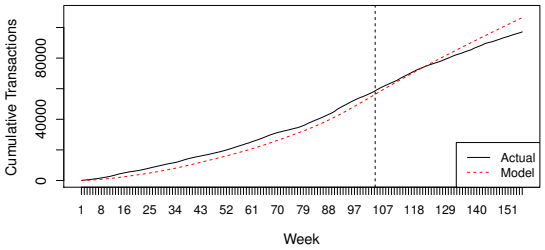

Figure 8: Actual and expected cumulative number of purchases per week for the standard Pareto/NBD model.

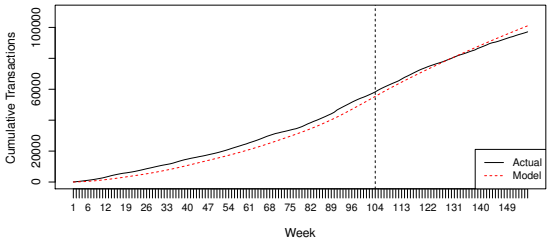

Figure 9: Actual and expected cumulative number of purchases per week for the extended Pareto/NBD model without covariates.

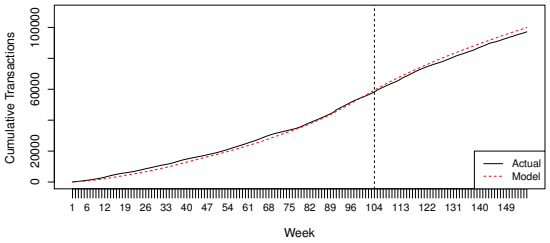

Figure 10: Actual and expected cumulative number of purchases per week for the extended Pareto/NBD model with covariates.

(2009) with covariates, respectively. These plots show that each model fits the data fairly well. The standard Pareto/NBD model seems to slightly overestimate the number of customers who made relatively many repeat purchases, while the extension by Abe (2009), both with and without covariates, seems to slightly underestimate the number of customers who made relatively many repeat purchases. In addition, both the in-sample fit and the out-of-sample fit are analysed by comparing the expected number of purchases per week of each model with the actual number of purchases per week, for both the training and test set. Figures 8, 9 and 10 show these expected number of purchases for the standard Pareto/NBD model, its extension by Abe (2009), and its extension by Abe (2009) with covariates, respectively. Note that the dashed vertical line indicates the end of the training period. In line with Figure 5, the standard Pareto/NBD model models the total number of purchases reasonably well, although it overestimates the number of purchases after roughly halfway through the testing period. The extended Pareto/NBD model fits the data quite well, and the inclusion of covariates leads to an even better fit. It overestimates the total number of purchases at the end of the test set, albeit barely. Since

there is a very small difference in model fit between the standard Pareto/NBD model and its extension without covariates, it seems that the extended model's assumptions about customer purchase behaviour are as realistic as those of the standard model.

One might wonder why the total number of purchases at the end of the test period, which is approximately 100,000, does not correspond to the total number of purchases in year 5 as displayed in Figure 1, which is approximately 120,000. This is because these results are based on a different time span, as there is a shift of roughly 0.5 years, and furthermore because customers who made no purchases in the training set were discarded from the data set. See Section 3.2.1 for details.

To obtain CLV predictions, the Pareto/NBD model and its extension are combined with the gamma-gamma submodel under the assumption that the distribution of average purchase values across customers is independent of the purchase process. To assess the validity of this assumption, Figure 11 shows a set of box plots that summarise the distribution of average purchase value, broken down by the number of repeat purchases in the training set. For example, the plot shows that the first repeat purchase by customers has a median purchase value of approximately €100. Although there exists a slight correlation (0.04) between the purchase value and the number of purchases, it is clear that the variation within each number-of-purchases group dominates the between-group variation. Therefore, I believe that this small correlation does not represent a substantial violation of the independence assumption.

Next, we validate the gamma-gamma submodel, which models the average purchase value

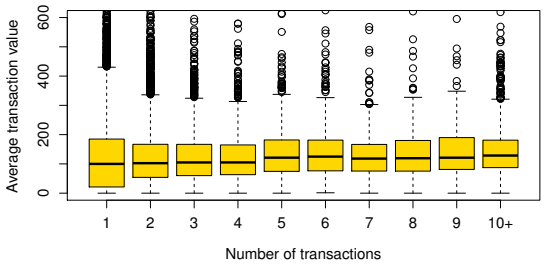

Figure 11: Set of box plots of average purchase value by frequency in the training set.

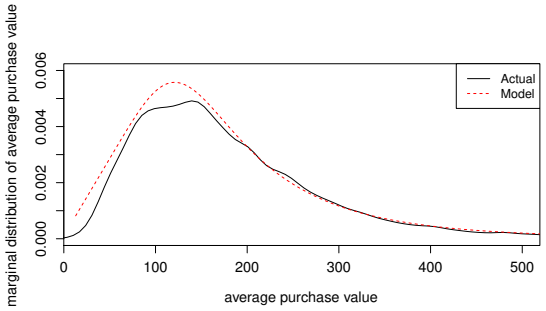

Figure 12: Actual and expected average purchase value across customers.

of customers. The theoretical mean and median purchase value of the fitted gamma-gamma distribution are €197 and €161 respectively, while the mean and median observed average repeat purchase value are €187 and €157 respectively. In addition, to visualise the fit of the model, the implied distribution of average purchase value across customers is compared with the non-parametric density of the observed average purchase values, which is shown in Figure 12. The graph is cut off at an average purchase value of €500 to increase visibility, as there are several customers with extremely high average purchase values up to €3500. The graph shows that the model expects there to be slightly more customers with an average purchase value between 0 and 200, and that it fits the number of customers with a average purchase value over €200 excellently. All in all, the gamma-gamma model fits the data reasonably well.

To obtain CLV predictions, the expected number of purchases by each customer in the test set, which is obtained from the Pareto/NBD model and its extension, is multiplied by the customer's expected average purchase value in the test set, which is obtained from the gamma-gamma submodel. The performance of the Pareto/NBD model and its extension with respect to predicting CLV are, along with the performance of the other two models, given in Section 4.4.

4.2 Duration model

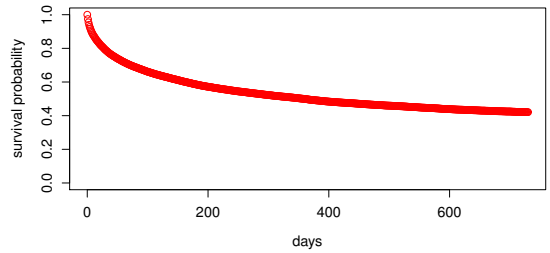

Figure 13: Kaplan-Meier estimate of the survival probability for purchasing.

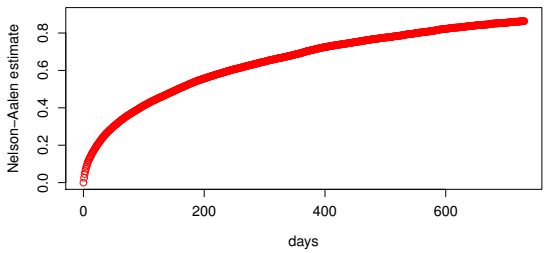

Figure 14: Nelson-Aalen estimate of the cumulative hazard rate function for purchasing.

First of all, the estimated survival and cumulative hazard rate function are analysed. Note that these estimates are similar for both the model with and without covariates.

Figure 13 shows the Kaplan-Meier survival function, where the x-axis represents the time in days and the y-axis shows the probability of survival. For example, the plot shows that the probability of surviving (i.e. not purchasing) 200 days after a customer's last purchase is approximately 60%. The probability of survival drops relatively quickly for roughly the first 100 days after a customer's last purchase. In other words, relatively many customers purchase within the first 100 days after their last purchase. After the first 100 days, the survival function flattens out, meaning that relatively few customers purchase after not having purchased for 100 days. Note that the survival probability does not converge to zero because each customer's last purchase is censored. Figure 14 shows the Nelson-Aalen cumulative hazard rate function. This plots leads us to the same conclusion: the estimate of the cumulative hazard rate function is steeper for the first 100 days, and there is therefore evidence that the 'risk' of purchasing is highest in the first 100 days after a customer's last purchase.

Next, we analyse the effect of incorporating covariates into the Cox proportional hazard model by looking at the estimated coefficients of the covariates. The estimated coefficients of the variables, along with a forest plot, are displayed in Figure 15. First of all, Figure 15 shows that each variable has a significant effect on the probability of survival as their p-values are sufficiently small. Furthermore, we can conclude from the forest plot that customers who are male, customers whose sex is not missing, customers whose age is not missing, customers who are subscribed to the newsletter, and customers who created an account on Winkelstraat.nl have an increased risk of purchasing. Note that since *Age* is a numerical variable, it cannot be interpreted using a forest plot. Nevertheless, since its coefficient is significantly larger than zero, we can conclude that older people have an increased risk of purchasing.

However, these results are only valid if the proportional hazard assumption holds. This assumptions states that explanatory variables are multiplicatively related to the hazard function, meaning that these variables change the risk of purchasing. They should not, however, change the time at which the hazard is high or low. This corresponds to a constant hazard ratio for different values of variables. Therefore, whether or not this assumption holds can be checked by comparing log-minus-log plots for different values of a variable, where log-minus-log plots are plots of the logarithm of the negative logarithm

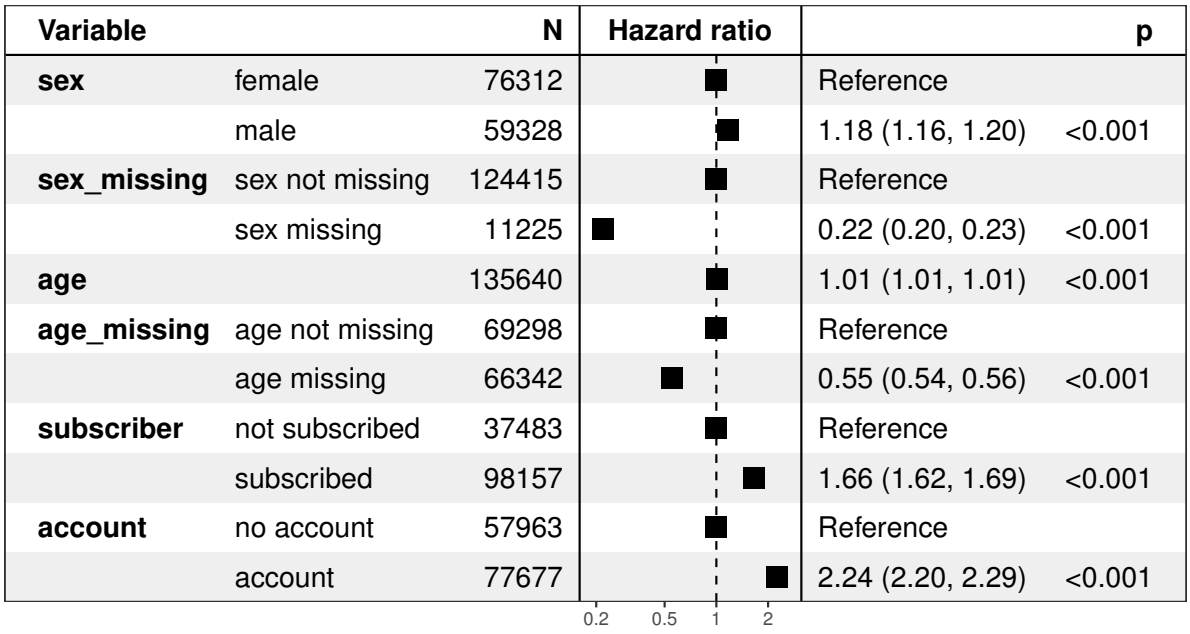

Figure 15: Summary statistics and corresponding forest plot of the covariates in the Cox proportional hazard model. The left panel shows the covariates, their different categories, and the number of customers that belong to the corresponding category. The central panel shows a forest plot, which displays relative hazard ratio's with respect to other categories of the corresponding covariate. The right panel shows the estimated coefficients, their 95% confidence interval, and their p-value.

of the survival function. Note that the logarithm of the negative logarithm of the survival function equals the logarithm of the cumulative hazard rate function (see Equation B.7). The proportional hazard assumption holds if the log-minus-log plots of different values of a variable are parallel. Figure 16 shows these plots for each variable. Note that since the variable *Age* is numerical, one cannot compare its different values. Therefore, in order to enable the proportional hazard assumption for this variable to be checked to some extent, two log-minus-log plots are compared with respect to whether the customer's age is below or above average. The log-minus-log plots of the variables *Sex* and *Sex Missing* diverge slightly, indicating that the proportional hazard assumption for these variables may not hold. The log-minus-log plots of the other variables are parallel, and hence the proportional hazard assumption holds for these variables.

The predictive performance of the duration model is, along with the performance of the other two models, given in Section 4.4.

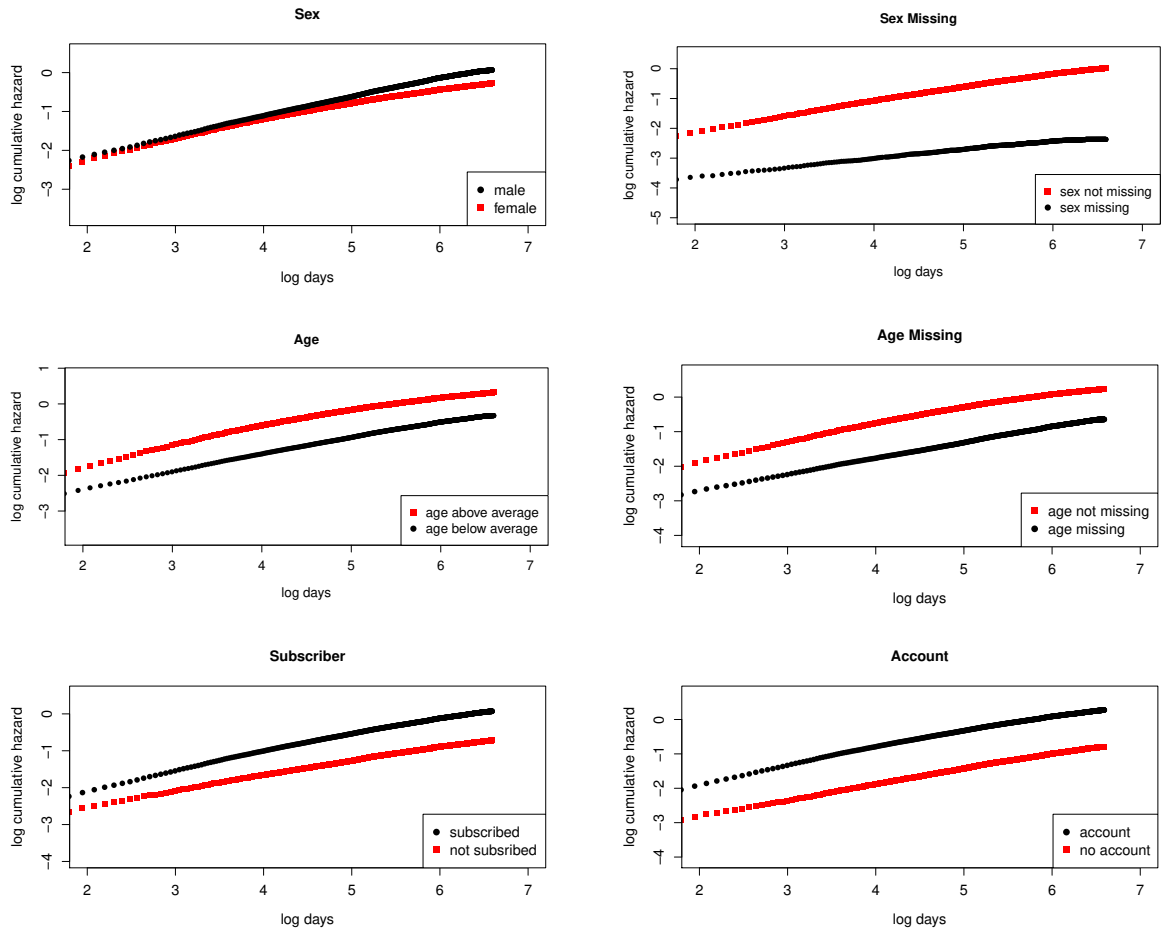

Figure 16: Log-minus-log plots of estimated survival functions for each covariate.

