Vrije Universiteit Amsterdam

Otrium

Otrium

Master Thesis

Prediction of customer lifetime value in e-commerce fashion retail

Author: Cathelijn Kuijt (2625803)

1st supervisor: Prof. Dr. Rob van der Mei
Daily supervisor: Sam Sundar (Otrium, Head of Data&Analytics)
2nd reader: Dr. Eliseo Ferrante

A thesis submitted in fulfillment of the requirements for the VU Master of Science degree in Business Analytics

July 9, 2022

Abstract

This research focusses on the prediction of customer lifetime value (CLV) for the online outlet fashion store Otrium. Customer relationship management (CRM) becomes more and more important to consider in business decisions. Tracking CLV allows the company to better allocate the marketing spend and maintain the high value customers. Otrium defines the CLV as the total profit a customer deliver in their entire lifetime while considering any costs that are associated with the orders. The goal of this research is predicting the CLV per customer for 12 months in the future using historical purchase information about the customers.
This prediction is done using two methods, where one methods uses probability prediction models and the other uses machine learning techniques. These are compared on their performance on multiple areas. The results showed that:

1. For the 12 month CLV prediction for the entire customer base, the machine learning method performed best.
2. For the monthly CLV prediction, the probability models performed better.
3. The customer level 12 month CLV was best predicted by the machine learning method.

For the Otrium use case, where the 12 month CLV metric will be used most in customer groups based on certain characteristics, the customer level CLV is most important. Therefore, the machine learning would be the best choice.

Contents

1

Introduction

Background. Otrium (`www.otrium.nl`) is an online fashion outlet store, which sells fashion items of various brands. The mission is to decrease the amount of excess inventory for fashion stores and to make sure that all clothing is worn. The company partners with the brands to help them sell their last season's items with discounts, solving the challenge of unsold inventory. The partnership with brands includes deals in which the brand hands over a specific amount of inventory to Otrium, which is then stored in one of the Otrium warehouses. Otrium tries to sell as many of these items as possible using various discounts via their own platform, while maintaining the item price above an agreed selling price. The sold items are paid out to the brand after subtracting the commission, which is paid to Otrium for their services.
Currently, there are around 3 million items available on the platform with discounts up to 75% from more than 400 different brands. Since the founding in 2015, Otrium has acquired over 4 million registered members of which over 3 million have made one or more purchases on the platform. Otrium is currently available in 20 different countries with warehouses in the Netherlands, United Kingdom and USA.

Customer Relationship Management. Overall, the online retail market has grown massively in the past few years, where more and more customers prefer online shopping over physical shopping. This resulted in an increase in the number of online fashion stores, and subsequently, in the online outlet stores competing in the same market as Otrium. To stay ahead of these competitors the business strategy has to keep improving in order to keep increasing the profitability. The most important factor of profitability is the factor that provides the revenue: the customers. Therefore, Customer Relationship Management

(CRM) is a key focus point for many companies and focuses on the relationship between the company and current or potential customers.

Problem. There are multiple key performance indicators that can measure the performance of the company and on which various business decisions can be made regarding CRM. For example, one can choose to focus on the revenue of the customers or on the acquisition and retention of the customers. In light of both of these things, the main question is how do we keep customers coming back to our platform and especially the ones that bring in the most revenue?

The company can decide to use on the gross merchandising value (GMV) as a measure for their company-wide performance. However, some customers might costs more than what they bring in due to any related costs. Therefore, it is better to express profitability of a customer in terms of Customer Lifetime Value (CLV). This is defined as the total profit a customer will generate for a business throughout their relationship. Using this metric, decisions can be made with the goal of obtaining and maintaining the customers with the highest CLV and getting the highest revenue stream. From a marketing perspective, such decisions can include targeting specific customers within a certain group of historical and predicted high value customers. This enables better distribution of costs over high and low profitable customers to try to effectively produce the most revenue with every cent spent.

Research Questions. The main research question for this study is: *What is the best method to predict CLV per customer for Otrium?* For this prediction, CLV is divided into multiple components. Every component is predicted separately to, in the end, predict the monthly CLV per customer.

First, the monthly order frequency per customer and its order value were predicted resulting in the revenue per customer, which is the main component of the prediction. This part of the prediction can be done using machine learning models, known for their good performance in forecasting problems. However, research has shown that probabilistic models also tend to do well for this matter, so it was decided to compare the two.

Once the revenue per customer is determined, a few other factors have to be taken into account, such as return rates and any costs related to the order. For simplicity and maintainability, it was chosen to predict these components using several business assumptions made based on historical data.

The final outpu gives the monthly CLV per customer, from which the total CLV per customer can be obtained by taking the cumulative sum over the lifetime of the customer.

These two methods are compared based on their final CLV to find the prediction model with the best performance.

The performance of the models on the various components was compared in order to identify areas of improvement for both methods. With the final CLV prediction it was decided which model performed best on the basis of three subquestions:

1. *Which model predicted the best 12 months CLV for the entire customer base?*
2. *Which model predicted the best monthly CLV over the next 12 months?*
3. *Which model predicted the best 12 month CLV per customer?*

These questions were also answered for a shorter prediction horizon of 3 months to see how the models performed on the short term.

Report Structure. The report is structured as follows. First, a review of scientific literature on CLV and its prediction is given including the review of various performance measures, which can be found in Section 2. Second, Section 3 describes the data and the various models used for the prediction. The results of the model comparisons per component and the performance of the final CLV prediction are discussed in Section 4. Finally, a conclusion of this research is given in Section 5 along with suggestions for future research.

2

Literature Review

2.1 Customer Lifetime Value

Customer Lifetime Value (CLV) is slowly becoming one of the most important key performance indicators in the retail business. Therefore, much research is done regarding the metric itself and the prediction of it. For some customers it might happen that they have a negative CLV while having a high revenue due to free shipping coupons, for example. Therefore, CLV is a better metric to steer on than for example Gross Merchandising Value (GMV).

The way CLV is used within a company and the business decisions that are taken in light of CLV can differ widely. The well-known e-commerce company ASOS, for example, has conducted a study about CLV in order to nurture high-value customers to increase the average shopping frequency, identify customers that have a high churn risk and control the amount spent on retention orders to decrease the churn rate [4]. A study for Groupon used CLV mainly to focus on Customer Relationship Management (CRM) [19]. For example, it was used to identify the ideal target audiences for specific promotional offers and personalized customer messaging in order to trigger retention and maintain a positive relationship with customers.

The business strategy of a company, and therefore the calculation of CLV, differs on various aspects. First of all, it depends on the membership and contract strategy of the company. Some studies are conducted in settings where a contract or membership is involved, this can indicate that there is a subscription involved which for example includes regular customer fees [15], [8]. In this context, a company not only knows exactly when revenue for a customer comes in, but also the amount of money that comes in. And second,

the contract allows the company to know when a customer becomes inactive, which is when a contract ends.

For fashion retail stores such as Otrium the context is *non-contractual*, that is, customers can make a purchase at any time and of any amount, they are not bounded by any contract. This makes it more difficult to predict the CLV per customer and also to recognise a customer churning.

So not only is it needed to define CLV, the definition of churn should also be constructed, which can differ per business. Chamberlain et al. [4] and Van der Veld et al. [19] define CLV as the sales made, excluding returns, and decided to define a customer as churned if they have not placed an order in the past year. Berger at al. [2] uses the contribution margin instead, which takes the costs to acquire a customer into consideration as well as the cost of order processing, handling and shipping.

2.2 Prediction of CLV

2.2.1 Probability Models

Frequency prediction. Looking at studies about CLV prediction models, it shows that some widely used methods are the probabilistic models, also known as the 'Buy 'Til You Die' models. These models base their prediction on the past purchase behaviour of the customer. Using that behaviour, the models predict the number of purchases a customer is going to make as well as the probability that a customer will still be alive in that period. Ehrenberg et al. [13] used such a probability model for CLV prediction with the negative binomial distribution (NBD), where the assumption was made that the purchases made by a customer are random around their customer-specific, time-invariant rate and that this purchase rate differs per customer. However, the purchase rate of a customer is most likely not time-invariant because of external factors and seasonality. Therefore, Schmittlein et al. [7] took this into consideration and introduced the Pareto/NBD model, which allows for time-variant purchase rates along with the other assumptions. This model assumes that purchase frequency follows a Poisson distribution and that the lifetime of a customer follows an exponential distribution.

However, this model has one main challenge, which is the likelihood function. This causes the parameter estimation to be rather difficult. Accordingly, the Beta-geometric/NBD (BG/NBD) model was introduced to overcome this challenge [11]. The model closely mirrors the Pareto/NBD model, the only difference being that the churning of a customer

does not happen at any random point in time but always directly after a purchase. This means that a customer can become inactive after a purchase with a certain probability.

Another modification was made to the Pareto/NBD model by Bemmoar et al. [1], which modelled the customers' active duration as a Gamma/Gompertz distribution, which made the model more flexible.

Revenue prediction. These models can predict the purchase frequency of a customer in a specific time, but for the CLV, it is also necessary to know the value of those future orders. Since the spending data can be right skewed, meaning more purchases of a lower amount and a long tail of some purchases with extremely high amounts, Colombo et al. [6] assumed that it followed a gamma distribution. Using this assumption, Fader et al. [10] introduced the gamma-gamma model of monetary value, which can be used to predict the expected average order value.

However, this model assumes that the number of transactions is not correlated with the average order value. Glady et al. [14] found a possible dependency between the two, and therefore, proposed another Pareto/NBD based approach: the Pareto/Dependent model. This model takes into account a possible dependency rate, which can differ across customers.

2.2.2 Machine Learning Models

When thinking of predicting any kind of metric, a machine learning model is usually the first thing that comes to mind. So it makes only sense that there were also studies that used machine learning models for the prediction of CLV. Pfeiffer et al. [18], Pauwe et al. [17] and Jasek et al. [16] used the Markov Chain model in combination with decision trees to predict the next purchasing behaviour based on the current state of a customer. Van Der Veld et al. [19] uses random forests to predict CLV, which includes customer engagement features such as users' demographics, engagement and overall relationship with the company as features. Firstly, a binary classifier is run on the customers to split the buying and non-buying customers from each other. The customers that are expected to buy, are split into five groups and regressors are trained for each group. Drachen et al. [9] uses both random forest and extreme gradient boosting (XGBoost) to predict CLV in the freemium gaming industry. It uses features that measure the social behaviour of the customers for their prediction.