4.3 Gradient tree boosting

First of all, a grid search is performed to find the optimal values for the hyperparameters *max_depth* and *min_child_weight*. This grid search is shown in Figure 17 for both the model with the RMSE loss function and the model with the Fair loss function. Henceforth, these two models will be referred to as the RMSE model and the MAE model, respectively. Note that the latter model is called the MAE model because its performance is evaluated using the MAE, and because the Fair loss function approximates the MAE loss function. The figure shows that for the RMSE model the optimal values of *max_depth* and *min_child_weight* both lie around five. Therefore, to find the optimum values, a second grid search is performed which searches for values up to three above and below five for both hyperparameters. This second grid search shows that the optimum values are five and six for *max_depth* and *min_child_weight* respectively. Figure 17 shows that for the MAE model, the optimal value of *max_depth*

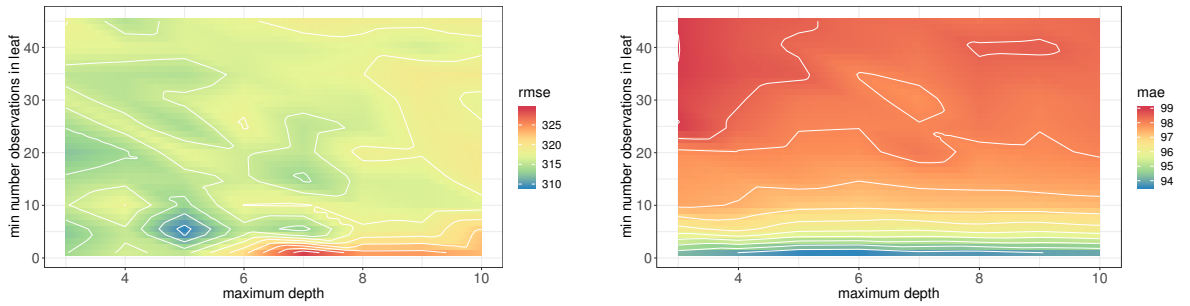

Figure 17: Grid search on *max_depth* and *min_child_weight* for the RMSE model (left) and the MAE model (right).

lies between 3 and 10, and the optimal value of *min_child_weight* lies between 1 and 3. Therefore, a second grid search is performed which searches for the optimal values in these smaller ranges. This second grid search shows that the optimal value is six for *max_depth* and one for *min_child_weight*.

Next, the hyperparameter γ is tuned by trying out different values ranging from 0 to 0.50, of which the corresponding loss for both the RMSE and MAE model is shown in Table 2. The table shows that the optimal value of γ is 0.05 for the RMSE model and 0 for the MAE model. For the MAE model, the hyperparameter γ does not seem to influence the model's performance by much, as the MAE ranges from 93.72 to 93.80. Therefore, the model's performance is also analysed for higher values of γ up to 5. The model's performance does not improve, however, and even declines for large values of γ. Therefore, we set γ to 0 for the MAE model. Note that the lowest obtained RMSE and MAE may be higher than was obtained when tuning *max_depth* and *min_child_weight* due to the use of randomisation in the model's algorithm. This also applies to all results in the remainder of this section.

Table 2: Loss for different values of γ for both the RMSE and the MAE model.

γ	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
RMSE	314.6	311.1	311.7	314.1	314.3	316.0	314.8	315.8	313.3	317.2	314.4
MAE	93.72	93.77	93.78	93.76	93.74	93.77	93.80	93.80	93.77	93.79	93.73

Furthermore, a grid search is performed to find the optimal values for the hyperparameters *subsample* and *colsample*. This grid search is shown in Figure 18 for both the RMSE and MAE model. The figure shows that the optimal values of *subsample* and *colsample* for the RMSE model are 0.7 and 0.8 and for the MAE model 0.9 and 1, respectively.

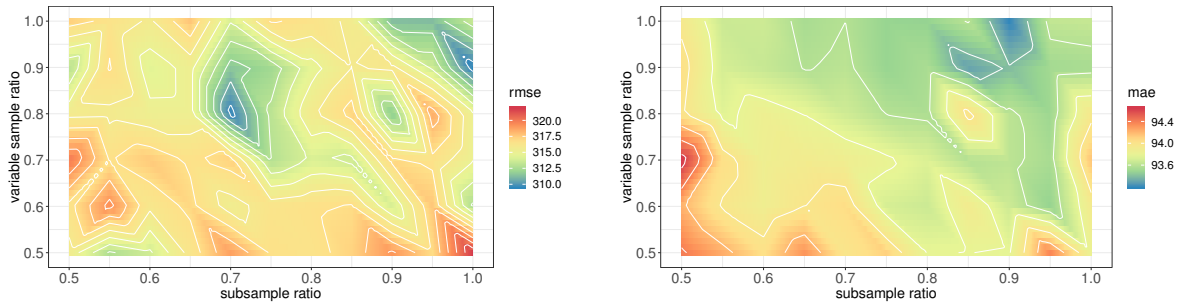

Figure 18: Grid search on *subsample* and *colsample* for the RMSE model (left) and the MAE model (right).

Subsequently, a grid search is performed to find the optimal values for the hyperparameters α and λ. This grid search is shown in Figure 19 for both the RMSE and the MAE model. For the RMSE model, the optimal values of α and λ are both 0.1. The grid search provides insufficient evidence on the optimal values for the MAE model. Therefore, a second grid search is performed which searches for $\alpha \in \{30, 50, 100\}$, while keeping the range of values for λ equal. This grid search shows that the optimal values of α and λ are 10 and 0.01, respectively.

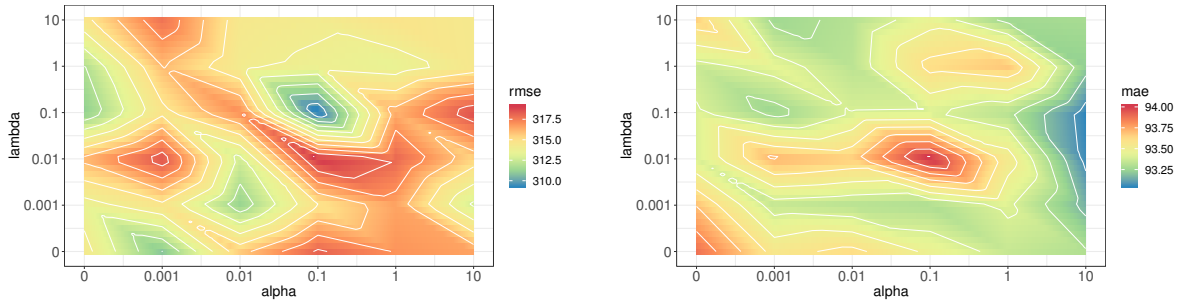

Figure 19: Grid search on α and λ for the RMSE model (left) and the MAE model (right).

Finally, the learning rate is lowered to increase the model's performance. To see the effect of lowering the learning rate, multiple learning rates are considered, and their corresponding loss is displayed in Table 3 for both the RMSE and the MAE model. One can see that a lower learning rate increases the model's performance, although a learning rate lower than 0.010 does not seem to improve the performance much. The optimal learning rate for the RMSE model is 0.001, with a corresponding optimal value of 2746 for the hyperparameter *rounds*. For the MAE model, the optimal learning rate is 0.010, with a corresponding optimal value of 4207 for the hyperparameter *rounds*. Note that the MAE model, even though it uses a higher learning rate, requires more boosting

rounds than the RMSE model, meaning that it converges slower and is computationally more expensive.

Table 3: Loss for different values of β for both the RMSE and the MAE model.

β	0.001	0.005	0.010	0.050	0.100
RMSE	312.7	314.5	312.8	315.4	316.3
MAE	93.1	93.2	93.0	93.2	93.2

After the optimal values of the hyperparameters are found, the model is trained once more with these parameter values, but this time without using cross-validation. After this final model is trained, the importance of the variables can be assessed by looking at the fractional contribution of each variable to the model based on the total gain (reduction of loss) of each variable's splits. A higher percentage indicates a more important variable. Figure 20 shows this percentage of the ten most important variables for both the RMSE and the MAE model. For the RMSE model, the monetary value and the frequency of customers are the most important variables, followed by their recency, return percentage, and age. The other variables are of little to no importance. For the MAE model, the frequency is the most important variable, followed by the monetary value and the recency. Furthermore, the percentage of returned items is of some importance, and the other variables are of no importance. Note that, for both models, the recency of customers should have more importance in studies in which the training data are measured over a longer time span, as in this case a low recency will most likely mean that the customer has

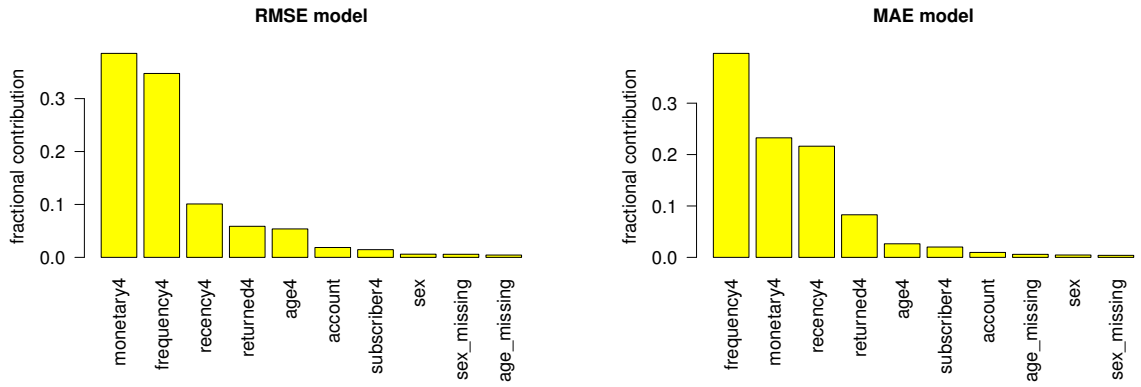

Figure 20: Fractional contribution of each variable to the model based on the total gain (reduction of loss) of each variable's splits, for both the RMSE model (left) and the MAE model (right).

become inactive. In both models, the dummy variables indicating a customer's favourite brand are not in the top 10 most important variables, having fractional contributions of less than 0.1%. Note that, although one can assess the importance of variables by means of their contribution to the model, one cannot determine what the effect of the corresponding variable is. For example, Figure 20 does not show whether older or younger people generally have a higher CLV.

The predictive performance of the gradient tree boosting model is, along with the performance of the other two models, given in Section 4.4.

4.4 Overall test set performance

Table 4: Predictive performance with respect to the test set of each applied method. GG is an abbreviation for the gamma-gamma submodel, and PHM is an abbreviation for the proportional hazard model. For each performance measure, the best model's performance is displayed in bold.

	Domain 1		**Domain 2**	**Domain 3**
	RMSE	MAE	RP (%)	CBPD (%)
1 Benchmarks				
1.1 Mean spend year 4	240.7	142.4	-	114.9
1.2 Status quo	256.0	122.3	34.2	114.9
1.3 Predicting zero	238.4	58.0	-	-100.0
2 Pareto/NBD + GG				
2.1 Standard Pareto/NBD	196.4	85.6	38.4	36.2
2.2 Abe (2009)	194.6	80.9	38.2	18.6
2.3 Abe (2009) + covariates	**191.6**	76.3	**40.1**	**14.6**
3 Duration model				
3.1 Non-parametric	195.0	66.8	35.6	-34.7
3.2 Cox PHM	194.3	67.8	38.5	-15.1
4 Gradient tree boosting				
4.1 RMSE loss function	193.1	80.6	39.6	17.1
4.2 Fair loss function	207.2	**54.7**	37.4	-71.5

This section provides a comparison between the predictive performance with respect to the test set of each applied model. Table 4 shows the RMSE, MAE, RP, and CBPD for each model. Recall that RP measures the model's ability to identify the top 10% of highest spending customers, and that the CBPD measures the percentage deviation of the model's predicted total customer base from the true value of the total customer base. In addition, Table 4 shows these four measures for three simple benchmark models. The

first benchmark model predicts all customers' average total spends in the previous year, for each customer. The second benchmark model is a status quo model that predicts each customer's total spend in the previous year. The third benchmark predicts zero spend for all customers. Furthermore, recall that this research focuses on the predictive performance of the models and less on the degree of interpretability of the results, as explained in Section 1.

Table 4 shows that the lowest RMSE obtained by the benchmark models is approximately 238 and the lowest MAE is 58. Both of these values are obtained by the model that predicts no spend for all customers. A MAE of 58 is a relatively low value, which is caused by the fact that the majority of the customers make no purchases in the test set. Hence, for the majority of the customers, predicting no spend for the forthcoming year is an excellent prediction. However, this benchmark model has a high RMSE compared to the non-benchmark models, which implies that the models should not be evaluated by their MAE alone but rather by their RMSE and MAE together. Therefore, a MAE higher than 58 obtained by the non-benchmark models does not necessarily mean that these models are bad predictors; however, these models are expected to have a MAE lower than 122, the second best MAE obtained by the benchmark models. With respect to the second domain, the status quo benchmark model obtains an RP of 34.2, and it is expected that the non-benchmark models will improve on this. The other two benchmark models do not predict future spend individually and are therefore not able to rank customers. With respect to the third domain, the benchmark models perform very poorly, as their prediction of the total customer base is twice as low or more than twice as high as the actual customer base.

The Pareto/NBD model and its extension perform better with respect to RMSE, RP and CBPD than the benchmark models. With respect to the MAE, they outperform the first two benchmark models but do not outperform the third benchmark model which predicts zero spend. However, I believe that their RMSE/MAE trade-off is better than that of all benchmark models. As expected from the analysis of the models' fit in Section 4.1, both the standard and extended model have a positive CBPD. Apparently, since CBPDs of 36.2, 18.6, and 14.6 only partially reflect their overestimation of the total number of purchases at the end of the test set (see Figures 8, 9, and 10), these models predict a

relatively large number of purchases for high spending customers and a relatively small number of purchases for low spending customers. When comparing the Pareto/NBD model's extension that uses covariates to the one that does not, one can see that the inclusion of covariates in the model increases its performance with respect to all domains. When comparing the standard Pareto/NBD model to its extension that uses covariates, we can see that the standard model is outclassed in all domains. The extended model performs even better without the inclusion of covariates as it has a better RMSE, MAE, and CBPD, and a similar RP. Apparently, since the extended Pareto/NBD model performs better than the standard Pareto/NBD model, the assumptions that the extended model makes about the purchase behaviour of customers may be more realistic than those of the standard model.

The duration model clearly outperforms the benchmark models, having lower RMSEs, better RMSE/MAE trade-offs, higher RPs, and better CBPDs. When comparing the duration model with covariates (Cox PHM) to the one without covariates (non-parametric), one can see that the inclusion of covariates in the duration model results in a slightly lower RMSE but a slightly higher MAE. However, it leads to significantly better results with respect to the second and third domain, and I therefore consider the duration model with covariates superior to the one without covariates. When comparing the duration model to the probability models, one can see that the best probability model (extended Pareto/NBD model with covariates) surpasses the duration model (with covariates) with respect to RMSE, RP, and CBPD, but not MAE. However, I believe that the probability model's better RMSE, RP, and CBPD outweigh the duration model's lower MAE.

Both gradient tree boosting models outperform the benchmark models, having lower RMSEs, better RMSE/MAE trade-offs, higher RPs, and better CBPDs. Moreover, the gradient tree boosting model that uses the Fair loss function has the lowest MAE of all models, including the benchmark model that only predicts zero spends. However, this low MAE comes at the expense of a reasonably high RMSE. Furthermore, it struggles to predict the value of the total customer base. Therefore, I believe the gradient tree boosting model that uses the RMSE loss function is a more adequate model for predicting CLV, as it has a better RMSE, RP, and CBPD. When comparing the best machine learning

model (with RMSE loss function) to the duration model (with covariates), it does not become clear which model has a better predictive performance, as the machine learning model has a better RMSE and RP but a worse MAE and CBPD. When comparing the best machine learning model to the best probability model, one sees that the probability model outperforms the machine learning technique with respect to all domains.

Note that for the machine learning technique, the RMSE and the MAE obtained on the test set is lower than those obtained during the parameter tuning on the training set, whereas one usually obtains a lower error on the (cross-validated) training set. This phenomenon is caused by the fact that the model is trained using data on customers who made a purchase at least once in year 3, or in other words, on active customers. Once the model is trained, the data on all customers in year 4 are inserted into the model, that is, including customers who did not purchase in year 4 (inactive customers). Since the CLV of inactive customers is generally much easier to predict than that of active customers, and since inactive customers were not considered during the training of the model, the RMSE and MAE on the test set will be lower than those obtained on the training set. Furthermore, the RMSE and MAE obtained on the training set are based on predicted and actual CLV that had not yet been discounted.

All in all, the extended Pareto/NBD model with covariates is the best model for predicting CLV. It performs better than any other model except with respect to MAE. In other words, probability models outperform duration and machine learning models with respect to predicting CLV. Furthermore, the duration model and machine learning technique perform similarly.

5 Conclusion

The prediction of CLV is important for online retailers as this allows them to increase their profits by allocating disproportionate marketing resources to their most profitable customers. The aim of this research was to compare different classes of prediction models by their predictive power with respect to predicting CLV. This research aim arose from the question whether machine learning methods that directly predict CLV

can compete with more traditional CLV methods. The research focused primarily on the predictive power of the models and less on the degree of interpretability of their results. The different classes of models included probability models, duration models, and machine learning techniques. The models applied are the Pareto/NBD model, the Cox proportional hazard model, and the gradient tree boosting model, which represent the probability models, duration models, and machine learning techniques, respectively. In addition, several extensions of or different implementations to these models were considered. In terms of predictive power, the extended Pareto/NBD probability model with covariates performed best. Seemingly, the strict assumptions that the probability model makes about customer purchase behaviour fits our problem better than the less restrictive duration model and machine learning technique. Furthermore, the duration model and the machine learning model performed similarly. However, when taking the interpretability of the results into account, machine learning models might be considered as inferior to the two other classes of models as it lacks interpretability, while probability and duration models provide excellent interpretation of their results.

The length of the training and test sets, and hence the length of the prediction horizon, was restricted by the machine learning technique as it requires labelled training data. If this technique had been excluded from the comparison of CLV models, a larger prediction horizon and a larger training set could have been used. The probability models and the duration model would most likely benefit from this larger training set, increasing their predictive power. Moreover, it is questionable whether gradient tree boosting is a suitable prediction technique when the prediction horizon is relatively large. Since the training set should consist of a time span at least as long as that of the test set (see Section 3.2.1), the model would be trained using relatively old data, which would compromise the model's validity if the data are not stationary. The probability models and the duration model do not suffer from this restriction.

A possible direction for future work with respect to the Pareto/NBD model would be to relax the assumption of constant time-invariant purchase rate, which is captured by a Poisson distribution, and replace it by a more realistic assumption. One might consider the BG/CNBD-k model (Mzoughia and Limam, 2014), which generalises the Poisson distribution to a two-parameter distribution, offering more flexibility and a better fit to

real world data. The performance of the duration model could be improved by making use of recurrent duration models. Since customers are able to purchase multiple times, the data consist of correlated data instances. Instead of treating each purchase as an independent observation, recurrent duration models take this correlated structure of the data into account. This would allow one, for example, to compute the probability that a customer makes the same number of purchases as in their previous year, which would presumably improve the predictions. Likewise, instead of using gradient tree boosting, one could use a machine learning technique that models temporal dynamic behaviour in time sequences, such as recurrent neural networks (RNNs). The ability of RNNs to recognise patterns across time in customer purchase behaviour could result in better predictive performance.

6 References

Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and Event History Analysis: A Process Point of View*. Statistics for Biology and Health. Springer New York.

Abe, M. (2009). "Counting your customers" one by one: A hierarchical Bayes extension to the pareto/NBD model. *Marketing Science*, 28(3):541–553.

Allenby, G. M., Leone, R. P., and Jen, L. (1999). A dynamic model of purchase timing with application to direct marketing. *Journal of the American Statistical Association*, 94(446):365–374.

Bae, S. M., Ha, S. H., and Park, S. C. (2005). A web-based system for analyzing the voices of call center customers in the service industry. *Expert Systems with Applications*, 28(1):29–41.

Baesens, B., Verstraeten, G., Van den Poel, D., Egmont-Petersen, M., Van Kenhove, P., and Vanthienen, J. (2004). Bayesian network classifiers for identifying the slope of the customer lifecycle of long-life customers. *European Journal of Operational Research*, 156(2):508–523.

Bitran, G. R. and Mondschein, S. V. (1996). Mailing decisions in the catalog sales industry. *Management Science*, 42(9):1364–1381.

Bolton, R. N. (1998). A dynamic model of the duration of the customer's relationship with a continuous service provider: The role of satisfaction. *Marketing Science*, 17(1):45–65.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Buckinx, W. and 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*, 164(1):252–268.

Chamberlain, B. P., Cardoso, A., Liu, C. H., Pagliari, R., and Deisenroth, M. P. (2017). Customer lifetime value prediction using embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, pages 1753–1762. ACM.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., and Li, Y. (2018). *XGBoost: Extreme Gradient Boosting*. R package version 0.71.2.

Cheung, K.-W., Kwok, J. T., Law, M. H., and Tsui, K.-C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2):231–243.

Datta, P., Masand, B., Mani, D., and Li, B. (2000). Automated cellular modeling and prediction on a large scale. *Artificial Intelligence Review*, 14(6):485–502.

Donkers, B., Verhoef, P. C., and de Jong, M. G. (2007). Modeling clv: A test of competing models in the insurance industry. *Quantitative Marketing and Economics*, 5(2):163–190.

Drew, J. H., Mani, D., Betz, A. L., and Datta, P. (2001). Targeting customers with statistical and data-mining techniques. *Journal of Service Research*, 3(3):205–219.

Dziurzynski, L., Wadsworth, E., and McCarthy, D. (2014). *BTYD: Implementing Buy 'Til You Die Models*. R package version 2.4.

Ehrenberg, A. S. (1959). The pattern of consumer purchases. *Applied Statistics*, pages 26–41.

Etzion, O., Fisher, A., and Wasserkrug, S. (2005). e-CLV: A modeling approach for customer lifetime evaluation in e-commerce domains, with an application and case study for online auction. *Information Systems Frontiers*, 7(4-5):421.

Fader, P. S. and Hardie, B. G. (2007). Incorporating time-invariant covariates into the Pareto/NBD and BG/NBD models.

Fader, P. S. and Hardie, B. G. (2009). Probability models for customer-base analysis. *Journal of Interactive Marketing*, 23(1):61–69.

Fader, P. S., Hardie, B. G., and Lee, K. L. (2005). RFM and CLV: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning (ICML 1996)*, volume 96, pages 148–156. Citeseer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Glady, N., Baesens, B., and Croux, C. (2009). A modified Pareto/NBD approach for predicting customer lifetime value. *Expert Systems with Applications*, 36(2):2062–2071.

Gönül, F. F., Kim, B.-D., and Shi, M. (2000). Mailing smarter to catalog customers. *Journal of Interactive Marketing*, 14(2):2–16.

Gupta, S. and Lehmann, D. (2005). Managing customers as investments - the strategic value of customers in the long run.

Gupta, S., Lehmann, D. R., and Stuart, J. A. (2004). Valuing customers. *Journal of Marketing Research*, 41(1):7–18.

Hastie, T., Tibshirani, R., , and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York.

Hung, S.-Y., Yen, D. C., and Wang, H.-Y. (2006). Applying data mining to telecom churn management. *Expert Systems with Applications*, 31(3):515–524.

Khajvand, M., Zolfaghar, K., Ashoori, S., and Alizadeh, S. (2011). Estimating customer lifetime value based on RFM analysis of customer purchase behavior: Case study. *Procedia Computer Science*, 3:57–63.

Knott, A., Hayes, A., and Neslin, S. A. (2002). Next-product-to-buy models for cross-selling applications. *Journal of Interactive Marketing*, 16(3):59–75.

Koh, H. C. and Gerry, C. K. L. (2002). Data mining and customer relationship marketing in the banking industry. *Singapore Management Review*, 24(2):1.

Korkmaz, E., Kuik, R., and Fok, D. (2013). "Counting Your Customers": When will they buy next? An empirical validation of probabilistic customer base analysis models based on purchase timing. *ERIM Report Series Research in Management*, (ERS-2013-001-LIS).

Levinthal, D. A. and Fichman, M. (1988). Dynamics of interorganizational attachments: Auditor-client relationships. *Administrative Science Quarterly*, pages 345–369.

Lewis, M. (2006). Customer acquisition promotions and customer asset value. *Journal of Marketing Research*, 43(2):195–203.

Liu, D.-R. and Shih, Y.-Y. (2005). Integrating AHP and data mining for product recommendation based on customer lifetime value. *Information & Management*, 42(3):387–400.

Malthouse, E. C. and Blattberg, R. C. (2005). Can we predict customer lifetime value? *Journal of Interactive Marketing*, 19(1):2–16.

Moro, S., Cortez, P., and Rita, P. (2015). Using customer lifetime value and neural networks to improve the prediction of bank deposit subscription in telemarketing campaigns. *Neural Computing and Applications*, 26(1):131–139.

Morrison, D. G. (1968). Analysis of consumer purchase data: A bayesian approach. *IMR; Industrial Management Review (pre-1986)*, 9(2):31–40.

Morrison, D. G. (1978). On linearly increasing mean residual lifetimes. *Journal of Applied Probability*, 15(3):617–620.

Mzoughia, M. B. and Limam, M. (2014). An improved BG/NBD approach for modeling purchasing behavior using COM-Poisson distribution. *International Journal of Modeling and Optimization*, 4(2):141.

Pfeifer, P. E. and Carraway, R. L. (2000). Modeling customer relationships as Markov chains. *Journal of Interactive Marketing*, 14(2):43.

Platzer, M. (2016). *BTYDplus: Probabilistic Models for Assessing and Predicting your Customer Base*. R package version 1.0.1.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Reinartz, W. J. and Kumar, V. (2000). On the profitability of long-life customers in a non-contractual setting: An empirical investigation and implications for marketing. *Journal of Marketing*, 64(4):17–35.

Reinartz, W. J. and Kumar, V. (2003). The impact of customer relationship characteristics on profitable lifetime duration. *Journal of Marketing*, 67(1):77–99.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge University Press.

Rodrıguez, G. (2005). Non-parametric estimation in survival models.

Rossi, P., Allenby, G., and McCulloch, R. (2005). Bayesian statistics and marketing.

Schmittlein, D. C., Morrison, D. G., and Colombo, R. (1987). Counting your customers: Who are they and what will they do next? *Management Science*, 33(1):1–24.

Schmittlein, D. C. and Peterson, R. A. (1994). Customer base analysis: An industrial purchase process application. *Marketing Science*, 13(1):41–67.

Shih, Y.-Y. and Liu, C.-Y. (2003). A method for customer lifetime value ranking — combining the analytic hierarchy process and clustering analysis. *Journal of Database Marketing & Customer Strategy Management*, 11(2):159–172.

Silverman, B. (1986). Density Estimation for Statistical Analysis.

Sohrabi, B. and Khanlari, A. (2007). Customer lifetime value (CLV) measurement based on RFM model.

Song, H. S., Kim, J. K., Cho, Y. B., and Kim, S. H. (2004). A personalized defection detection and prevention procedure based on the self-organizing map and association rule mining: Applied to online game site. *Artificial Intelligence Review*, 21(2):161–184.

Therneau, T. M. (2015). *A Package for Survival Analysis in R.* version 2.38.

Venkatesan, R. and Kumar, V. (2004). A customer lifetime value framework for customer selection and resource allocation strategy. *Journal of Marketing*, pages 106–125.

Appendices

Appendix A: Probability Models

This section includes a detailed description the Pareto/NBD model, along with a description of the gamma-gamma submodel and the extension of the Pareto/NBD model by Abe (2009). The Pareto/NBD model is a probability model that predicts the number of purchases of customers in future periods and the probability that these customers are still active purchasers. In combination with the gamma-gamma submodel, it provides estimates of customer spends in future periods. The extension of the model by Abe (2009) relaxes several assumptions that the Pareto/NBD model makes and uses a different estimation technique. It also allows for the incorporation of time-invariant covariates.