Chamberlain et al. [4] notes that the difficulty of predicting CLV lies in the fact that most customers have a CLV of zero and for the customers with a non-zero value, the

values differ massively. To overcome this issue, they suggest to model CLV percentiles with a random forest regressor after which they are mapped back to real CLV values.

2.2.3 Performance Measures for Model Comparison

In order to compare the performance of the used models, it has to be decided which performance measures should be used. Since the models can be used for different goals, the models are assessed with respect to different tasks.

The first measure focuses on the performance of CLV prediction on individual customer level. For this task, Donkers et al. [8] used the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) as percentage of average CLV. To improve robustness of these measures, Glady et al. [14] discards the largest 1% of the prediction errors so that the outliers don't dominate the analysis. Glady et al. [14] also uses the Spearman's correlation as a measure of correlation between the predicted and actual values. Donkers et al. [8] uses a hit rate criterion, where the customers are categorized based on their true CLV and splitted into groups. The hit-rate is then computed as the percentage of customers that falls in the correct category based on their predicted CLV.

Since the prediction is done on a monthly level, the second performance measure focusses on that. Jasek et al. [16] uses the Mean Absolute Percentage Error (MAPE) for this task. The last task assesses the performance on the whole customer base over all of the 12 predicted months. Donkers et al. [8] uses the percentage deviation from the true value of the total customer base. Jasek et al. [16] uses the forecast versus actuals metric, which is defined total predicted CLV divided by the total actual CLV value times 100.

3

Methodology

In this section the method used to predict the CLV per customer for the upcoming 12 months will be described. The first section will explain various things about the data, such as what variables it includes and the pre-processing steps. After which the various methods used for the prediction models will be explained along with their features and implementation. And lastly, the performance measures used to compare these models are described.

3.1 Data

The dataset contains information of transactions made on Otrium. The data consists of orders made between September 25 2015 until June 31 2022. Since only 4% of all orders was made before 2019, it was decided to only use orders made after January 1 2019. Also, since Otrium maintains a 60 day return period it was decided to only use data from before that return period. This way it is ensured that all numbers are final and all returns are factored in.

Figure 3.1 shows the total number of orders and total revenue per month. It shows that since 2019 the company has grown massively. There are a few social factors that might have had an influence in the growth in these past two years. First of all, the COVID pandemic, of which the start and the (unofficial) ending are indicated by the gray dotted lines. Because of the pandemic, a lot of physical stores closed down for months, which hugely impacted the performance of the entire e-commerce market. So from the start of this pandemic, an upward trend can be seen in the number of orders and revenue. The

(unofficial) ending of this pandemic and its lockdowns was around the end of 2021 which most likely was the cause of a massive dip in sales for Otrium.

The figure also shows that there is some seasonality in sales. For example, every year there is a peak in sales in November. This can be explained by the fact that Black Friday falls in November, where the items on the platform are offered with an even higher discount than usual. These promotions in combination with Christmas shopping result in a higher sales compared to the rest of the year followed by a slower sales period.

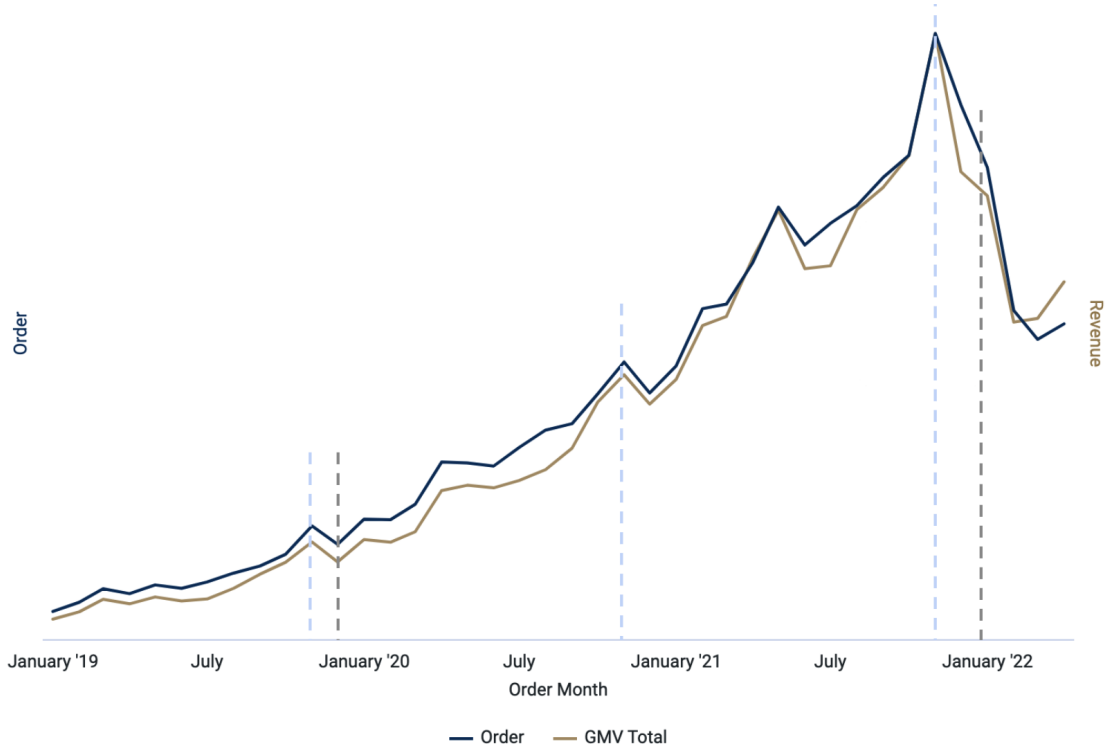


Figure 3.1: Number of orders and revenue of those orders from January 2019 up to April 2022. *Blue dashed lines*: November, which is the month of Black Friday. *Gray dashed lines*: The start and (unofficial) end of the COVID pandemic.

The attributes that were available concerning these orders are shown in Table 3.1.

Table 3.1: Attributes in the dataset available per order

Metric	Definition
Order Date	Date at which the order was placed
Order ID	Unique id of the order
Customer ID	Unique id of the customer placing the order
Shipping Country	Country to which the order was shipped

Table 3.1 continued from previous page

Platform	If the order was placed on the app or on the web page
Final Amount	The final amount of the order (€)
Average Item Price	Average prices of the items in the order (€)
Refund Amount	Total amount of the orders that were returned (€)
Main Brand	The brand with most revenue in the order
Coupon %	The coupon percentage used for the order
Quantity Sold - Gross	Number of items in the order
Quantity Returned	Number of items in the order that were returned

Figure 3.2 shows the fraction of customers per number of orders placed after January 1 2019. This shows that the majority of the customers placed one order. For these customers it might be difficult to predict the CLV, since there is not much known about their purchase pattern.

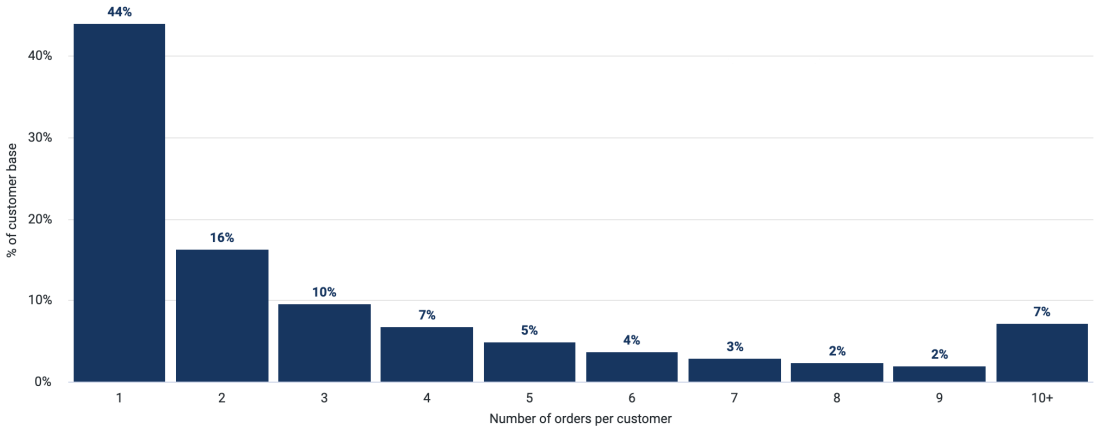


Figure 3.2: Fraction of the customer base per total number of orders placed on the platform

The dataset also included a few details about the customers that made the orders, Table 3.2 shows these attributes.

Table 3.2: Attributes in the dataset available per customer

Metric	Definition
Customerid	Unique id of the customer
Gender	Gender of the customer
First Purchase Date	Date when the customer placed their first order
Age	Age of the customer

In order to compute the actual profit for Otrium of a specific order, the costs of that

order have to be taken into consideration. These include costs that are associated with the shipping and packaging of an order or a return. But also ancillary revenues, which are values that the customer pays in order to compensate for part of these costs. Table 3.3 shows the various costs and ancillary revenues that are available in the dataset with a short description. Section 3.4 will give more details into these costs.

Table 3.3: Attributes in the dataset for any costs and ancillary revenues

Metric	Definition
Commission rate	The % commission of the final revenue of the order
CPO Shipping	Cost per order for the shipping of the order
CPO Outbound	Cost per order for the rest of the process of an outbound order
CPI Outbound	Cost per item for the process of an outbound item in an order
CPO Shipping Return	Cost per order for the shipping of a returned order
CPO Return	Cost per order for the rest of the process of a returned order
CPI Return	Cost per item for the process of an returned item
CEM Costs	Costs for the customer support
Shipping Label	The amount paid by the customer for shipping an order
Return Label	The amount paid by the customer for shipping a returned order
CPRO	Cost per retention order for marketing

As mentioned before, it was decided to only use orders made after January 1 2019. In light of that, it was decided that customers that placed there first order before then were dropped from the dataset. The reason for that was that some of the model features require the entire order history to be available.

Since 98% of the customers were missing a value for age, it was decided to remove age from the dataset. Other than age, there were no more fields that contained any missing values.

3.2 Frequency and Revenue Prediction

The first two steps in predicting CLV are predicting how many orders a customer is going to place in the future and the value of those orders. The combination of the two will result in predicted revenue per customer, which is the total amount a customer has spent on the platform in a specific period of time. Two methods were used for this and this section will explain those models along with the features used and the implementation of the model.