First of all, the Pareto/NBD model will be described in detail. Thereafter, the gamma-gamma submodel and finally the extension of the Pareto/NBD model by Abe (2009) will be covered.

A.1 Pareto/NBD model

The Pareto/NBD model (Schmittlein *et al.*, 1987) is a probability model that uses the past transaction data of customers to predict their number of purchases in a future period. Furthermore, it also computes the probability that a customer is still actively making purchases. The model can only be applied to non-contractual, continuous data, i.e. the company does not observe the time a customer becomes inactive and a customer's purchases can occur at any point in time. It is assumed that a customer is 'alive' (actively purchasing) for an unobserved period of time and finally 'dies' (becomes permanently inactive). The description of the Pareto/NBD model in this section is based on the description as given in Schmittlein *et al.* (1987).

A.1.1 Model description

The Pareto/NBD model makes the following five assumptions on customer purchase behaviour:

1. While a customer is alive, his number of purchases is characterised by a Poisson distribution with purchasing rate λ.
2. The customer's lifetime is characterised by an exponential distribution with death rate μ.
3. Heterogeneity in the purchasing rate λ across customers is captured by a gamma distribution with parameters r and α.
4. Heterogeneity in the death rate μ across customers is captured by a gamma distribution with parameters s and β.
5. The purchasing rate λ and the death rate μ are distributed independently of each other.

An important feature of the Pareto/NBD model is that it only requires the recency, frequency, and observation length of customers' transaction data to predict their future purchase behaviour. Here, recency is the time since the last purchase of a customer, where a higher recency indicates that the customer has bought more recently. Furthermore, frequency is the total number of repeat purchases that a customer has made during the observation period, that is, the total number of purchases excluding the first. In other words, the Pareto/NBD model only requires aggregated transaction data of customers, and does not require other information, such as exact purchase times. For each customer, this information can be written as (X, t, T), where X is the frequency, t the recency, and T the length of the observation period. This notation allows us to mathematically express the previous five assumptions as follows:

1. While a customer is alive, his number of purchases is characterised by a Poisson

distribution with purchasing rate λ:

$$P[X = x|\lambda, \tau, T] = \begin{cases} \dfrac{(\lambda T)^x}{x!}e^{-\lambda T} & \text{if } \tau > T \\ \dfrac{(\lambda \tau)^x}{x!}e^{-\lambda \tau} & \text{if } \tau \leqslant T \end{cases} \quad \text{for } x = 1, 2, ... \tag{A.1}$$

2. The customer's lifetime is characterised by an exponential distribution with death rate μ:

$$f(\tau) = \mu e^{-\mu\tau} \quad \text{for } \tau \geqslant 0. \tag{A.2}$$

3. Heterogeneity in the purchasing rate λ across customers is captured by a gamma distribution with parameters r and α:

$$g(\lambda|r, \alpha) = \frac{\alpha^r}{\Gamma(r)}\lambda^{r-1}e^{-\alpha\lambda}, \tag{A.3}$$

where

$$\Gamma(r) = \int_0^\infty t^{r-1}e^{-t}\mathrm{d}t. \tag{A.4}$$

4. Heterogeneity in the death rate μ across customers is captured by a gamma distribution with parameters s and β:

$$h(\mu|s, \beta) = \frac{\beta^s}{\Gamma(s)}\mu^{s-1}e^{-\beta\mu}, \tag{A.5}$$

where

$$\Gamma(s) = \int_0^\infty t^{s-1}e^{-t}\mathrm{d}t. \tag{A.6}$$

5. The purchasing rate λ and the death rate μ are distributed independently of each other.

Assumptions 1 and 3 imply that a customer's number of purchases is characterised by a Poisson distribution with gamma-distributed purchasing rate, which is also known as the negative binomial distribution (NBD):

$$P[X = x|r, \alpha, \tau > T] = \binom{x + r - 1}{x}\left(\frac{\alpha}{\alpha + T}\right)^r\left(\frac{T}{\alpha + T}\right)^x \quad \text{for } x = 0, 1, 2, ... \tag{A.7}$$

Furthermore, assumptions 2 and 4 imply that a customer's lifetime is characterised by an exponential distribution with gamma-distributed death rate, which is also known as the Pareto distribution:

$$f(\tau|s,\beta) = \frac{s}{\beta}\left(\frac{\beta}{\beta+\tau}\right)^{s+1} \quad \text{for } \tau > 0. \tag{A.8}$$

The Pareto/NBD model owes its name to the fact that it is based on these two distributions.

A.1.2 Estimation

Estimating the Pareto/NBD model involves estimating the four model parameters r, α, s, and β. These parameters can be estimated by maximising the likelihood for observed transaction data, which is given by

$$L(r,\alpha,s,\beta) = \prod_{i=1}^{M} P[X_i = x_i, t_i, T_i|r,\alpha,s,\beta], \tag{A.9}$$

where M is the number of customers. An expression for $P[X_i = x_i, t_i, T_i|r,\alpha,s,\beta]$ can be derived by splitting this probability into the case where customer i is still alive at time T, and the case where he is not, which results in the following expression:

$$\begin{aligned} P[X = x, t, T|r,\alpha,s,\beta] = \\ \int_0^\infty \int_0^\infty P[X = x, t, T|\lambda,\mu,\tau > T]\ P[\tau > T|\lambda,\mu]\ g(\lambda|r,\alpha)\ h(\mu|s,\beta)\ \mathrm{d}\lambda\mathrm{d}\mu\ + \\ \int_0^\infty \int_0^\infty P[X = x, t, T|\lambda,\mu,t < \tau < T]\ P[t < \tau < T|\lambda,\mu] \\ g(\lambda|r,\alpha)\ h(\mu|s,\beta)\mathrm{d}\lambda\mathrm{d}\mu, \end{aligned} \tag{A.10}$$

where $g(\lambda|r,\alpha)$ and $h(\mu|s,\beta)$ are given in A.3 and A.5. This expression contains four new probabilities for which we need to derive expressions. Given the probability density function of the exponential distribution in Equation A.2, we obtain an expression for the

probability that a customer is alive at time T

$$P[\tau > T|\lambda, \mu] = 1 - P[\tau \leqslant T|\lambda, \mu] = e^{-\mu T} \tag{A.11}$$

and for the probability that a customer is not alive at time T

$$P[t < \tau < T|\lambda, \mu] = P[\tau \leqslant T|\lambda, \mu] - P[\tau \leqslant t|\lambda, \mu] = e^{-\mu t} - e^{-\mu T}. \tag{A.12}$$

To derive expressions for the remaining two probabilities, that is, the probability of observing $X = x$ repeat purchases with the last purchase at time t, we first introduce the variable ζ_x, which denotes the time of the x-th purchase. Using Equation A.1, we can see that ζ_x is the sum of i.i.d. exponentially distributed interpurchase times, each with mean $1/\lambda$. This means that ζ_x is characterised by a gamma distribution with shape parameter x and rate parameter $1/\lambda$, with density function

$$f_\zeta(\zeta_x|x, \lambda) = \frac{\lambda^x}{\Gamma(x)} \zeta_x^{x-1} e^{-\lambda \zeta_x}, \tag{A.13}$$

where the gamma function $\Gamma(x)$ is similar to the ones in Equations A.4 and A.6. Furthermore, let the event that no purchase occurred in the interval $(t, T]$ be denoted as ϕ_{T-t}. Because it is assumed that the number of purchases are characterised by a Poisson distribution (assumption 1), the probability that ϕ_{T-t} occurs equals $e^{-\lambda(T-t)}$. Now, the event of observing $X = x$, t, and T can be captured by the events $\zeta_x = t$ and ϕ_{T-t} together. Note that the events $\zeta_x = t$ and ϕ_{T-t} are independent of each other when conditioned on $\tau > T$. This allows us to derive an expression of the probability of observing $X = x$, t and T, given λ, μ and $\tau > T$:

$$P[X = x, t, T|\lambda, \mu, \tau > T] = f_\zeta(t|\lambda, \mu, \tau > T) \ * \ P[\phi_{T-t}|\lambda, \mu, \tau > T] = \frac{\lambda^x t^{x-1}}{\Gamma(x)} e^{-\lambda t} \ * \ e^{-\lambda(T-t)} = \frac{\lambda^x t^{x-1} e^{-\lambda T}}{\Gamma(x)}. \tag{A.14}$$

To obtain an expression for this probability for the case where $t < \tau < T$, we need to condition on $\tau = y$ and use the law of total probability:

$$P[X = x, t, T|\lambda, \mu, t < \tau < T] =$$
$$\int_t^T f_\zeta(t|\lambda, \mu, y > t) \ * \ P[\phi_{y-t}|\lambda, \mu, y > t] \ * \ f_\tau(y|\lambda, \mu) \ \mathrm{d}y =$$
$$\int_t^T \frac{\lambda^x t^{x-1}}{\Gamma(x)} e^{-\lambda t} \ * \ e^{-\lambda(y-t)} \ * \ \mu e^{-\mu y} \ \mathrm{d}y = \int_t^T \frac{\lambda^x t^{x-1} e^{-\lambda y}}{\Gamma(x)} \mu e^{-\mu y} \ \mathrm{d}y, \quad \text{(A.15)}$$

where $f_\tau(.)$ is the probability density function of τ as specified in Equation A.2, and $f_\zeta(.)$ is the probability density function of ζ_x as specified in Equation A.13.

Finally, substituting Equations A.3, A.5, A.11, A.12, A.14, A.15 into Equation A.10, and thereafter substituting Equation A.10 into Equation A.9 for all customers yields the expression for the likelihood function. The likelihood function is then maximised by a numerical optimisation algorithm, as explicit solutions for the maximum likelihood estimates do not exist.

A.1.3 Forecasting

The goal is to forecast the number of customer purchases X^* in interval $(T, T^*]$. To derive an expression of this forecast we make use of the following results. First of all, we exploit the memoryless property of the exponential distribution, which implies that, given that a customer is alive at time T, his remaining lifetime is again exponentially distributed with death rate μ. Morrison (1978) used Bayesian updating to show that, given the observed transaction data (X, t, T) of customers, this death rate μ is again gamma distributed, but with updated parameters $s^* = s$ and $\beta^* = \beta + T$. Likewise, given that a customer is alive at time T, the number of purchases in his remaining lifetime follows again a Poisson process with purchasing rate λ. Furthermore, it can be shown using Bayesian updating that λ follows again a gamma distribution, but with updated parameters $r^* = r + x$ and $\alpha^* = \alpha + T$ (Morrison, 1968). In addition, we use the fact that dead customers at time T do not make future purchases. These results allow us to specify the distribution of X^*

as follows:

$$P[X^* = x^*|r, \alpha, s, \beta, X = x, t, T, T^*] =$$
$$P[X^* = x^*|r, \alpha, s, \beta, X = x, t, T, T^*, \tau > T] \ * \ P[\tau > T|r, \alpha, s, \beta, X = x, t, T] =$$
$$P[X^* = x^*|r + x, \alpha + T, s, \beta + T, T^*] \ * \ P[\tau > T|r, \alpha, s, \beta, X = x, t, T]. \quad \text{(A.16)}$$

Using Equation A.16, the expected number of purchases in the time interval $(T, T^*]$ can now be expressed as

$$\mathbb{E}[X^*|r, \alpha, s, \beta, X, t, T, T^*] = \mathbb{E}[X^*|r + x, \alpha + T, s, \beta + T, T^*] \ *$$
$$P[\tau > T|r, \alpha, s, \beta, X = x, t, T]. \quad \text{(A.17)}$$

The first term on the right hand side is the expectation of a negative binomial distribution with the updated model parameters $r + x$ and $\alpha + T$, given that we also know the updated duration time parameters s and $\beta + T$, and is given by (Schmittlein *et al.*, 1987)

$$\mathbb{E}[X^*|r + x, \alpha + T, s, \beta + T, X = x, t, T, T^*] =$$
$$\frac{(r + x)(\beta + T)}{(\alpha + T)(s - 1)} \left[1 - \left(\frac{\beta + T}{\beta + 2T}\right)^{s-1}\right]. \quad \text{(A.18)}$$

The second term on the right hand side in Equation A.17 is more complex to derive because it depends on the relationship between α and β. Therefore, Schmittlein *et al.* (1987) consider the following three cases:

Case 1: $\alpha > \beta$

$$P[\tau > T|r, s, \alpha > \beta, X = x, t, T] =$$
$$\left\{1 + \frac{s}{r + x + s}\left[\left(\frac{\alpha + T}{\alpha + t}\right)^{r+x}\left(\frac{\beta + T}{\alpha + t}\right)^{s} F(a_1, b_1; c_1; z_1(t)) - \left(\frac{\beta + T}{\alpha + T}\right)^{s} F(a_1, b_1; c_1; z_1(T))\right]\right\}^{-1},$$
$$\text{where} \quad a_1 = r + x + s; \quad b_1 = s + 1; \quad c_1 = r + x + s + 1; \quad z_1(y) = \frac{\alpha - \beta}{\alpha + y}. \quad \text{(A.19)}$$

Case 2: $\alpha < \beta$

$$P[\tau > T | r, s, \alpha < \beta, X = x, t, T] = \left\{ 1 + \frac{s}{r+x+s} \left[\left(\frac{\alpha+T}{\beta+t} \right)^{r+x} \left(\frac{\beta+T}{\beta+t} \right)^{s} F(a_2, b_2; c_2; z_2(t)) - \left(\frac{\alpha+T}{\beta+T} \right)^{r+x} F(a_2, b_2; c_2; z_2(T)) \right] \right\}^{-1},$$

$$\text{where} \quad a_2 = r+x+s; \quad b_2 = r+x; \quad c_2 = r+x+s+1; \quad z_2(y) = \frac{\beta-\alpha}{\beta+y}. \tag{A.20}$$

Case 3: $\alpha = \beta$

$$P[\tau > T | r, s, \alpha = \beta, X = x, t, T] = \left\{ 1 + \frac{s}{r+x+s} \left[\left(\frac{\alpha+T}{\alpha+t} \right)^{r+x+s} - 1 \right] \right\}^{-1}. \tag{A.21}$$

The function $F(a, b; c; z)$ in Equations A.19 and A.20 is the Gauss hypergeometric function, and is given by the following power series for $|z| < 1$:

$$F(a, b; c; z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}, \tag{A.22}$$

where

$$(q)_n = \begin{cases} 1 & \text{if } n = 0 \\ q(q+1)...(q+n-1) & \text{if } n > 0. \end{cases} \tag{A.23}$$

An expression of the expected number of future customer purchases can now be obtained by substituting Equations A.18, A.19, A.20, and A.21 into A.17, along with the estimated model parameters $\hat{r}$, $\hat{\alpha}$, $\hat{s}$, and $\hat{\beta}$.