3.2.1 Probability Models

The first part of this section explains the BG/NBD model which was used for the prediction of the number of transactions of a customer in a specific period of time in the future. The second part explains the Gamma model which was used to determine the value of the predicted transactions per customer.

3.2.1.1 BG/NBD Model

The BG/NBD model is an extension of the Pareto/NBD probability model that predicts the order frequency in a certain period of time, this section will highlight the most important aspects of the model based on the description in [11] and [7].

This model tries to predict if a customer is alive after a certain time and how many times they will order in this period based on the historical transaction data of that customer. When dealing with a non-contractual and non-membership setup, the prediction of churn and the order prediction per customer is not as straightforward as it is for a contractual setup. Therefore, the Pareto model enforces a set of assumptions for both components. Along with these assumptions the model also requires only two features: (1) *recency* (i.e. the time of the customer's last purchase), and (2) *frequency* (i.e. the total number of purchases a customer made in their lifetime up until period). The BG/NBD model mirrors this Pareto model almost exactly, except for the point in time that a customer might churn (Assumption). The assumptions for the model are listed below [11]:

Assumption 1 ***Poisson purchases.*** *While active, the number of transactions made by a customer in a time period of length t is distributed Poisson with mean* $t\lambda$.

Following the assumptions enforced by Poisson, these purchases have exponential interarrival times with a specific arrival rate. Since the exponential distribution comes with a memoryless property, which means that all purchases are random and independent of each

other. For retail purchases, this assumption is quite natural since customers can place orders whenever they like and generally the orders are placed independently of each other.

Assumption 2 ***Geometric churn.*** *After every transaction, a customers becomes inactive with a probability p.*

This is where the BG/NBD model differs from the Pareto/NBD model. Where the Pareto/NBD model assumes that the churn of a customer can happen at any point in time independent of the time of the purchase, the BG/NBD model assumes a customer always churns immediately after making a purchase. Which is the same as assuming that a customer churns because of something relating their purchase. Although the assumption for Pareto/NBD seems more logical in Otrium's case, since it is also possible that a customer churns because they are unsatisfied with the items on the platform, it was still decided to use the BG/NBD model. The reason for this decision is that the computation for the BG/NBD model generally takes less time and is easier to implement [11].

Assumption 3 ***Gamma purchase rates.*** *Heterogeneity in the transaction rate λ across customers follows a gamma distribution.*

Every customer is different, also in their purchase pattern. Some customers may buy more frequently than others, so therefore their purchase rates are different from each other. Therefore, Fader et al. [11] proposed to assume that the purchase rates are Gamma distributed to account for this variability between customers.

Assumption 4 ***Beta churn probabilities.*** *Heterogeneity in churn probabilities across customers follows a beta distribution.*

Not only purchase patterns may differ between customers, but also the churn per customer can vary. Some customers might churn after their first purchase, while others might stay for years after their first order. Therefore, Fader et al. [11] proposed to assume that these churn probabilities follow a Beta distribution.

Assumption 5 ***Independent rates.*** *The transaction rate λ and the churn probabilities p vary independently across customers.*

Overall, one would say that if a customer buys frequently, you would assume that the customer is satisfied with the company and therefore would be less likely to churn. However, it can only take one bad purchase for the customer to churn, which is completely random

per customer. Therefore, it makes sense to assume that their is no specific correlation between the transaction and churn rate.

Using these assumptions and their distributions, the probability of someone buying at a specific time as well as the expected number of transaction up to a certain time can be computed. These derivations can be found in [11].

3.2.1.2 Gamma-Gamma Model

After obtaining the predicted frequency per customer in a specific period of time, one has to determine the value of those purchases in €. The Gamma model is used for this, which is described in [10] and this section will explain the most important parts.

Assumption 6 *The monetary value of a customer's given transaction varies randomly around their average transaction value.*

The average transaction value is known to be right-skewed, since the cheaper items are in general purchased more frequently than the more expensive ones. To account for this variability, the monetary value is modelled as a Gamma distribution.

Assumption 7 *The average transaction values vary across customers but do not vary over time for any given individual.*

It was assumed that customers do not increase or decrease in their average order value per customer. Although this might not always be true, generally a customer who buys cheaper items their first order, is most likely to continue buying items in that price range. Other customers might just have more expensive taste, which will most likely not change drastically over their lifetime. Therefore, the scale parameter of the monetary's Gamma distribution mentioned in Assumption 6 follows a Gamma distribution across the customer base with its own scale and shape parameter [10], hence the name Gamma-Gamma.

Assumption 8 *The distribution of average transaction values across customers is independent of the transaction process.*

Fader et al. [10] assessed this assumption and came to the conclusion that the correlation between the transactions and their values is not significant, which therefore allows to assume independence.

Using these assumptions and the characteristics of the Gamma distribution, the average expected transaction value and the conditional expected mean purchase value per customer were computed. The derivations can be found in [10].

3.2.1.3 Features and Implementation

Features. These two models only needed four different variables as input: *lifetime*, *recency*, *frequency* and *monetary*. The *lifetime* is the number of days since the first purchase of the customer up until the end of the feature period. The *recency* is the number of days since the last order was placed. The *frequency* is the number of orders that was placed in the entire lifetime of the customer, so since their first purchase up until the end of the feature period. And lastly, the *monetary* is the total amount the customer spent on all the purchases made in their lifetime up to the end of the feature period.

Implementation. The implementation for the two prediction models is shown in Algorithm 1. The functions used are from the `lifetimes` package. And the output is the predicted frequency and predicted revenue per month per customer.

Algorithm 1 Implementation of the BG/NBD and Gamma-Gamma model using

Require: import `lifetimes`

```
features = summary_data_from_transaction_data(data)
BG_NBD = fit.BetaGeoFitter(features)
Gamma_Gamma = fit.GammaGammaFitter(features)

while i in (months in prediction period) do
  pred_frequency[i] = BG_NBD.expected_freq_up_to_time(until last day of i )
  pred_ordervalue[i] = Gamma_Gamma.revenue_up_to_time(until last day of i )
  if i > 0 then
    pred_frequency[i] = pred_frequency[i] - pred_frequency[i − 1]
    pred_ordervalue[i] = pred_ordervalue[i ] - pred_ordervalue[i − 1]
  end if
end while
=0
```

3.2.2 Machine Learning Method

This section will explain the second method used to predict the revenue per customer. The gradient boosting method used for the frequency prediction will be explained, followed by the assumptions made to predict the revenue per customer. The remainder of the section is dedicated to explaining the features and the implementation used for the models.

3.2.2.1 Gradient Boosting

The second model used was a gradient tree boosting method. These models are seen as the more easily understood machine learning models as they closely mirror the human decision making. It was chosen to use Extreme Gradient Boosting (XGBoost), which is known to be very effective on both small and large datasets and which doesn't need any normalized features. It also includes regularization hyper-parameters, which means that mimimal to no feature selection is needed, since the model already handles that. And also, the model is known for being fast and easy to implement. This section will highlight the most important characteristics of the model as described in [5] and [12].

In order to understand gradient boosting, there are a few other methods that will be explained that are the basis of the model. Starting with 'simple' decision tree learning, which is a supervised learning method. A decision tree consists of nodes, which is splitted downwards into branches up until the leaf nodes at the end of the tree. Every node represents a test on a certain attribute, the outcome of this test will determine which way down the branches the model goes. So, as an example, let us simplify the decision of a customer buying an item *yes* or *no* as a classification problem. Where there might be multiple factors that can influence this decision.An example of such a tree is shown in Figure 3.3, where "1" is when the customer buys the item and "0" if not based on the color and prize of an item.

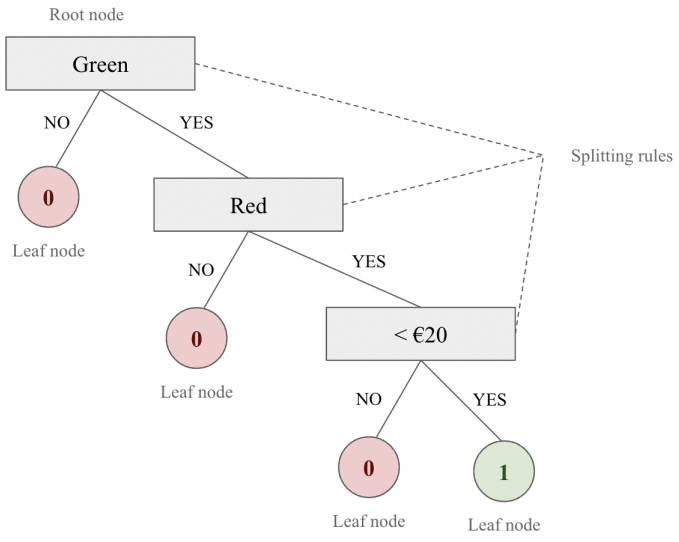


Figure 3.3: Example of a binary decision tree for a customer buying an item "1" or not "0"

The decision tree is trained on a sample of items where these characteristics can differ per item. After training, the model can predict if the customer is going to buy an item with certain attributes, without ever having seen that item. The decision tree can also be used for regression problems for which the leaf nodes are continuous numeric values. This is the case for this research where the goal is to predict the number of orders per customer.

Some extensions of decision tree learning use *bagging* [3], for example random forest. In those models, multiple decision trees are trained in parallel on subsets of the data. The outcome of the model takes the average of the outputs of the various decision trees.

The gradient boosting method applies bagging, but instead of training them in parallel, the different trees learn from each other, which is called boosting. So there are multiple iterations, where in each iteration a tree is trained on a subset and after each iteration tree is updated according to a certain objective. Hence, the name gradient boosting, where the model gradually learns throughout the training based on a certain objective.

Using features from historical data, this model can predict the customer level order frequency.

3.2.2.2 Business Rules

After obtaining the order frequency per customer for a specific period of time, the value was determined. For time and maintainability purposes it was decided not to use a model for this prediction, but to use assumptions using data of the last 12 months. Based on experience, it is known that in general the average order value mainly depends on the number of orders per customer and the market.