A.2 Gamma-gamma submodel

The Pareto/NBD model predicts the number of purchases that customers make in a future period. However, our goal is to predict the amount that customers will spend in a future

period. Therefore, Fader *et al.* (2005) created a gamma-gamma submodel that predicts the average amount spent per purchase per customer. Then, one can obtain predictions of future spends by multiplying predictions of the number of purchases with predictions of the average amount spent per purchase. The description of the gamma-gamma submodel follows the description as given in Fader *et al.* (2005).

A.2.1 Model description

The gamma-gamma submodel assumes the following:

1. The monetary value of a customer's purchase varies randomly around the customer's mean purchase value.
2. The customer's mean purchase value does not change over time, and varies across customers.
3. The distribution of mean purchase values across customers and the transaction process are independent of each other.

The observed mean purchase value $\bar{z}$ for each customer is obtained by

$$\bar{z} = \frac{1}{x} \sum_{i=1}^{x} z_i, \tag{A.24}$$

where x is the number of purchases and z_i is the observed monetary value of the i-th purchase. This mean purchase value $\bar{z}$, however, cannot be used as estimate of the unobserved true mean purchase value μ as purchase data tend to be right-skewed: the left tail of the distribution of purchase values is bounded at zero and its right tail can be very long due to several exceptionally high-spending customers. Therefore, we assume that Z_i follows a gamma distribution with shape parameter p and rate parameter ν. Since the model assumes that the mean purchase value varies across customers, a gamma distribution with parameters q and γ is assumed for the rate parameter ν. Furthermore, it is assumed that the shape parameter p does not vary across customers. To recapitulate, we can write $Z_i|\nu \sim \text{Gamma}(p, \nu)$ and $\nu \sim \text{Gamma}(q, \gamma)$.

Utilising the following two relationships involving the gamma distribution,

I the sum of x i.i.d. Gamma(p,ν) random variables follows a gamma distribution with shape parameter px and scale parameter ν, and

II multiplying a Gamma(px,ν) random variable by the scalar $1/x$ is gamma distributed with shape parameter px and scale parameter νx,

it follows that $\bar{Z} \sim \text{Gamma}(px, \nu x)$. Now, we can write the expectation of Z_i as

$$\mathbb{E}[Z_i] = \mathbb{E}\Big[\mathbb{E}[Z_i|\nu]\Big] = \mathbb{E}\left[\frac{p}{\nu}\right] = p * \mathbb{E}\left[\frac{1}{\nu}\right] = \frac{\gamma p}{q-1}, \tag{A.25}$$

where the first equality follows from the law of total expectation, the second equality follows from the fact that $Z_i|\nu \sim \text{Gamma}(p,\, \nu)$ has mean $\frac{p}{\nu}$, and the last equality follows from the fact that $\frac{1}{\nu} \sim \text{Inv-Gamma}(q,\, \gamma)$ has mean $\frac{\gamma}{q-1}$.

A.2.2 Estimation

Estimates of the model parameters p, q, and γ are obtained by maximising the likelihood over all customers, that is, by maximising

$$L(p, q, \gamma|\text{ data}) = \prod_{i=1}^{M} f(\bar{z}_i|p, q, \gamma, x_i), \tag{A.26}$$

where M is the number of customers. The conditional probability density function of $\bar{z}$ follows from the assumed gamma distributions and is given by (Fader *et al.*, 2005)

$$f(\bar{z}|p, q, \gamma, x) = \frac{\Gamma(px+q)}{\Gamma(px)\Gamma(q)} \frac{\gamma^q \bar{z}^{px-1} x^{px}}{(\gamma + \bar{z}x)^{px+q}}, \tag{A.27}$$

where $\Gamma(\cdot)$ is the gamma function as given in A.4.

A.2.3 Forecasting

To obtain an expression for the expected average purchase value of customers, Fader *et al.* (2005) employ Bayes' theorem to derive the posterior distribution of ν for a customer with observed mean purchase value $\bar{z}$ across x purchases:

$$g(\nu|p,q,\gamma,\bar{z},x) = \frac{(\gamma+\bar{z}x)^{px+q}\nu^{px+q-1}e^{-\nu(\gamma+\bar{z}x)}}{\Gamma(px+q)}, \tag{A.28}$$

where $\Gamma(\cdot)$ is the gamma function as given in A.4. This results in a gamma distribution with shape parameter $px+q$ and scale parameter $\gamma+\bar{z}x$. Now the expected mean purchase value Z for a customer with observed mean purchase value $\bar{z}$ can be defined as

$$\mathbb{E}[Z|p,q,\gamma,\bar{z},x] = \frac{(\gamma+\bar{z}x)p}{px+q-1} = \left(\frac{q-1}{px+q-1}\right)\frac{\gamma p}{q-1} + \left(\frac{px}{px+q-1}\right)\bar{z}. \tag{A.29}$$

Note that this is a weighted average of the prior mean purchase rate $\frac{\gamma p}{q-1}$ and the observed mean purchase value $\bar{z}$. Larger values of x (a larger number of observed purchases) will place less weight on the prior mean and more weight on the observed mean purchase value $\bar{z}$.

A.3 Extended Pareto/NBD model by Abe (2009)

The Pareto/NBD model is extended by Abe (2009), who suggests using a hierarchical Bayesian (HB) framework. This framework adjusts the heterogeneity assumptions on the purchasing and death rate of the Pareto/NBD, and relaxes the independence assumption between them. Furthermore, instead of estimating the model parameters analytically, the HB framework uses MCMC simulation to estimate them. By avoiding analytical estimation, the model allows for the incorporation of covariates into the model. The description of this model closely follows the description as given in Abe (2009).

A.3.1 Model description

The first two assumption of the HB framework are equivalent to those of the Pareto/NBD model:

1. While a customer is alive, his number of purchases is characterised by a Poisson distribution with purchasing rate λ.
2. The customer's lifetime is characterised by an exponential distribution with death rate μ.

However, the assumptions on independently gamma-distributed purchasing and death rates of the Pareto/NBD model are replaced by

3. Heterogeneity in the purchasing rate λ and the death rate μ across customers is captured by a multivariate log-normal distribution.

Unlike the Pareto/NBD model, whereby independent distributions are assumed for λ and μ, this assumption permits correlation between the purchasing and death rate. Abe (2009) gives two reasons for assuming log-normally distributed purchasing and death rates. The first reason is that Bayesian updating of log-normal distributions is a standard procedure and simple to compute. The second reason is that correlation between $\log(\lambda)$ and $\log(\mu)$ can be retrieved using the variance-covariance matrix of the normal mixture distribution. This correlation is hard to compute when using correlated gamma distributions. The third assumption can be mathematically expressed as

$$\begin{bmatrix} \log(\lambda) \\ \log(\mu) \end{bmatrix} \sim MVN(\boldsymbol{\theta}, \boldsymbol{\Gamma}_0), \tag{A.30}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\Gamma}_0$ are the mean and covariance matrix of the multivariate normal distribution, respectively. Note that mathematical expressions of the first two assumptions are given in Equations A.1 and A.2.

In contrast to the Pareto/NBD model, the HB framework allows for the incorporation of time-invariant covariates. These covariates might provide important information that

can boost the predictive performance of the model, and give insight into the profiles of high-spending and low-spending customers. The covariates are included in the model by specifying the logarithms of λ_i and μ_i with a linear regression as follows, where the index i emphasises that the purchasing and death rate correspond to customer i:

$$\begin{bmatrix} \log(\lambda_i) \\ \log(\mu_i) \end{bmatrix} \equiv \theta_i = \boldsymbol{\beta}' \mathbf{d}_i + \mathbf{e}_i, \text{ with } \mathbf{e}_i \sim MVN(\mathbf{0}, \boldsymbol{\Gamma}_0), \tag{A.31}$$

where $\mathbf{d}_i$ is a $K \times 1$ vector containing K characteristics of customer i, $\boldsymbol{\beta}$ is a $K \times 2$ parameter matrix, and $\mathbf{e}_i$ is a 2×1 vector of error terms. When $\mathbf{d}_i$ contains only an intercept, this model reduces to the model as specified in Equation A.30.

A.3.2 Estimation

As opposed to the Pareto/NBD model, the parameter estimates of λ and μ cannot be computed individually in the Bayesian framework. Therefore, in order to estimate the model parameters, Abe (2009) introduces two latent variables w and y. The latent variable w equals 1 if the customer is active at time T, and 0 otherwise, and the latent variable y equals the time of death when $w = 0$. If these latent variables are observed, the likelihood function for the recency-frequency data (x, t, T) in the case where $w = 1$, conditional on the values of λ and μ, becomes

$$P[X = x, t, T|\lambda, \mu, w = 1]\ P[w = 1|\lambda, \mu] = \frac{\lambda^x t_x^{x-1}}{\Gamma(x)} e^{-(\lambda+\mu)T}, \tag{A.32}$$

which follows from Equations A.11 and A.14. In case $w = 0$, the likelihood function for the recency-frequency data (x, t, T), conditional on the values of λ and μ, becomes

$$P[X = x, t, T|\lambda, \mu, y, w = 0]\ f_\tau(y|\lambda, \mu) = \frac{\lambda^x t_x^{x-1}}{\Gamma(x)} \mu e^{-(\lambda+\mu)y}, \tag{A.33}$$

where the derivation is similar to Equation A.15. These two cases can be combined into a more compact notation, and the conditional likelihood then becomes

$$L(X = x, t, T|\lambda, \mu, w, y) = \frac{\lambda^x t_x^{x-1}}{\Gamma(x)} \mu^{1-w} e^{-(\lambda+\mu)(wT+(1-w)y)}. \tag{A.34}$$

In addition, the probability of a customer being alive at time T (i.e. $w = 1$) can be written as (Schmittlein and Peterson, 1994):

$$P[w = 1|\lambda, \mu, X = x, t, T] = P[\tau > T|\lambda, \mu, X = x, t, T] = \frac{1}{1 + \frac{\mu}{\lambda + \mu}[e^{(\lambda+\mu)(T-t)} - 1]}, \quad \text{(A.35)}$$

where we use the fact that $w = 1$ if and only if $\tau > T$.

However, since neither w nor y is observed, they are treated as latent variables in the model and are sampled alongside the model parameters in the Bayesian updating approach. Bayesian updating requires the specification of prior distributions for the model parameters. The log-normal prior distribution for θ_i is already specified in Equation A.31. The parameters of this log-normal distribution, $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}_0$, are assumed to to follow a multivariate normal and an inverse Wishart distribution, respectively:

$$\boldsymbol{\beta} \sim MVN(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0), \quad \text{(A.36)}$$

$$\boldsymbol{\Gamma}_0 \sim IW(\nu_{00}, \boldsymbol{\Gamma}_{00}), \quad \text{(A.37)}$$

which are standard priors for the (log-)normal model. Finally, one assumes diffuse priors for the constants $\boldsymbol{\beta}_0$, $\boldsymbol{\Sigma}_0$, ν_{00} and $\boldsymbol{\Gamma}_{00}$.

The model parameters $\{\boldsymbol{\theta}_i, y_i, w_i \ \forall i; \boldsymbol{\beta}, \boldsymbol{\Gamma}_0\}$ can now be estimated by Markov Chain Monte Carlo (MCMC) simulation. The joint density is estimated by sequentially generating each parameter, conditional on the previously generated parameters, from its conditional distribution until the Markov chain has converged. A description of this procedure is given below (Korkmaz *et al.*, 2013):

Step 0: Set an initial value for $\boldsymbol{\theta}_i^{(0)} \ \forall i$.

Step 1: (a) Generate $w_i|\boldsymbol{\theta}_i \ \forall i$ according to equation A.35. Note that $\boldsymbol{\theta}_i$ from the previous iteration must be exponentiated to transform to λ_i and μ_i.

(b) If $w_i = 0$ (i.e. the customer is dead), generate lifetime $y_i|w_i, \boldsymbol{\theta}_i$ using a truncated exponential distribution. From Equation A.34 it follows that

$y_i|w_i, \boldsymbol{\theta}_i$ follows an exponential distribution with rate $\lambda_i + \mu_i$ and truncation such that $t_i < y_i < T_i$.

Step 2: Generate $\boldsymbol{\beta}, \boldsymbol{\Gamma}_0|\boldsymbol{\theta}_i \ \forall i$ using a standard multivariate normal regression update (Rossi *et al.*, 2005).

Step 3: Generate $\boldsymbol{\theta}_i|\boldsymbol{\beta}, \boldsymbol{\Gamma}_0, w_i, y_i$ using Equation A.34. After multiplying Equation A.34 with the log-normal prior, λ_i and μ_i are sequentially generated by an independent Metropolis-Hastings algorithm with a log-normal proposal distribution. Finally, λ_i and μ_i are logarithmically transformed to $\boldsymbol{\theta}_i$.

Step 4: Iterate steps 1 to 3 until convergence is achieved.

A.3.3 Forecasting

Once the Markov chain has converged, simulation draws $(\lambda_i^{(s)}, \mu_i^{(s)}, y_i^{(s)}, w_i^{(s)})$ can be obtained for each customer i. These draws can then be used to compute the number of purchases in future time period $(T, T^*]$. First of all, one draws a sample of the remaining lifetime of a customer from Equation A.2 with parameter $\mu_i^{(s)}$. Next, a sample of the number of future purchases is drawn from Equation A.1 with parameter $\lambda_i^{(s)}$. These two steps are repeated S times for each customer. Note that for $w_i^{(s)} = 0$, the number of future purchases will be zero. Thereafter, these S draws are aggregated for each customer by taking the mean of the S draws of the number of future purchases. Finally, forecasts of the amount spent per customer are obtained by combining the model with the gamma-gamma submodel, similarly to the standard Pareto/NBD model.

Appendix B: Duration Models

Duration models are concerned with the study of survival times and the factors that influence them. The study of survival times is also known as survival analysis. Usually, duration models are applied on data that consist of individuals that were followed over a specified time period. The focus lies on the time at which a specific event occurs, which is referred to as the failure time. Examples of events are deaths, hospital discharges, machinery breakdowns, and purchases. The time from the beginning of an observation period and the occurrence of an event is called the survival time. The objectives of duration models include predicting the time at which a yet unobserved event occurs for a particular subject, analysing event time patterns, comparing survival time distributions in different groups, and examining what factors influence the occurrence time of events.