Table 3.4: Correlation between the value of the orders and some order specifics

Attribute	Order Value
Order Count	0.064
Netherlands	-0.013
Belgium	-0.028
Germany	0.082
France	-0.074
Female	-0.0065
Male	0.016

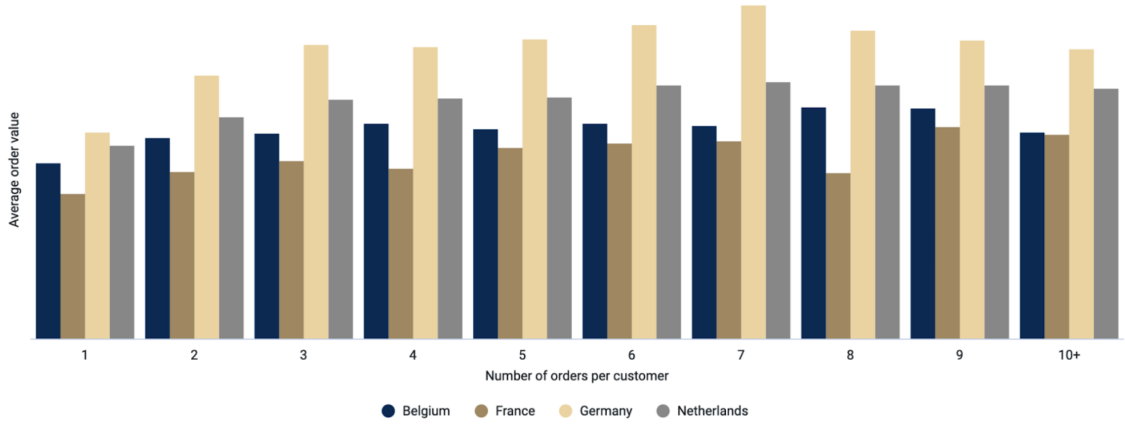


Figure 3.4: The average order value per number of orders per customer and country

Table 3.4 shows the correlation between the order value and these metrics and for comparison also with the gender of the customer. This shows that the the correlations are all relatively small, but it does show that the average order value correlates stronger with the markets than the genders. It shows that if the order has shipping country Germany, that the order value is generally slightly higher than for the other countries, which is also shown in Figure 3.4.

The order count has a positive correlation with the order value. Which means that the first order of a customer has generally a lower order value than the orders following it.

So for the prediction of the order value, the average order value of last 6 months for the number of purchases made in the past, including the predicted frequency, in combination with the country in which the customer has ordered most frequently is taken.

3.2.2.3 Features and Implementation

Along with the features used for the BG/NBD model, some other features were used in order to predict the order frequency using the XGBoost method.

Features. It was decided to both include *time-invariant* features as time-variant features to be able to account for seasonality in sales during the year. The *time-invariant* features were customer specific attributes such as the features used for the BG/NBD model as recency, frequency and monetary. Some other attributes were added as well to cover more of the customer behaviour, which makes it easier for the model to recognize certain patterns

in customer purchases.

Time-invariant features
It was decided to construct a score per customer based on the recency, frequency and monetary metrics. A customer with a low recency, high frequency and high monetary value, generally can be seen as a loyal customer. But a customer with a high recency and a high frequency, might be a customer that was loyal once, but has churned now, so therefore gets a different score.

This score was determined using clusters for the three metrics. The optimal number of clusters per metric was determined using the Elbow method, after which the clusters were defined using the KNN clustering method. Every customer was then assigned to a cluster and the overall score for that customer was defined as the sum of the three clusters that that customer belonged to.

To take into consideration the most recent activity of a customer, the number of days between last three orders was used. For instance, the number of days between the last order and the second to last order was included as *DayDiff*. The same was done for the second to last with the third to last (*DayDiff2*) and the one before that (*DayDiff3*).

Also, the *lifetime* of the customer was included, which is the number of days since the customer made their first order up to the first day of the month that had to be predicted.

Next, some categorical features were constructed and included as dummy variables. These features included the *Gender* of the customer which was either *women*, *men* or *unknown*. The *Most ordered country* was the country where most of the orders from that specific customer were shipped to. And also two features about the platform at which the order was placed, web or app, for the first order of the customer (*Platform FO*) and the platform that most revenue was ordered on (*Platform AVG*).

Another categorical feature that was included had to do with the brands that the customer purchased items from. Since there are over 200 brands available on Otrium, it was decided to only select the top-10 brands in sales and specify the rest as other to reduce the number of dummy variables. Using these new definitions the brand that had the most revenue in the first order of the customer (*Brand FO 10*) was included and the brand with the most revenue over all orders of the customer (*Brand AVG 10*).

The last feature on customer behaviour was related to their historical monthly purchases for the last 12 months called *M1* to *M12*, where *M1* is the last month of the feature data and *M12* 12 months before that.

Time-variant features
The sales shown earlier in Figure 3.1 showed that there is seasonality during the year. Therefore, for the 12 month prediction of monthly CLV time-variant features were added to account for that.

To account for a specific trend in the Otrium-wide performance, the sales of the whole company were included for the last 3 months called *Otrium_ M1*, *Otrium_ M2* and *Otrium_ M3*, where *Otrium_ M1* is again the sales of the last month of the feature dataset.

Since the model used one feature set when predicting 12 months, which will be explained in the implementation paragraph, the number of months since the measured features differed. This was taken into consideration by adding the feature *Months since feature end*.

The last feature also focused on the company-wide performance and accounted for the monthly seasonality during the year. It looked at the monthly performance of the company in the last complete year and computed a relative performance compared to January of that year. So the sales in that year for the to-be-predicted month was divided by the sales in January of the last complete year. This feature was included as *Relative O-Performance*.

Hyperparameter tuning. The XGBoost method has multiple parameters that had to be tuned in order to achieve the best performance. The parameters are described in Table 3.5.

The *max_ depth* and *min_ child_ weight* were tuned first using a Grid Search with 3 cross-validations and with MAE scoring. Then *gamma* was tuned by trying a number of values between 0 and 0.6 and with the use of cross validation finding the optimal value. Subsequently, another two GridSearch's were executed, first for the parameters *subsample* and *colsample_ bytree* and then for the parameters *alpha* and *lambda*. The last parameter to be tuned was the learning rate, which was tuned by trying various values between 0 and 0.5 and again finding the optimal value using cross-validation.

Table 3.5: Short explanation of hyper parameters for XGBoost regression.

Parameter	Description
Learning Rate	How fast the model learns
Max depth	Depth of tree
Min child weight	Min. sum of instance weight needed in a child

Table 3.5 continued from previous page

Gamma	The minimum loss reduction required to make a split
Subsample	Subsample ratio of the training instances
Colsample by tree	Subsample ratio of the columns
Alpha	L1 regularization term on weights
Lambda	L2 regularization term on weights

Implementation. In order to train on the time-variant variables used in this model, a training set with a sliding window was constructed (Figure 3.5). One series of the training set consisted of one feature set of 24 months. Because of the data availability start, the feature set sizes varied between 20 and 24 months. With this feature set, the target variable was first taken for 1 month in the future then for 2 months in the future, and so forth. This was done for 6 months in the future. So for one set, a customer with that feature set is included 6 times. It was decided to use only 6 months in the future in these series instead of 12 months, because otherwise the model would train on very old feature data compared to the prediction period.

The training set consisted of four of those sets, where each feature set was shifted with one month and where all dates were from before the maximum feature date. This maximum feature date is 12 months before the end of the data set, because those months are used as testing period.

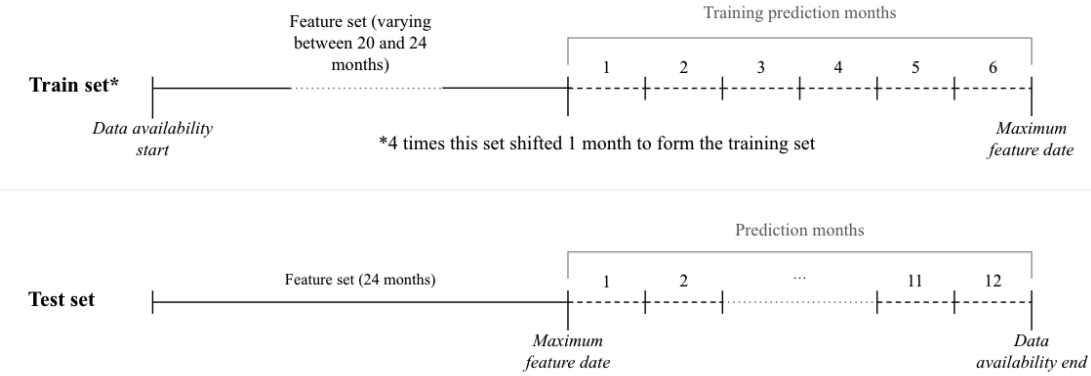


Figure 3.5: The setup of the train and test set used to predict the monthly frequency using the gradient boosting method.

After training the model on this training set, the next 12 months were predicted in order to check the performance of the model. Important to note is that there were also new customers in those 12 prediction months, but since these customer don't have any behavioural

data so the model is not able to predict the frequency for those customers.

The `XGBRegressor` from the `xgboost` package was fitted on the training set mentioned above using the tuned hyperparameters. The fitted model was then used to predict the frequency of the test set. Using this frequency the revenue per customer was determined using the assumptions made depending on the order count including the predicted frequency and the country of the customer. So every order in the predicted frequency was assigned a different order value. However, since the prediction model predicts float numbers rather than integers, the order value was multiplied by any partial orders.

3.3 Net Revenue Prediction

After obtaining the monthly revenue per customer, the amount that was returned in € by customers was predicted. This was done using assumptions concerning the return rate of the order value. Figure 3.6 shows that the return rate is highly correlated with market and gender. For example, the return rate in Germany is overall the highest, where as France sees the least returns. It also shows that in all countries, females have a higher return rate than men.

Since return rates tend to differ highly between customers and it might be a more personal thing, it was decided to also take that into consideration. Therefore, the average return rate of a customer was used whenever the customer has made 4 or more orders. If the amount of orders placed is lower, the average return rate for the gender and most ordered country was chosen as the return rate.

3.4 CLV Prediction

After executing the steps mentioned above, the netto amount spent on the platform by a specific customer was obtained in some period in the future. However, this is not the amount that counts as final profit for Otrium, which is needed for the CLV metric. In order to obtain that final profit, a few other costs and ancilliary revenues had to be taken into account. An overview of the build-up of CLV defined by Otrium is shown in Figure 3.7. This section explains the build-up step-by-step and how it was chosen to predict those the various metrics for the final CLV prediction model.