B.1 Censoring

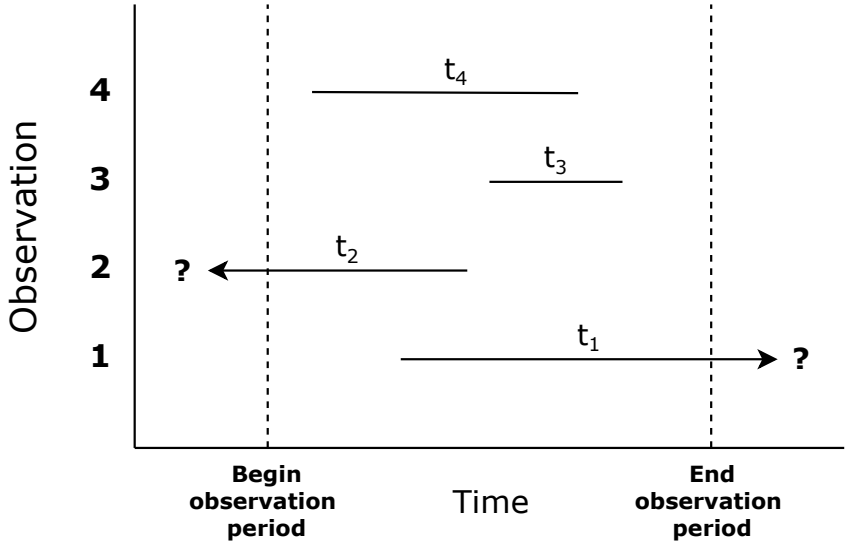

Figure B.1: Different types of observations. Observation 1 is right-censored, observation 2 is left-censored, and observations 3 and 4 are complete.

A key characteristic of survival data is that the survival time of individuals may only be partially observed, which is referred to as censored data. The most commonly encountered type of censoring is right censoring, where the occurrence of the event of interest is known to have occurred after a certain time. In this case, only the time span at which the

event did not occur is observed. Right censoring may occur when an individual does not experience the event for the duration of the study, or when an individual drops out of the study. Left censoring is another type of censoring, where the event of interest is known to have occurred before a certain time. In this case, the event of interest has occurred before the start of the observation period. For example, when modelling pregnancy duration, starting the observation period at the 250-day mark may result in women that already had their babies. Figure B.1 shows both types of censoring graphically, along with complete observations.

There are several mechanisms that can lead to censored data, and are categorised as Type I, Type II, or random censoring. In Type I censoring, the study stops at a pre-specified time by the researcher, and individuals who have not experienced the event before the end of the study are censored. In Type II censoring, the study stops when a pre-specified number of events have occurred, and individuals who have not experienced the event before the end of the study are censored. In random censoring, individuals have a censoring time that is statistically independent of their failure time. Careful attention to this type of censoring is essential because dependent censoring times causes biased survival estimates. For example, if patients that are close to dying are more likely to drop out of the study than other patients, the survival estimates will be negatively affected.

B.2 Basic principles

Let $T \geqslant 0$ be a random variable representing the waiting time until the occurrence of an event. The probability density function of T is denoted by $f(t)$, and the cumulative density function is given by

$$F(t) = P[T < t] = \int_0^t f(u)\, du, \tag{B.1}$$

which can be interpreted as the probability that the event occurs before time t. One usually considers the survival function, which is the complement of $F(t)$, and is given by

$$S(t) = 1 - F(t) = P[T \geqslant t] = \int_t^\infty f(u)\, du, \tag{B.2}$$

which can be interpreted as the probability that the event occurs at time t or later. A different specification of the distribution of T is given by the hazard function and is defined as

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P[t \leqslant T < t + \Delta t | T \geqslant t]}{\Delta t}, \tag{B.3}$$

which can be interpreted as the instantaneous failure rate at time t. The hazard function can be rewritten as

$$\begin{aligned} \lambda(t) &= \lim_{\Delta t \to 0} \frac{P[t \leqslant T < t + \Delta t | T \geqslant t]}{\Delta t} \\ &= \frac{1}{P[T \geqslant t]} \lim_{\Delta t \to 0} \frac{P[t \leqslant T < t + \Delta t]}{\Delta t} \\ &= \frac{1}{S(t)} \lim_{\Delta t \to 0} \frac{P[t \leqslant T < t + \Delta t]}{\Delta t} \\ &= \frac{1}{S(t)} \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \\ &= \frac{f(t)}{S(t)}, \end{aligned} \tag{B.4}$$

which is a more useful result. In the first step, we use the definition of conditional probability $P(A|B) = \frac{P(A \cap B)}{P(B)}$ and the fact that $P[t \leqslant T < t + \Delta t \ \cap \ T \geqslant t] = P[t \leqslant T < t + \Delta t]$. In the second step, we use Equation B.2, and in the fourth step we use the definition of the derivative. This formula can be further rewritten to

$$\begin{aligned} \lambda(t) &= \frac{f(t)}{S(t)} \\ &= \frac{\frac{\partial}{\partial t} F(t)}{S(t)} \\ &= \frac{\frac{\partial}{\partial t}(1 - S(t))}{S(t)} \\ &= -\frac{\frac{\partial}{\partial t} S(t)}{S(t)} \\ &= -\frac{\partial}{\partial t} \ln S(t). \end{aligned} \tag{B.5}$$

To find the inverse relation, we first define the cumulative hazard function (c.h.f.) as

$$\Lambda(t) = \int_0^t \lambda(u)\, du, \tag{B.6}$$

which can be used to obtain a formula for the probability of surviving to time t as a function of the hazard at all times up to t (using Equation B.5):

$$S(t) = \exp(-\Lambda(t)). \tag{B.7}$$

B.3 Multiplicative intensity model

The multiplicative intensity model is a statistical model for processes that are observed on a fixed time interval. Typically, individuals under observation are not observed over the whole study time period, and to accommodate for this, it is assumed that the intensity process takes a certain form. All methods discussed in Appendix B follow the multiplicative intensity model.

Suppose there are n individuals of which a counting process registering the number of occurrences of an event of interest is known. Let $N_i(t)$ be the observed number of events for individual i in the time interval $[0, t]$. Furthermore, let $Y_i(t)$ be an indicator whether individual i is at risk 'just before' time t. More formally,

$$Y_i(t) = I\{T_i \geqslant t\}. \tag{B.8}$$

Then, the intensity process of $N_i(t)$ is assumed to take the form

$$h_i(t) = \alpha(t)Y_i(t), \tag{B.9}$$

where $\alpha(t)$ is a non-negative function called the intensity rate. Aggregating the individual counting processes, that is, considering the process $N(t) = \sum_{i=1}^{n} N_i(t)$ counting the total number of observed events, results in the aggregated intensity process

$$h(t) = \sum_{i=1}^{n} \lambda_i(t) = \alpha(t)Y(t), \tag{B.10}$$

where $Y(t) = \sum_{i=1}^{n} Y_i(t)$, the total number of individuals at risk 'just before' time t. In the special case of survival data, $\alpha(t)$ is the hazard function.

In case that the intensity process may be specified by a q-dimensional parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)'$, it takes the form

$$h(t; \boldsymbol{\theta}) = \alpha(t; \boldsymbol{\theta}) Y(t). \tag{B.11}$$

B.4 Non-parametric estimation

Non-parametric estimation, as opposed to parametric estimation, does not require the researcher to choose a distribution that approximates the hazard and survival function. When modelling human survival, distributions may not have sufficient flexibility to represent the actual shape of the hazard and survival function, and therefore non-parametric estimation may be more suitable. However, non-parametric estimation does not allow for the incorporation of covariates into the model.

Suppose there are n observations, and that no observation ties exist. Sort all observations by their duration such that

$$t_1 < t_2 < t_3 < \ldots < t_n. \tag{B.12}$$

In case there exist observation ties, the total number of recorded survival times is smaller than n. Let d_i be the number of events that occur at time t_i ($d_i = 1$ if there are no ties at time t_i). A natural, non-parametric estimator of the hazard function if given by

$$\hat{P}[T_i = t_i | T_i \geqslant t_i] = \hat{\lambda}_i = \frac{d_i}{Y(t_i)}. \tag{B.13}$$

Then, the Kaplan-Meier estimator for the probability of survival at time t is the product over the failure times of the conditional probabilities of surviving to the next failure time, and is formally given by

$$\hat{S}(t) = \prod_{i|t_i \leqslant t} (1 - \hat{\lambda}_i) = \prod_{i|t_i \leqslant t} \left\{ 1 - \frac{d_i}{Y(t_i)} \right\}. \tag{B.14}$$

Although censoring is not particularly specified in this equation, its presence does not affect the validity of this formulation. Standard errors can be computed using

Greenwood's formula given by

$$\widehat{Var}(\hat{S}(t)) = \hat{S}(t)^2 \sum_{i|t_i \leqslant t} \frac{d_i}{Y(t_i)\left\{Y(t_i) - d_i\right\}}. \tag{B.15}$$

The Nelson-Aalen estimator can be used to estimate the c.h.f., and is given by

$$\hat{\Lambda}(t) = \sum_{i|t_i \leqslant t} \hat{\lambda}_i = \sum_{i|t_i \leqslant t} \frac{d_i}{Y(t_i)}, \tag{B.16}$$

and its variance may be estimated by

$$\widehat{Var}(\hat{\Lambda}(t)) = \sum_{i|t_i \leqslant t} \frac{(Y(t_i) - d_i)\, d_i}{Y(t_i)^3}. \tag{B.17}$$

B.5 Parametric estimation

An alternative way of estimation in duration models is the use of parametric models. Parametric models make assumptions about the patterns of survival times, which can be represented by probability distributions. The hazard and survival function are assumed to have a specific type of shape, and the exact shape is determined by parameters that are estimated from the data. Several distributions have been proposed in the literature, and the most common ones are the exponential, Weibull, Gamma, Log-normal, Log-logistic, and Gompertz distribution. In addition, parametric estimation allow for the incorporation of covariates into the model.

B.5.1 Maximum Likelihood

A common estimation method for duration analysis in maximum likelihood estimation. Suppose there are n observations with lifetimes governed by a survival function $S(t;\boldsymbol{\theta})$ with associated density $f(t;\boldsymbol{\theta})$ and hazard function $\lambda(t;\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)'$ is a q-dimensional parameter vector. Furthermore, let $c_1, c_2, \ldots, c_n$ be given censoring times. Then, for each individual, we do not necessarily observe the survival time t_i itself, but only the censored survival time $\tilde{t_i} = \min(t_i, c_i)$ along with a death indicator $d_i = I\{\tilde{t_i} = t_i\}$

which equals 1 if the actual survival time is observed, and 0 if the censored time is observed. If $d_i = 1$, the ith individual contributes $f(\tilde{t}_i; \boldsymbol{\theta})$ to the likelihood, and if $d_i = 0$, the ith individual contributes $S(\tilde{t}_i; \boldsymbol{\theta})$ to the likelihood. Both contributions can be written in a single expression, and the likelihood contribution of individual i may be written as

$$\begin{aligned} L_i(\boldsymbol{\theta}) &= f(\tilde{t}_i; \boldsymbol{\theta})^{d_i} \, S(\tilde{t}_i; \boldsymbol{\theta})^{1-d_i} \\ &= \lambda(\tilde{t}_i; \boldsymbol{\theta})^{d_i} \, S(\tilde{t}_i; \boldsymbol{\theta}), \end{aligned} \tag{B.18}$$

where Equation B.4 is used in the second step. Because the likelihood contributions from all individuals are independent, the full likelihood can be written as

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{i=1}^{n} L_i(\boldsymbol{\theta}) \\ &= \prod_{i=1}^{n} f(\tilde{t}_i; \boldsymbol{\theta})^{d_i} \, S(\tilde{t}_i; \boldsymbol{\theta})^{1-d_i} \\ &= \prod_{i=1}^{n} \lambda(\tilde{t}_i; \boldsymbol{\theta})^{d_i} \, S(\tilde{t}_i; \boldsymbol{\theta}). \end{aligned} \tag{B.19}$$

Taking logarithms and using Equation B.7, we obtain the log-likelihood function

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n} \{d_i \log \lambda(\tilde{t}_i; \boldsymbol{\theta}) - \Lambda(\tilde{t}_i; \boldsymbol{\theta})\}, \tag{B.20}$$

which is maximised with respect to $\boldsymbol{\theta}$ to obtain parameter estimates $\hat{\boldsymbol{\theta}}$.

B.5.2 Accelerated Lifetime Model

Until now we have been concerned with individuals' lifetimes that all follow the same survival function $S(t)$. However, these lifetimes may be affected by explanatory variables, and therefore the inclusion of these variables may be beneficiary to survival models. The accelerated lifetime model (ALM) is a survival model that incorporates explanatory variables. It assumes that the survival function has the same shape for all individuals, and that explanatory variables affect survival time by altering the speed at which individuals move along the curve. Therefore, it scales the time by a function of explanatory variables in the survival function.

Let $S_0(t_i)$ be the baseline survival function of individual i, that is, the survival function of an individual with his explanatory variable(s) taking the value zero. This corresponds to non-parametric estimation of the survival function, as described in Section B.4. Then, time is rescaled by $\exp(\boldsymbol{x}_i'\boldsymbol{\beta})$, and the survival function for individual i becomes

$$S(t_i; \boldsymbol{x}_i) = S_0(\exp(\boldsymbol{x}_i'\boldsymbol{\beta})t_i), \tag{B.21}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iq})'$ is a q-dimensional vector of covariates of individual i with coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_q)'$. The factor $\exp(\boldsymbol{\beta})$ is called the acceleration factor, and represents the speed at which an individual moves along the survival curve. If the factor is larger than one, then individuals with higher values of $\boldsymbol{x}_i$ will tend to have earlier event times. Vice versa, if the factor is smaller than one, then individuals with higher values of $\boldsymbol{x}_i$ will tend to have later event times.

Additionally, using Equation B.2 and the fact that $\frac{\partial}{\partial t}F(t) = f(t)$, the density function of the lifetime of individual i becomes

$$\begin{aligned} f(t_i; \boldsymbol{x}_i) &= \frac{\partial}{\partial t}(1 - S(t_i; \boldsymbol{x}_i)) \\ &= \exp(\boldsymbol{x}_i'\boldsymbol{\beta}) f_0(\exp(\boldsymbol{x}_i'\boldsymbol{\beta})t_i), \end{aligned} \tag{B.22}$$

where $f_0(\cdot)$ is the baseline density. Furthermore, using Equations B.4 and B.22, the hazard function of individual i becomes

$$\lambda(t_i; \boldsymbol{x}_i) = \exp(\boldsymbol{x}_i'\boldsymbol{\beta})\lambda_0(\exp(\boldsymbol{x}_i'\boldsymbol{\beta})t_i), \tag{B.23}$$

where $\lambda_0(\cdot)$ is the baseline hazard. Here, the baseline density and baseline hazard are non-parametric estimates of the density and hazard function, as described in Section B.4. Since ALM contains both a parametric and a non-parametric part, it is said to be semi-parametric. Note that in the absence of covariates, the ALM's semi-parametric estimation reduces to ordinary non-parametric estimation.