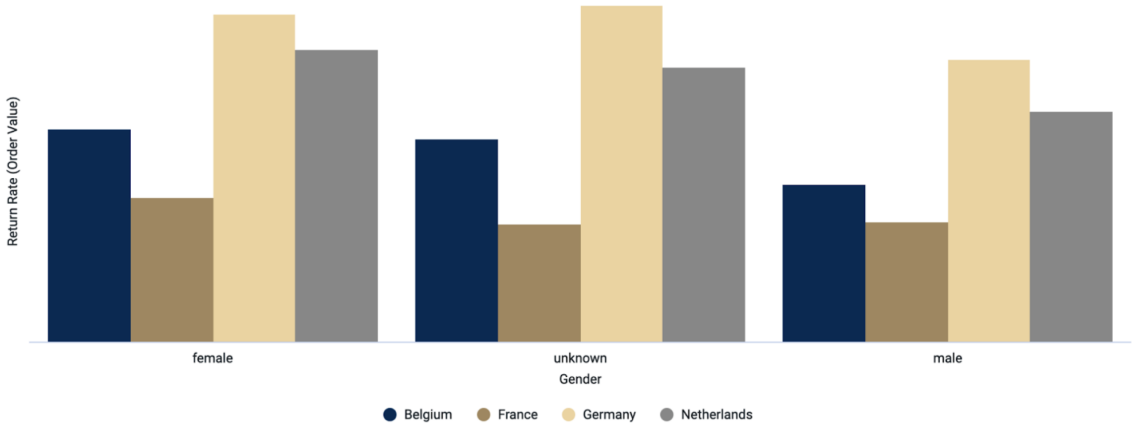


Figure 3.6: The return rates of the order value per gender and country

Ancilliary revenues. Whenever a customer makes a purchase they also have to pay for shipping costs, except when they have a discount code for free shipping (*Shipping Rev.* in Figure 3.7). The value for this shipping cost was added to the net revenue prediction, assuming no free shipping coupon is used.

When a customer returns their order, they have to pay a return label amount (*Return Label Rev.*). For the prediction, whenever an order was a predicted return, the value for these costs payed by the customers was added to the revenue. Again, the assumption was made that there was no free return coupon used.

And lastly, there might have been transaction costs related to the order (*Transaction Rev.*). However, since these transaction costs are only payed by the customers whenever they pay with certain payment methods such as creditcard, which nearly never happens, it was decided to exclude these costs from the prediction.

Commission. Otrium is a platform that sells items from brands, so whenever a purchase is made the revenue is actually the brands' (Brand Revenue). However, Otrium is paid for their services and the amount Otrium gets is determined through commission. The commission rate is different per brand, which is agreed upon with the brand at the very beginning of the relationship and which can be adjusted in consultation with the brand. However, since the prediction does only specify if a customer purchases in a certain period of time and not which brand they purchase, the brand specific commission rate can not be used. The company wide average commission rate of the last 6 months was used instead.

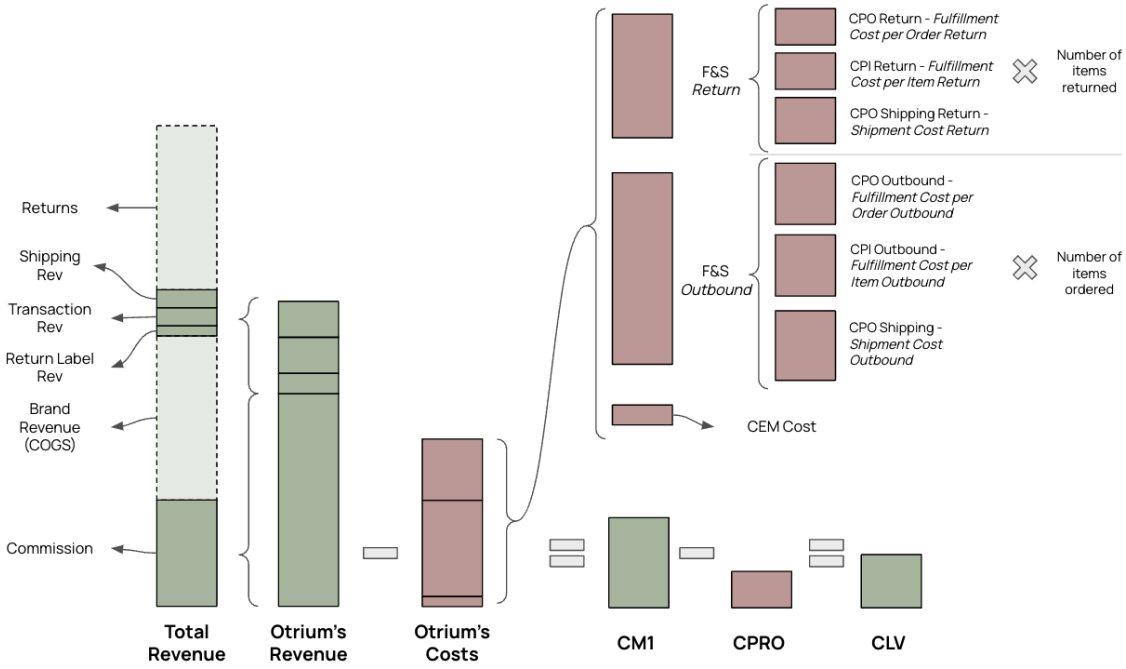


Figure 3.7: Otriums' CLV definition build-up

Applying the commission rate to the predicted net revenue gives Otriums' revenue for that customer for that period of time.

Fulfillment & shipping costs. There are also costs associated with an order, which are subtracted from Otrium's revenue.

First, there are fulfillment and shipping outbound (F&S Outbound) costs, which are all the costs related to an outgoing package. These costs include the costs associated with the shipping of the order (CPO Shipping), the packaging of an order (*CPO Outbound*) and the cost of packaging the different items within the order (*CPI Outbound* × *Number of items in the order*).

Therefore, the number of items in a future order had to be determined as well. This was done in the same manner as the prediction of the revenue using the company-wide average item price and the country of last 12 months. Dividing the predicted revenue by this average item price results in the predicted number of items.

Whenever something in the order is returned there are also fulfillment and shipping return (*F&S Return*) costs. Those include, again, the shipping costs of the return (*CPO Shipping Return*), any costs associated with handling the return whenever it is back at the

warehouse (*CPO Return*) and the costs per item in the return (*CPI Return* × *Number of returned items*). To obtain the returned number of items the net revenue was divided by the predicted average item price.

The costs for both F&S Outbound and F&S Return only differ slightly over time per country. Therefore, it was decided to take the values of the last available month per shipping country and use those for the prediction.

CEM costs. Otrium uses a company that takes care of the customer service for any complaints, so there are also costs associated to that (*CEM costs*). The value of these costs per order are determined every year, so the value of the last available year was used for the prediction and subtracted from the revenue.

Cost per retention. The last cost factor taken into account is the Cost Per Retention Order (*CPRO*), which are the costs related to the marketing actions taken in order to have customers make a repeat purchase. The average value of the last 6 months was used as a prediction of the upcoming months.

So in order to obtain the predicted CLV from the predicted net revenue, first the commission rate is applied to it after which the ancillary revenues were added. Subtracting the predicted costs from this resulted in the final CLV prediction.

3.5 Model Performance

In order to investigate and compare the performance of the various models a few performance measures were chosen. There are multiple components as to which the performance of the models can differ from each other.

First, the predictive performance of the total CLV over the whole customer base of the various models was measured using percentage deviation between *Forecast versus Actual* percentage (FvsA) [8]. The measure can be described as:

$$\text{FvsA} = \frac{\sum_{t=1}^{h} F_t - \sum_{t=1}^{h} A_t}{\sum_{t=1}^{h} A_t} \times 100\% \tag{3.1}$$

Where F_t is the total predicted CLV of all customers, A_t the total actual CLV in month t and h the horizon. Using this metric, it can be determined if the overall prediction of the various models is an over- or underprediction compared to the actual.

Second, the Mean Absolute Percentage Error (MAPE) was used for the assessment of the monthly CLV. The MAPE is computed by the following equation:

$$\text{MAPE} = \frac{1}{h}\sum_{t=1}^{h}\left|\frac{A_t - F_t}{A_t}\right| \tag{3.2}$$

Where A_t and F_t are, respectively, the actual monthly CLV in month i and the predicted monthly CLV in month i during the the h months in the test set.

The last performance metric focused on the customer specific prediction of CLV, for which the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used. The MAE is computed by:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|A_j - F_j| \tag{3.3}$$

Where A_j and F_j are, respectively, the actual CLV of customer j and the predicted CLV of customer j during the entire prediction period in the test set and n the number of customers. And the RMSE is computed by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(A_j - F_j)^2} \tag{3.4}$$

The two prediction models were also compared to two benchmark models. The first benchmark model used the actual CLV of a customer of a specific month as a prediction for that month a year later. The second benchmark predicted a CLV of zero for all customers in all months.

The most used metric is the 12 months CLV of a customer so the performance of the models were be measured based on that. Since it is also interesting to see how the model performed for predictions of a shorter period, the same metrics were computed for the first 3 months of the prediction.

4

Results and Discussion

In this section, the results of the prediction components mentioned above will be shown and discussed. Firstly, the performance of the two models in predicting the frequency and revenue per customer will be compared. Secondly, the implementation of the business rules in order to predict net revenue per customer will be investigated for both models. Lastly, the models will be assessed on their performance of predicting CLV.

Important to note is that for the comparison of the results, the customers that made their first purchase in the testing period were excluded. These customers have no historical purchase data, therefore the model will not be able to predict the CLV for these customers.

4.1 Frequency and Revenue Prediction

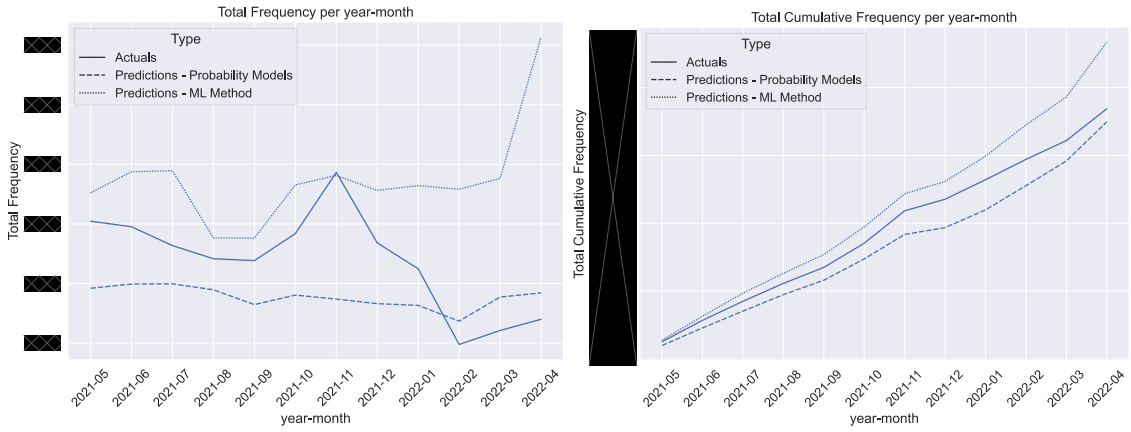
4.1.1 Frequency Prediction

This section compares the BG/NBD model with the Gradient Boosting (XGBoost) model for the prediction of transactions in the test set.

Figure 4.1(a) shows the monthly predicted frequency using the two models and the actuals and Figure 4.1(b) shows the cumulative frequency from the start to the end of the prediction period. These results show that the Probability models, with the BG/NBD model for the frequency prediction, overall underpredicts the actuals, except for the last 3 months of the test set for which the model seems to fit the actual frequency reasonably well. After 12 months the BG/NBD model had a percentage deviation between forecasted and actual frequency (FvsA) of -5.2%.

The XGBoost model slightly overpredicts the actuals. In contrary to the BG/NBD model, the XGBoost model also uses time-variant features, which gives it the ability to model seasonality. The results show that the model predicts the peak in transactions

around November and a dip after January. The largest difference between the prediction and the actuals occured after the dip in January 2022, where the model predicts that the orders trend upward again. However, the actuals show that the downward trend continues even further. Since the features only include data from before May 2021 and the drop in actuals most likely occured because of changes in the market with the ending of COVID, it makes sense that the model does not have the ability to predict that drop. The 12 month frequency percentage deviation between forecast and actuals for the XGBoost model was +26.6%.



(a) Monthly frequency prediction against the actuals.

(b) Cumulative monthly frequency in the prediction period.

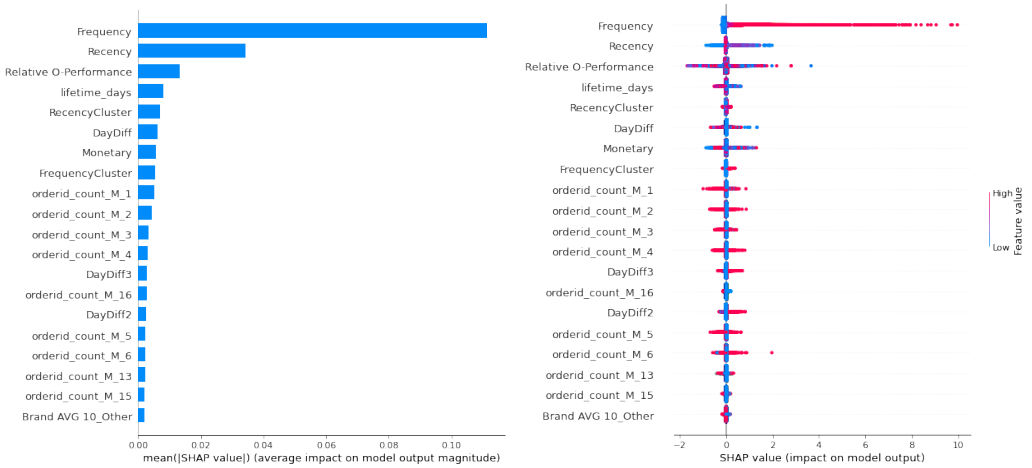
Figure 4.1: Monthly frequency predictions of the two models compared to the actual frequency in the 12 months of the test set using data from before May 2021 for the features.

The remainder of this section will be dedicated to a deep-dive into the feature importance and the tuning of the hyperparameters of the XGBoost model.

Feature importance. To investigate the relationships between the various variables with the output, it was chosen to look into their Shapley values. These values are defined as the (weighted) average of marginal contributions.

Figure 4.2(a) shows the top-20 features with the highest impact on the predicted frequency. This shows that the frequency, recency and relative performance are top-3 most important features. Figure 4.2(b) shows the Shapley values for the observations in the training set to give more details about the impact of the variables. The color of the observations represent the value of the feature, where red is a high value and blue a low value.

The horizontal location of the observation shows the SHAP value. This indicates whether the impact of the feature on the output is positive or negative. This shows that *frequency* has a high and positive impact on the predicted frequency. Indicating that more transactions made in the past by a customer result in a higher predicted frequency. This result corresponds with what was expected, since a customer with a lot of historical purchases is more likely to be a loyal customer and to buy again in the future.



(a) Feature importance of XGBoost features.

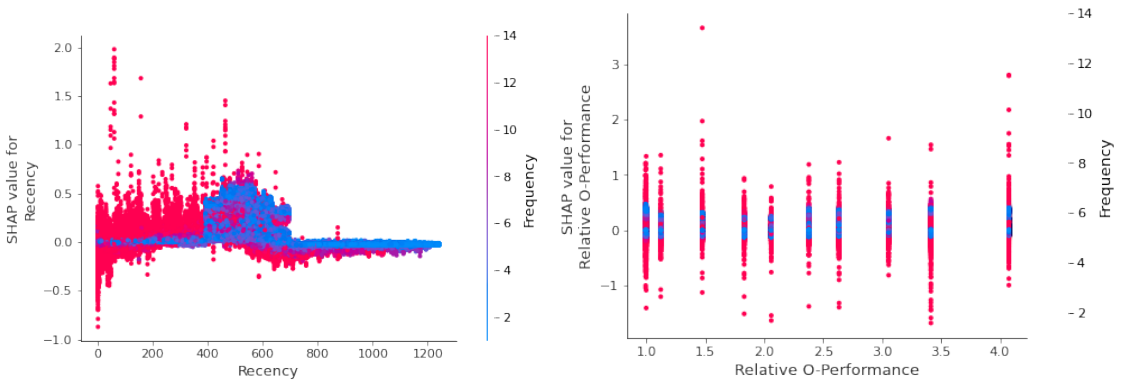
(b) Shapley values of XGBoost on the training set for the investigation of the feature importance.

Figure 4.2: Feature importance investigation using Shapley values.

To figure out the relationship between the *recency* and *relative O-performance* and the output, Figure 4.3 shows the dependence plots of these two features. This shows that the lower the recency value, so the more recent the last purchase was, the higher the predicted frequency (See Figure 4.3(a)). This was expected since a more recent purchase indicates that the customer has been active recently, while a high recency value can indicate that the customer has churned.

For the *relative O-performance* it would be expected that a higher value would result in a higher frequency to account for seasonality. However, no clear relationship can be distinguished from the plot (See Figure 4.3(b)).

Hyperparameter tuning. The *max_ depth* and *min_ child_ weight* were tuned using a Grid Search. The results are shown in Figure 6.1 in the Appendix. These plots show that



(a) Dependency plot of the recency to investigate the relationship with the predicted frequency.

(b) Dependency plot of the relative O-performance to investigate the relationship with the predicted frequency.

Figure 4.3: Feature importance investigation using shapley values.

the lowest MAE (0.117) is achieved with a value of 9 for *max_ depth* and a value of 4 for *min_ child_ weight*.

Using these two values, the optimal value for gamma was determined by trying out different values of gamma between 0 and 1 of which the results are shown in Table 4.1. This shows that the optimal MAE- value is achieved with gamma equal to 0.

Table 4.1: Comparison of different values of gamma using cross-validation

Gamma	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
MAE	**0.10687**	0.10723	0.10716	0.10714	0.10775	0.10730

Two other Grid Searches were executed to first find the optimal values for *subsample* and *colsample_ bytree* and subsequently for *alpha* and *lambda*. The results of these two Grid Searches are shown in Figures 6.2 and 6.3 in the Appendix. These figures show that the optimal value is achieved with the parameter *subsample* set to 1.0, *colsample_ bytree* to 1.0, *alpha* to 0.75 and *lambda* to 1.0.

Lastly, the optimal learning rate was determined by trying various values between 0 and 0.5 and comparing the MAE. The results in Table 4.2 show that the optimal value is given by a learning rate of 0.3.

Table 4.2: Comparison of different values of learning rate using cross validation

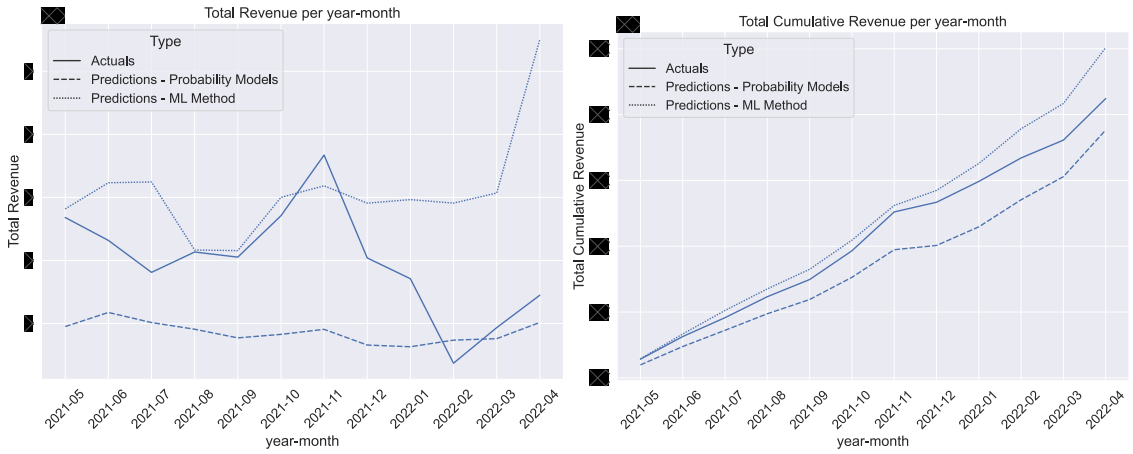
Learning rate	**0.01**	**0.1**	**0.2**	**0.3**	**0.4**
MAE	0.268	0.115	0.114	**0.107**	0.111

4.1.2 Revenue Prediction

This section compares the two methods to predict revenue using the Gamma-Gamma model and the business rules, described in Sections 3.2.1.2 and 3.2.2.2, respectively. Figure 4.4 shows the monthly results of the two methods in the testing period.