B.5.3 Proportional Hazard Model

Another duration model that allows for the incorporation of explanatory variables is the proportional hazard model (PHM). The description of the proportional hazard model in this subsection closely follows the description as given in Aalen *et al.* (2008). The PHM assumes that explanatory variables are multiplicatively related to the hazard function, or, in other words, explanatory variables are assumed to change the chance of failure. Note that these variables do not change the time at which the hazard is high or low. Formally, it is assumed that the hazard function of individual i takes the form

$$\lambda(t_i|\boldsymbol{x}_i) = r(\boldsymbol{x}_i(t), \boldsymbol{\beta})\lambda_0(t_i), \tag{B.24}$$

where $\boldsymbol{x}_i = (x_{i1}(t), x_{i2}(t), \ldots, x_{iq}(t))'$ is a q-dimensional vector of covariates of individual i that can be both fixed or time-varying, $r(\boldsymbol{x}_i(t), \boldsymbol{\beta})$ is a relative risk function that describes how the size of the hazard function depends on explanatory variables, and $\lambda_0(t_i)$ is the baseline hazard that describes the shape of the hazard function over time. Note that this model contains both a non-parametric part (the baseline hazard) and a parametric part (the relative risk function), and is therefore said to be semi-parametric. Note that in the absence of covariates, the PHM's semi-parametric estimation reduces to non-parametric estimation (Rodrıguez, 2005). In the literature, the relative risk functions has been specified in multiple ways. For example, the exponential relative risk function takes the form $r(\boldsymbol{x}_i(t), \boldsymbol{\beta}) = \exp(\boldsymbol{x}_i(t)'\boldsymbol{\beta})$, the linear relative risk function takes the form $r(\boldsymbol{x}_i(t), \boldsymbol{\beta}) = 1 + \boldsymbol{x}_i(t)'\boldsymbol{\beta}$, and the excess relative risk model takes the form $r(\boldsymbol{x}_i(t), \boldsymbol{\beta}) = \prod_{j=1}^{p} 1 + \beta_j x_{ij}(t)$. The use of the exponential relative risk function results in the well-known Cox regression model. In the Cox regression model, $\frac{\partial\, ln\lambda(t_i|\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i} = \beta$, and the coefficient can be interpreted as the constant proportional effect of the explanatory variables on the hazard function.

Using Equations B.6, B.7, and B.24, the survival function of individual i can be written as

$$\begin{aligned} S(t_i|\boldsymbol{x}_i) &= \exp(-\Lambda(t_i)) \\ &= \exp(-r(\boldsymbol{x}_i(t), \boldsymbol{\beta})\Lambda_0(t_i)). \end{aligned} \tag{B.25}$$

Furthermore, using Equations B.4, B.24, and B.25, the density function of individual i

can be written as

$$\begin{aligned} f(t_i|\boldsymbol{x}_i) &= \lambda(t_i|\boldsymbol{x}_i)S(t_i|\boldsymbol{x}_i) \\ &= r(\boldsymbol{x}_i(t), \boldsymbol{\beta})\lambda_0(t_i)\exp(-r(\boldsymbol{x}_i(t), \boldsymbol{\beta})\Lambda_0(t_i)). \end{aligned} \tag{B.26}$$

The semi-parametric nature of the model does not allow for the use of ordinary likelihood methods to estimate the regression coefficients. Therefore, one has to resort to a partial likelihood. First note that the intensity process of $N_i(t)$ may be written as

$$h_i(t) = Y_i(t)\,\alpha(t|\boldsymbol{x}_i(t)). \tag{B.27}$$

Using Equation B.24, the intensity process of $N_i(t)$ can be rewritten as

$$h_i(t) = Y_i(t)\,\alpha_0(t)\,r(\boldsymbol{x}_i(t), \boldsymbol{\beta}). \tag{B.28}$$

Next, the intensity process of the aggregated counting process $N_{\blacksquare}(t)$ can be written as

$$\lambda_{\blacksquare}(t) = \sum_{l=1}^{n} \lambda_l(t) = \sum_{l=1}^{n} Y_l(t)\,\alpha_0(t)\,r(\boldsymbol{x}_l(t), \boldsymbol{\beta}). \tag{B.29}$$

The intensity process of $N_i(t)$ may be factorized as $\lambda_i(t) = \lambda_{\blacksquare}(t)\pi(i|t)$, where

$$\pi(i|t) = \frac{\lambda_i(t)}{\lambda_{\blacksquare}(t)} = \frac{Y_i(t)r(\boldsymbol{x}_i(t), \boldsymbol{\beta})}{\sum_{l=1}^{n} Y_l(t)r(\boldsymbol{x}_l(t), \boldsymbol{\beta})}, \tag{B.30}$$

which can be interpreted as the conditional probability that an event occurs for individual i at time t. To obtain the partial likelihood for $\boldsymbol{\beta}$, these conditional probabilities are multiplied over all event times. Assume that there are no tied event times and let the n observations be sorted by their duration such that

$$t_1 < t_2 < \ldots < t_n. \tag{B.31}$$

The partial likelihood then becomes

$$L(\boldsymbol{\beta}) = \prod_{t_j} \pi(i_j|t_j) = \prod_{t_j} \frac{Y_{i_j}(t_j)r(\boldsymbol{x}_{i_j}(t_j), \boldsymbol{\beta})}{\sum_{l=1}^{n} Y_l(t_j)r(\boldsymbol{x}_l(t_j), \boldsymbol{\beta})}, \tag{B.32}$$

where i_j is the index of the individual who experiences an event at T_j. Furthermore, let the risk set $R_j = \{l|Y_l(t_j) = 1\}$ be the set of individuals who are at risk 'just before' time t_j. Then, the partial likelihood can be rewritten as

$$L(\boldsymbol{\beta}) = \prod_{t_j} \frac{r(\boldsymbol{x}_{i_j}(t_j), \boldsymbol{\beta})}{\sum_{l \in R_j} r(\boldsymbol{x}_l(t_j), \boldsymbol{\beta})}. \tag{B.33}$$

Note that the likelihood does not include the hazard function, which implies that it is not necessary to specify the hazard function in order to estimate $\boldsymbol{\beta}$.

In case there exist tied data an adjustment to the likelihood function is needed. Sort the durations such that

$$t_1 < t_2 < \ldots < t_k, \tag{B.34}$$

with $k \leqslant n$. Let d_j be the number of events occurring at time t_j, and let $D(t_j)$ denote the set of individuals that died at time t_j. The partial likelihood then becomes

$$L(\boldsymbol{\beta}) = \prod_{j=1}^{k} \frac{\prod_{m \in D(t_j)} r(\boldsymbol{x}_m(t_j), \boldsymbol{\beta})}{\{\sum_{l \in R(t_j)} r(\boldsymbol{x}_l(t_j), \boldsymbol{\beta})\}^{d_j}}. \tag{B.35}$$

B.6 Forecasting

In the case of non-parametric estimation, the probability that the event for individual i ends within the next Δt period, given that it has not ended at time t_i, is given by

$$\begin{aligned} P[T_i \leqslant t_i + \Delta t | T_i > t_i] &= 1 - P[T_i > t_i + \Delta t_i | T_i > t] \\ &= 1 - \frac{P[T_i > t_i + \Delta t_i]}{P[T_i > t_i]} \\ &= 1 - \frac{S(t_i + \Delta t_i)}{S(t_i)}. \end{aligned} \tag{B.36}$$

Furthermore, this probability is derived similarly as in Equation B.36 for the parametric models, the difference being that one conditions on the fact that the explanatory variables of the corresponding individual are known. This probability then becomes

$$P[T_i \leqslant t_i + \Delta t | T_i > t_i, \boldsymbol{x}_i] = 1 - \frac{S(t_i + \Delta t_i | \boldsymbol{x}_i)}{S(t_i | \boldsymbol{x}_i)}. \tag{B.37}$$

Appendix C: Gradient Tree Boosting

This section provides a detailed description of the technique called gradient tree boosting. Gradient tree boosting (Friedman, 2002) is a machine learning technique for both classification and regression problems that uses an ensemble of 'weak' decision trees to obtain a 'strong' predictor. Compared to a single decision tree, gradient tree boosting outputs predictions with both lower bias and variance. To do this, it uses a forward stage-wise modelling approach that allows for the optimisation of a differentiable loss function that measures how well the model performs. The high performance of gradient tree boosting has made this a popular technique among data scientists. It has been used consistently to win machine learning competitions on Kaggle, a platform for predictive modelling and analytics competitions.

Before one can understand the concept of gradient tree boosting, one has to be acquainted with both decision trees and gradient boosting. Moreover, to be able to understand gradient boosting, one has to be acquainted with basic ensemble techniques such as bagging and generic boosting. Therefore, this section continues with a detailed description of decision trees, whereafter several ensemble techniques are described including gradient boosting.

C.1 Decision tree

Decision tree learning is a supervised machine learning technique that predicts a response variable based on several input variables by using a tree-like structure. A decision tree groups data instances by posing a series of questions about the features associated with the instances. It starts using the whole set of data instances and splits it into multiple subsets for each possible answer to its question. Each of these subsets is then split into smaller subsets in a similar fashion. This process is repeated until a stopping criterion is met. In each of the final subsets, a constant is predicted for the response variable. Once the tree is grown, new data instances follow the path from the whole set of data instances to one of the final subsets of data instances, and the value that was assigned to the corresponding subset is predicted for the response variable.

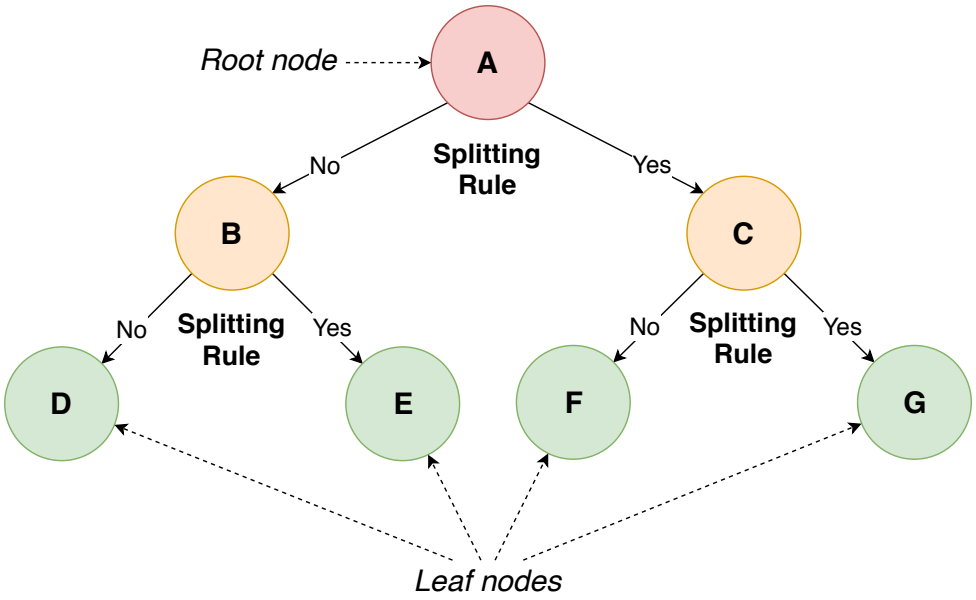

Figure C.1: Graphical representation of a binary tree with four terminal nodes.

A tree consists of nodes that represent the subsets of data instances. These nodes are split into sub-nodes according to a certain splitting rule. Nodes that do not split are called leaf nodes or terminal nodes, and a single constant is predicted for the response variable of all data instances that fall into these nodes. Nodes are usually only allowed to have exactly two sub-nodes and in this special case the tree is called a binary tree. Figure C.1 shows an example of a binary tree. The first node (node A) contains the set of all data instances and is called the root node. This set of data instances is split into two subsets (node B and C) according to some splitting rule. These two nodes are then also split into two subsets (nodes D, E, F, and G) according to their own splitting rules. The final four nodes are not split any further and are therefore leaf nodes.

There are two types of decision trees, each depending on the type of response variable that is used. Classification trees deal with discrete response variables, whereas regression trees deal with continuous response variables. These two types of trees differ in how the best splitting variable and split point is chosen. Since the response variable in this study is continuous, I will only describe regression trees in the remainder of this subsection. This description closely follows the description as given in Hastie *et al.* (2009).

Growing a tree involves deciding on what feature and condition to use at each split. Given a learning set $L = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$ with continuous y, the objective of tree learning is to find

a function $F^*(\boldsymbol{x})$ that maps the explanatory variables $\boldsymbol{x}$ to the response variable y, such that the expected value of a certain loss function $\Psi(y, F(\boldsymbol{x}))$ is minimised

$$F^*(\boldsymbol{x}) = \underset{F(\boldsymbol{x})}{\arg\min}\, E_{y,\boldsymbol{x}}[\Psi(y, F(\boldsymbol{x}))]. \tag{C.1}$$

A common choice for the loss function is least-squares: $\Psi(y, F) = (y - F)^2$. Now suppose that the data are partitioned into M regions $R_1, R_2, ..., R_M$, and that a constant c_m is predicted in each region. An approximation of $F^*(\boldsymbol{x})$ will then be

$$F(\boldsymbol{x}) = \sum_{m=1}^{M} c_m I(\boldsymbol{x} \in R_m). \tag{C.2}$$

Note that for a least-squares loss function, the optimal c_m is the average of y_i in region R_m:

$$\hat{c}_m = ave(y_i | \boldsymbol{x}_i \in R_m). \tag{C.3}$$

Finding the optimal binary partition in terms of $\Psi(y, F(\boldsymbol{x}))$ is generally computationally infeasible, and therefore a greedy algorithm is used. Starting with the complete set of data instances, consider a splitting variable j and split point s that split the set of data instances into regions

$$R_1(j, s) = \{\boldsymbol{x} | \boldsymbol{x}_j \leqslant s\} \quad \text{and} \quad R_2(j, s) = \{\boldsymbol{x} | \boldsymbol{x}_j > s\}. \tag{C.4}$$

The optimal value of j and s are then found by solving

$$\min_{j,s} \left[\min_{c_1} \sum_{i | \boldsymbol{x}_i \in R_1(j,s)} \Psi(y_i, c_1) + \min_{c_2} \sum_{i | \boldsymbol{x}_i \in R_2(j,s)} \Psi(y_i, c_2) \right]. \tag{C.5}$$

Note that for a least-squares loss function, the inner minimisation problems are solved by

$$\hat{c}_1 = ave(y_i | \boldsymbol{x}_i \in R_1(j, s)) \quad \text{and} \quad \hat{c}_2 = ave(y_i | \boldsymbol{x}_i \in R_2(j, s)). \tag{C.6}$$

In words, the algorithm computes the loss for every possible pair (j, s) and chooses the pair that results in the lowest loss, assuming that the accumulation of locally best results delivers the global best partition of the feature space. Since determining the split point s for splitting variable j can be done very quickly, scanning through all (j, s) pairs is

computationally feasible. After the best split is found, the data are partitioned into the two resulting regions, and the splitting process is repeated for these new regions.

In addition to finding the best splitting variable and split point at each split, growing a tree also involves knowing when to stop splitting. Large trees tend to overfit the data, leading to unreliable estimates for new data instances, whereas small trees might underfit the data, i.e. not capturing the underlying structure of the data. Reducing the size of a tree is called pruning and there are several ways to do this.