Since the correlation between the monetary value and the frequency was 0.038, the assumption about there being very little relation between the transaction process and and its value seemed questionable. However, for this research the relation was assumed negligible, allowing the use of Gamma-Gamma. When looking at the results of the Gamma-Gamma model used for the BG/NBD model, one can see that the model is heavily underpredicting. When comparing it to the frequency prediction results, it can clearly be seen that the performance decreased for this model. The 12 month FvsA for the revenue prediction with the probability models was equal to -11.3%. Which compared to the FvsA for the frequency prediction is indeed slightly worse.

In comparison, the cumulative revenue prediction for the XGBoost model in combination with business rules, shows that the prediction fits the actuals reasonably well. The FvsA improved from +31.0% from the frequency prediction to +18.3% after applying the business rules for the revenue prediction. However, although this improves the prediction performance for this case, it in general means that prediction of the order values using the business rules underpredict the actuals.



(a) Monthly revenue of the two models compared to the actuals.

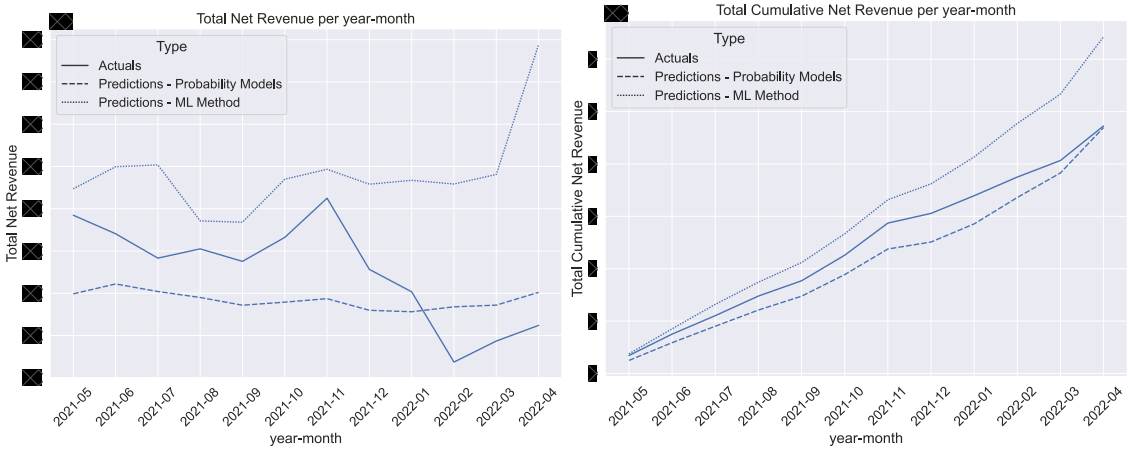
(b) Monthly cumulative revenue of the two models.

Figure 4.4: The monthly revenue predictions for the two models compared to actuals in the test set.

4.2 Net Revenue Prediction

Before looking at the final CLV model, the results of the business rules applied to predict the net revenue for both models are discussed. Figure 4.5 shows the results of these implementations. Comparing these results to the results from the revenue prediction, one can see that the business rules generally overpredict the net revenue. Meaning, the return rate defined by the business rules is lower than the actuals. Because of this overprediction, the underprediction of the Gamma-Gamma model in the BG/NBD model decreased to an FvsA of -0.6% while the overprediction of the XGBoost model slightly increased to an FvsA of +36.0%

A reason for this difference in return rates can be explained by the free return policy that was set in October 2021. When customers don not have to pay for returns anymore, they generally return more than when they do need to pay a fee. This resulted in a slight increase in average return rates across markets.



(a) Monthly Net Revenue prediction of the two models compared to the actuals.

(b) Cumulative Net Revenue prediction of the two models compared to the actuals.

Figure 4.5: The monthly net revenue predictions for the two models compared to actuals in the test set.

4.3 CLV Prediction

After predicting the net revenue per month per customer, the last step was to factor in the commission, ancillary revenues and various costs to obtain the predicted CLV. Table 4.3 shows the performance metrics of the two models along with those of the benchmark for both a short period of 3 months as the full prediction period of 12 months.

The two models are compared to two benchmark models. The first benchmark takes the CLV of a customer in a specific month as the CLV of that customer last year in that same month. The second benchmark predicts zero for all customers in every month for which no FvsA and MAPE are available. The results show that the second benchmark is performing pretty well and outperforms the probability models in both the short and the long term on customer level. The reason for this being that the majority of the customers in the test set places no order. Therefore, predicting zero for every customer is not that far off. However, the RMSE for the probability models is better than the benchmark, so we will look at a trade-off of MAE and RMSE.

Looking at the FvsA metric, it shows that on the short term the first benchmark model is performing better than the other two methods. The probability models performed slightly

better than the ML method, although both heavily underpredicting. For the long-term prediction both models are over predicting with the ML method performing significantly better than the probability models. And instead of underpredicting, they are both over-predicting in the long period prediction.

Looking at the performance of the models on the MAPE for the monthly CLV prediction comparison, it shows that the probability models outperformed the ML method for both the longer and the shorter period. The difference between the two is negligible when looking at the short period prediction, but for the long term prediction the MAPE for the probability models is significantly better.

The last two metrics look at the 3 and 12 month CLV on a customer level. This shows that the ML method outperforms the probability models in both prediction periods. Also, the probability models showed to perform worse than both of the benchmark models. Therefore, the probability models would not be recommended for the customer specific prediction of CLV.

Table 4.3: Performance metrics for the two frequency prediction models compared to the two benchmark models.

Period	Model	FvsA (%)	MAPE (%)	MAE	RMSE
Short period (3 months)	Benchmark - last year	- **17.7**	82.6	10.5	68.0
	Benchmark - all zero	- 100.0	-	8.0	74.2
	Probability method	- 35.2	**42.1**	11.1	53.0
	ML method	- 41.9	44.6	**7.3**	**52.9**
Long period (12 months)	Benchmark - last year	+ 155.3	443.8	44.2	559.9
	Benchmark - all zero	- 100.0	-	35.0	637.3
	Probability models	+ 58.7	**131.8**	45.0	700.8
	ML method	+ **26.1**	223.7	**27.8**	**567.3**

To summarize, the results show that for the short-term CLV prediction for the entire customer base, the benchmark model that uses the value for last year, outperforms the other two methods. The ML method outperforms the other methods when predicting the customer base CLV for 12 months in the future.

The results showed that the probability models outperformed the other models in both the long- and short-term prediction on a monthly basis.

The machine learning performed best when predicting the CLV per customer for both the short- and long-period prediction.

5

Conclusion and Future Research

Customer Lifetime Value is becoming increasingly important for various kind of businesses to base strategy decisions on. This leads to increased business value of predicting CLV per customer accurately. The aim of this research was to compare two prediction models on their predictive performance.

The prediction of CLV was divided into different components, where the frequency and revenue prediction were the components that differed for the two models. For these two components, it was decided to compare a probability prediction model, which has been used extensively in previous research, with a machine learning model. The latter was chosen because machine learning models are generally used most for prediction purposes. The models used were the BG/NBD model with the Gamma-Gamma model for revenue and the XGBoost model for frequency along with rules for revenue based on historical data.

The net revenue was predicted by applying business rules based on historical data to the revenue prediction obtained by the two models. Lastly, the costs associated with the predicted orders was factored in to obtain a CLV prediction per customer.

This section will summarize the most important conclusions per component and suggestions for future research in order to improve the various steps. Also, the performance of the final CLV models will be compared focussing on both a short and a long prediction period. The performance metrics used measure performance based on the entire customer base, monthly and per customer.

Machine Learning method. For the frequency prediction, it shows that the machine learning model is able to predict the seasonality more accurately and fits the monthly data

reasonably well. However, it does show that there is a small overprediction throughout the prediction period. This can be explained by the fact that the XGBoost model rarely predicts zero. So instead of zero, it will for example predict 0.0004. The sum of all customers with such a small partial order can result in an overprediction. To overcome this challenge it might be an option to use a binary classifier first for the classification of buying and non-buying customers as was done in [19]. Another option could be to predict CLV percentiles instead as described in [4].

Since the dataset was relatively small, the model was only trained on series that predicted 6 months in advance. So for future research it might improve the model to train on series that predict 12 months in advance, when the dataset allows it.

For future research, it might also be interesting to add some more features. For example, since Otrium uses a lot of promotion periods, it might improve the model to incorporate those to predict for possible higher frequencies. Also, it might be interesting to add more specifics about the activity of the customer on the platform such as product page views or sessions.

After applying the business rules, the revenue was predicted. The results show that the assumptions on average order value are generally underpredicting the actuals, since the overprediction that was seen in the frequency prediction slightly decreased in the revenue prediction resulting in a better fit. For future research, it might improve the result to use a prediction model for the order value as well.

Probability models. For the BG/NBD model together with the Gamma-Gamma model resulted in an underprediction of both the frequency and revenue. For this BG/NBD model, it was assumed that customers churn only directly after a purchase was made. However, in practice this is generally not the case, since a customer can churn because of numerous reasons at any point in time. So for future research, the Pareto/NBD model might be a better fit.

Also, the assumption that the purchase rate and the monetary value are not correlated might not be entirely true in Otriums' case. So for future research, the Pareto/Dependent model introduced in [14] might be a better fit.

Return rate prediction. The results for net revenue showed that the number of returns is larger than predicted, resulting in an overprediction of net revenue. Most likely, this is because of the free return policy that was implemented in October 2021. For future

research this has to be taken into consideration to more accurately define business rules for return rates per country and gender.

Cost prediction. For the prediction of the various costs, any free shipping coupons were not taken into consideration. However, there is a policy at Otrium where the customer has free shipping, whenever an order is above €150. Since the order value is also predicted, the free shipping coupons can also be taken into account for the prediction of the final CLV.

CLV prediction. Looking at the performance on final CLV, the results show that the machine learning model performs better when looking at the MAE and RMSE for both the short and long period prediction. The probability model performs better on a monthly basis in both periods, which was not expected because the BG/NBD model does not account for seasonality. For the entire customer base the XGBoost method outperforms the BG/NBD in the long period prediction, but performs worse for the short period prediction.

When looking at the results of the 12 month CLV of the entire customer base, one can see that there is a significant difference between the actual and the predicted CLV after 12 months for both models. However, it should be considered that the model has to learn from data that happened 12 months in advance. Therefore, the chances of such a model performing perfectly are small because of external factors. An example is the ending of COVID in the beginning of 2022, this resulted in a dip in actual sales. The prediction models however are not able to predict such a dip based on data from the year before, resulting in an overprediction of the data.