A very basic pruning method is to stop splitting when the number of data instances in a node is below a certain threshold. Another basic method is to stop splitting if the decrease in loss is smaller than a certain threshold. These methods are too short-sighted, however, as seemingly worthless splits may lead to good splits later on. Therefore, one usually grows a tree as large as possible and then prunes the tree using cost-complexity pruning. Cost-complexity pruning leverages the out-of-sample improvement of adding branches with the added complexity. Let a subtree $T \subset T_0$ be any tree that can be obtained by pruning T_0, that is, by collapsing any number of its non-terminal nodes. Furthermore, let $|T|$ be the number of terminal nodes in T. Now define the following variables:

$$N_m = \#\{\boldsymbol{x}_i \in R_m\}, \tag{C.7}$$

$$\hat{c}_m = \min_{c_m} \sum_{i|\boldsymbol{x}_i \in R_m} \Psi(y_i, c_m), \tag{C.8}$$

$$Q_m(T) = \frac{1}{N_m} \sum_{i|\boldsymbol{x}_i \in R_m} \Psi(y_i, \hat{c}_m). \tag{C.9}$$

The cost-complexity criterion can then be defined as

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|, \tag{C.10}$$

which is the sum of the prediction errors in each terminal node along with an extra term that penalises the number of terminal nodes in the tree. The goal is to find the subtree $T_\alpha \subseteq T_0$ that minimises $C_\alpha(T)$, for each value of α. Note that there are infinite values of α but that one only considers values that change the number of nodes in the tree. One then chooses $T_{\hat{\alpha}}$ as final tree, where the value $\hat{\alpha}$ results in the lowest out-of-sample

loss, which is usually computed using 5-fold or 10-fold cross-validation. The penalty term $\alpha \geqslant 0$ controls the trade-off between the tree complexity and the prediction error. Larger values of α will result in smaller trees and smaller values of α will result in larger trees. If $\alpha = 0$, T_α will be a full grown tree, and if $\alpha \to \infty$, T_α will be a tree consisting of a single node.

One can prove that there exists a unique smallest subtree T_α that minimises $C_\alpha(T)$ for a given α. Let the internal nodes that produce the smallest increase in prediction error ($\sum_m N_m Q_m(T)$) be collapsed successively until one ends up with only the root node. This results in a finite sequence of subtrees, and Breiman *et al.* (1984) and Ripley (2007) show that this sequence must include T_α.

C.2 Ensemble learning

Ensemble learning is a machine learning concept in which the idea is to combine the predictions of multiple base learners to obtain better predictive performance than could be obtained from any of the single base learners alone. The main idea behind ensemble learning is that a group of 'weak' base learners, which are learners that predict relatively poorly, come together to from a 'strong' learner, thus increasing the model's performance. The base learners can be trained using several different learning techniques, but also using a single learning technique multiple times. Ensemble learning can be used both for regression and classification problems. In a regression setting, the predictions of the base learners are combined using a (weighted) average. In a classification setting, the predicted class is the class that has been predicted the most by the base learners.

C.2.1 Bagging

Bagging (Breiman, 1996) is an ensemble algorithm that is based on combining predictions of multiple base learners trained on bootstrapped training sets. Let L be a learning set that consists of labelled instances $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$, where y is either a categorical or continuous response variable, and $\boldsymbol{x}$ is a vector of explanatory variables. Assume there exists a learning system that maps the explanatory variables to the response variable using the

learning set and call this predictor $\phi(\boldsymbol{x}; L)$. Furthermore, assume that there exist multiple learning sets $\{L_k\}_{k=1}^K$, each consisting of N independent and identically distributed (i.i.d) observations. The goal is to exploit the existence of multiple learning sets to improve on the predictor $\phi(\boldsymbol{x}; L)$, which only uses a single learning set.

The idea is to construct a predictor $\phi(\boldsymbol{x}; L_k)$ on each learning set L_k and then aggregate these predictions into a single prediction. If the response variable is continuous, one takes the average of all $\phi(\boldsymbol{x}; L_k)$: $\phi_A(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^K \phi(\boldsymbol{x}; L_k)$, where the subscript A in ϕ_A denotes aggregation. If the response variable is categorical and takes values $j \in \{1, ..., J\}$, then one aggregates all $\phi(\boldsymbol{x}; L_k)$ by taking the mode. More formally, let $N_j = \#\{k;\ \phi(\boldsymbol{x}; L_k) = j\}$ and take $\phi_A(\boldsymbol{x}) = argmax_j\, N_j$, that is, the j for which N_j is largest.

Usually, one only has a single learning set at their disposal and therefore cannot exploit the existence of multiple learning sets. However, one can construct replicates of the original learning set by bootstrapping the learning set L multiple times. More formally, one creates K bootstrapped learning sets $\{L_k^{(B)}\}_{k=1}^K$ by drawing N random instances with replacement from the original learning set L, K times. If the response variable is continuous, take $\phi_A(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^K \phi(\boldsymbol{x}; L_k^{(B)})$, and if the response variable is categorical, take $\phi_A(\boldsymbol{x}) = argmax_j\, N_j$, where $N_j = \#\{k;\ \phi(\boldsymbol{x}; L_k^{(B)}) = j\}$. Aggregating the predictions of predictors that were constructed on bootstrapped samples of the learning set is called '**b**ootstrap **agg**regat**ing**', and one often uses the acronym bagging. To summarise, Algorithm 1 describes the bagging algorithm in pseudo-code.

Algorithm 1 Bagging

Require: learning set $L = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$, predictor ϕ, integer K (number of bootstrap samples).
  for $k = 1$ to K **do**
    $L_k^{(B)}$ = bootstrap sample from L (N random draws with replacement)
    $\Phi_k = \phi(\boldsymbol{x}; L_k^{(B)})$
  end for
  if y is continuous **then**
    $\phi_A(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^K \Phi_k$
  else
    $\phi_A(\boldsymbol{x}) = argmax_j\, (\#\{k;\ \Phi_k = j\})$, for classes $j \in \{1, ..., J\}$
  end if
  Output $\phi_A(\boldsymbol{x})$

Breiman (1996) shows that the variance of the bagged estimator $\phi_A(\boldsymbol{x})$ is always smaller than or equal to the variance of a single estimator $\phi(\boldsymbol{x})$. This variance reduction is relatively large for unstable predictors such as decision trees. Moreover, the magnitude of the bias is roughly the same for both the bagged and the single predictor. This means that the performance of unstable predictors will improve by the bagging procedure, and that the performance of stable predictors will roughly stay the same.

C.2.2 Boosting

Boosting is an ensemble method that, as opposed to bagging, generates the base learners sequentially instead of in parallel, and uses weighted random subsamples of the data instead of random bootstrap samples. After a base learner is trained on a subsample of the learning set, the whole learning set is used to test the model, resulting in prediction errors for each data instance. The instances are then reweighted, such that instances that were predicted poorly are assigned large weights. These instances with larger weights have a higher probability to be selected in the next subsample. This way, each newly created base learner places emphasis on instances that are difficult to predict. By doing so, boosting tries to decrease the bias of the predictor. Furthermore, by taking a (weighted) average of the predictions of many base learners, boosting also reduces variance. Algorithm 2 shows the pseudo-code of the generic boosting algorithm.

Algorithm 2 Generic boosting

Require: learning set $L = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$, predictor ϕ, integer K (number of iterations).
  Initialise weights $\boldsymbol{w} = \mathbf{1}$
  for $k = 1$ to K **do**
    L' = subsample of L (weighted random sample with weights $\boldsymbol{w}$)
    $\Phi_k = \phi(\boldsymbol{x}, L')$
    Test Φ_k using L and update weights $\boldsymbol{w}$ according to prediction errors
  end for
  if y is continuous **then**
    $\phi_A(\boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^K \Phi_k$
  else
    $\phi_A(\boldsymbol{x}) = argmax_j \, (\#\{k;\, \Phi_k = j\})$, for classes $j \in \{1, ..., J\}$
  end if
  Output $\phi_A(\boldsymbol{x})$

Algorithm 2 has been implemented in many different ways. These implementations differ in how the data instances are tested and weighted. A popular boosting algorithm for binary categorical data is AdaBoost (Freund and Schapire, 1996). In addition to the generic boosting algorithm, AdaBoost assigns weights to each base learner, which are based on a logarithmic function of their average classification error. Base learners with an accuracy of over 50% are given positive weights, base learners with an accuracy below 50% are given negative weights, and base learners with an accuracy of 50% are given no weight. Furthermore, the weights in AdaBoost are updated exponentially after each iteration, and therefore instances that were predicted good (bad) are given a relatively small (large) weight. At the end of the algorithm, the predictions of the base learners are aggregated by a weighted majority vote, where base learners with a high classification rate have higher weights.

C.2.3 Gradient Boosting

Boosting suits additional models that are based on minimising a certain loss function. A loss function evaluates how well a model performs by measuring the deviation between a model's prediction and the actual value of data instances. By minimising this loss function the model finds optimal parameter values which will result in more accurate predictions. In many optimisation problems, however, finding the minimum of a loss function is computationally infeasible. A solution to this problem is to approximate the minimum of the loss function by using a forward stage-wise modelling approach, such as gradient descent. Gradient boosting (Friedman, 2001) is a technique that builds a model in a forward-stage wise fashion, and generalises it by allowing the minimisation of a loss function by gradient descent. The description of gradient boosting in this subsection closely follows the description as given in Friedman (2002).

Given a learning set $L = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$, the objective of gradient boosting is to find a function $F^*(\boldsymbol{x})$ that maps the explanatory variables $\boldsymbol{x}$ to the response variable y, such that the expected value of a certain loss function $\Psi(y, F(\boldsymbol{x}))$ is minimised

$$F^*(\boldsymbol{x}) = \underset{F(\boldsymbol{x})}{\arg\min}\, E_{y,\boldsymbol{x}}[\Psi(y, F(\boldsymbol{x}))]. \tag{C.11}$$

Common choices for the loss function include least-squares: $\Psi(y, F) = (y - F)^2$, least absolute deviation: $\Psi(y, F) = |y - F|$, and Huber-M: $\Psi(y, F) = (y - F)^2 I(|y - F| \leqslant \delta) + 2\delta(|y - F| - \delta/2) I(|y - F| > \delta)$, where $I(\cdot)$ is the indicator function. Boosting approximates the function $F^*(\boldsymbol{x})$ by means of an additive expansion of the form

$$F(\boldsymbol{x}) = \sum_{k=0}^{K} \beta_k \, \phi_k(\boldsymbol{x}), \tag{C.12}$$

where K is the number of iterations, $\phi_k(\boldsymbol{x})$ is the fitted base learner in the k-th iteration, and β_k is the expansion coefficient of the k-th base learner. Using Equation C.12, we can define $F_k(\boldsymbol{x})$ for $k = 1, ..., K$ as

$$F_k(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x}) + \beta_k \phi_k(\boldsymbol{x}), \tag{C.13}$$

where $F_0(\boldsymbol{x})$ is an initial constant estimate of $F^*(\boldsymbol{x})$. Gradient boosting finds $\phi_k(\boldsymbol{x})$ and the optimal value of β_k in Equation C.12 for loss function $\Psi(y, F(\boldsymbol{x}))$ using a two step gradient descent procedure. First, the base leaner $\phi_k(\boldsymbol{x})$ is fit to the so-called pseudo-residuals

$$\tilde{y}_{ik} = -\left[\frac{\partial \Psi(y_i, F(\boldsymbol{x}_i))}{\partial F(\boldsymbol{x}_i)} \right]_{F(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x})}, \tag{C.14}$$

and afterwards, the optimal value of β_k is determined by

$$\beta_k = \arg\min_{\beta} \sum_{i=1}^{N} \Psi(y_i, F_{k-1}(\boldsymbol{x}_i) + \beta \phi_k(\boldsymbol{x}_i)). \tag{C.15}$$

In words, gradient boosting finds a function that maps the explanatory variables to the response variable by combining boosting with gradient descent. It starts with an initial guess as model, and keeps adding sub-models to this model. Each sub-model explores the path that his predecessor has already followed and tries to improve this model by following the direction of the negative gradient to reach closer to the local minimum of the loss function. Algorithm 3 shows the pseudo-code of the gradient boosting algorithm.

Algorithm 3 Gradient boosting

Require: learning set $L = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$, predictor ϕ, integer K (number of iterations), differentiable loss function $\Psi(y, F(\boldsymbol{x}))$.

$F_0(\boldsymbol{x}) = \arg\min_\beta \sum_{i=1}^N \Psi(y_i, \beta)$

for $k = 1$ to K **do**

$\tilde{y}_{ik} = -\left[\frac{\partial \Psi(y_i, F(\boldsymbol{x}_i))}{\partial F(\boldsymbol{x}_i)}\right]_{F(\boldsymbol{x})=F_{k-1}(\boldsymbol{x})}, \; i = 1, ..., N$

Fit base learner $\phi_k(\boldsymbol{x})$ to $\tilde{y}_{ik}$, i.e. train using the learning set $\{(\tilde{y}_{ik}, \boldsymbol{x}_i)\}_{i=1}^N$

$\beta_k = \arg\min_\beta \sum_{i=1}^N \Psi(y_i, F_{k-1}(\boldsymbol{x}_i) + \beta\phi_k(\boldsymbol{x}_i))$

$F_k(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x}) + \beta_k\phi_k(\boldsymbol{x})$

end for

Output $F_K(\boldsymbol{x})$

C.2.4 Gradient Tree Boosting

Gradient boosting is used typically with decision trees (see Section C.1) of a fixed size as base learners. For this case Friedman (2002) proposes an adjustment to the gradient boosting algorithm which enhances the quality of fit of each regression tree and is known as gradient tree boosting. Let the base learner $\phi(\boldsymbol{x})$ be an L-terminal node regression tree. At each iteration k in the algorithm, the feature space is divided into L-disjoint regions $\{R_{lk}\}_{l=1}^L$, where a separate constant value is predicted in each region. Then, $\phi_k(\boldsymbol{x})$ can be written as

$$\phi_k(\boldsymbol{x}; \{R_{lk}\}_{l=1}^L) = \sum_{l=1}^L \bar{y}_{lk} \, I(\boldsymbol{x} \in R_{lk}), \tag{C.16}$$

where $\bar{y}_{lk} = ave(\tilde{y}_{ik} | \boldsymbol{x}_i \in R_{lk})$ is the average of $\tilde{y}_{ik}$ in Equation C.14 in each region R_{lk}. Furthermore, decision trees allow β_k in Equation C.15 to be solved separately within each region R_{lk}, and Equation C.15 reduces to

$$\beta_{lk} = \arg\min_\beta \sum_{\boldsymbol{x}_i \in R_{lk}} \Psi(y_i, F_{k-1}(\boldsymbol{x}_i) + \beta). \tag{C.17}$$

The current model $F_{k-1}(\boldsymbol{x})$ is then updated separately in each corresponding region

$$F_k(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x}) + \sum_{l=1}^L \beta_{lk} \, I(\boldsymbol{x} \in R_{lk}). \tag{C.18}$$