Also, it should be considered that there are a lot of customers that only placed one order in the feature set. This makes it difficult to perfectly predict the customer level CLV for those customers, since there is not much known about their behaviour.

Recommendation. Looking at the results, both the probability models (BG/NBD with Gamma-Gamma) and machine learning method (XGBoost with business rules) have areas in which they outperform the other models. Depending on use of CLV, the best method can be chosen. Since Otrium will mostly use the 12 month CLV grouped by various metrics such as most ordered country or brand, it would be recommended to use the XGBoost method for this.

6

Appendix

The figures below show the grid search performed to tune the hyperparameters of the XGBoost model.

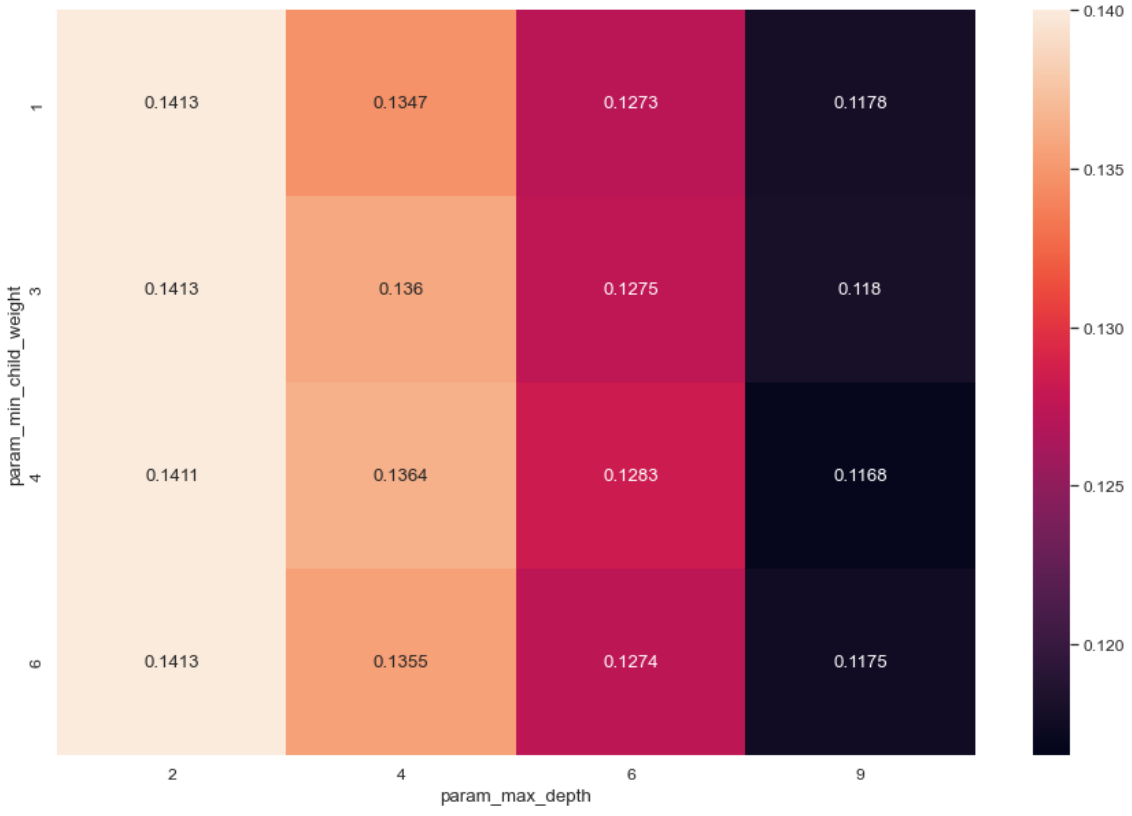


Figure 6.1: Grid search result for the parameters *max_ depth and min_ child_ weight with MAE scoring*

Figure 6.2: Grid search for *subsample* and *colsample_ bytree*

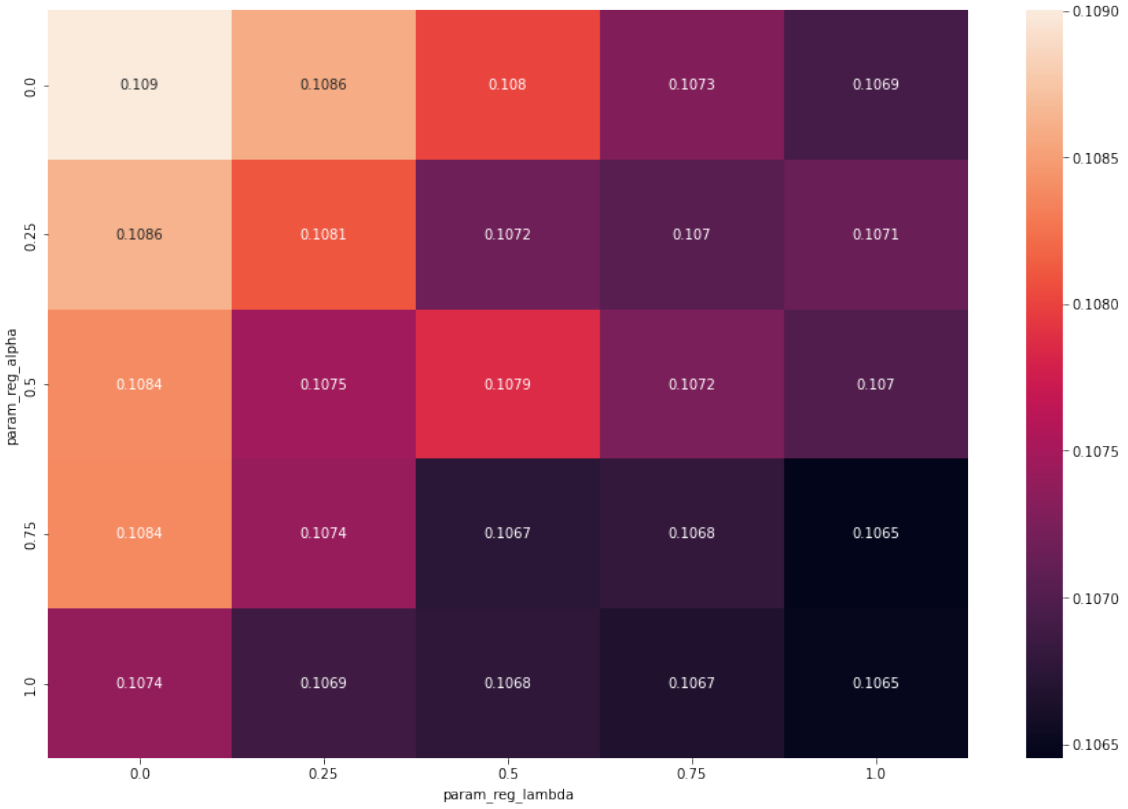


Figure 6.3: Grid search for *lambda* and *alpha*

References

[1] Bemmaor, A. & Glady, N. (2012). Modeling purchasing behavior with sudden "death": A flexible customer lifetime model. *Management Science*, **58**, 1012–1021. 7

[2] Berger, P.D. & Nasr, N.I. (1998). Customer lifetime value: Marketing models and applications. *Journal of Interactive Marketing*, **12**, 17–30. 6

[3] Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**, 123–140. 18

[4] Chamberlain, B.P., Cardoso, A., Liu, C.B., Pagliari, R. & 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 '17, 1753–1762, Association for Computing Machinery, New York, NY, USA. 5, 6, 7, 38

[5] Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, **abs/1603.02754**. 17

[6] Colombo, R. & Jiang, W. (1999). A stochastic rfm model. *Journal of Interactive Marketing*, **13**, 2–12. 7

[7] David C. Schmittlein, D.G.M. & Colombo, R. (1987). "Counting Your Customers: Who Are They and What Will They Do Next? *Management Science , Jan., 1987, Vol. 33, No. 1, pp. 1-24*. 6, 13

[8] Donkers, B., Verhoef, P. & de Jong, M. (2007). Modeling clv: A test of competing models in the insurance industry. *Qme-Quantitative marketing and economics*, **5**, 163–190. 5, 8, 26

[9] Drachen, A., Pastor, M., Liu, A., Fontaine, D.J., Chang, Y., Runge, J., Sifa, R. & Klabjan, D. (2018). To be or not to be...social: Incorporating simple social features in mobile game customer lifetime value predictions. In *Proceedings of*

the Australasian Computer Science Week Multiconference, ACSW '18, Association for Computing Machinery, New York, NY, USA. 7

[10] Fader, P., Hardie, B. & Lee, K. (2005). Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research American Marketing Association ISSN*, **XLII**, 415–430. 7, 15

[11] Fader, P., Hardie, B. & Lee, K. (2005). "counting your customers" the easy way: An alternative to the pareto/nbd model. *Marketing Science*, **24**, 275–284. 6, 13, 14, 15

[12] Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, **29**, 1189 – 1232. 17

[13] Frisbie, G.A. (1980). Ehrenberg's negative binomial model applied to grocery store trips. *Journal of Marketing Research*, **17**, 385–390. 6

[14] Glady, N., Baesens, B. & Croux, C. (2009). A modified pareto/nbd approach for predicting customer lifetime value. *Expert Systems with Applications*, **36**, 2062–2071. 7, 8, 38

[15] Heitz, C., Dettling, M. & Ruckstuhl, A. (2011). Modelling customer lifetime value in contractual settings. *IJSTM*, **16**, 172–190. 5

[16] Jašek, P., Vraná, L., Sperkova, L., Smutny, Z. & Kobulsky, M. (2018). Modeling and application of customer lifetime value in online retail. *Informatics*, **5**. 7, 8

[17] Paauwe, P., Putten, P. & Wezel, M. (2007). Dtmc: An actionable e-customer lifetime value model based on markov chains and decision trees. vol. 258, 253–262. 7

[18] Pfeifer, P.E. & Carraway, R.L. (2000). Modeling customer relationships as markov chains. *Journal of Interactive Marketing*, **14**, 43–55. 7

[19] Vanderveld, A., Pandey, A., Han, A. & Parekh, R. (2016). An engagement-based customer lifetime value system for e-commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 293–302, Association for Computing Machinery, New York, NY, USA. 5, 6, 7, 38