An advice to improve the short-term return forecast

MASTER THESIS AT BOL.COM

UNIVERSITY OF TWENTE

VERSION 1.0

Author:
M. Maljaars (Moniek)

Examination committee

Dr. Engin Topan	*University of Twente*
Dr. C.G.M. Groothuis - Oudshoorn	*University of Twente*
Joost W. Miltenburg MSc	*Bol.com*

Educational Institution
University of Twente
Faculty of Behavioral Management and Social Sciences
Department of Industrial Engineering and Business Information Systems

Educational Program
MSc. Industrial Engineering and Management
Specialization: Production and Logistics Management
Orientation: Supply Chain and Transportation Management

August, 2020

bol.com

UNIVERSITY OF TWENTE.

Management Summary

Bol.com has expanded significantly in the recent years. However, the increasing sales also enlarges the number of items that are returned. Currently, Bol.com is facing difficulties with the short-term return forecast. A lack of a separate short-term return forecast and a lack of unified storage data causes difficulties to match the workforce and the actual work on a daily basis and may lead to an inefficient process that results in high additional costs, dissatisfied employees and a negative effect on customer service.

The purpose of this thesis is to develop a model that forecasts the number of returns for the short-term, in order to improve the accuracy and efficiency at the warehouse. The proposed forecast is based on the return requests that are stored in a database called Boomerang, in which customers register their returns on the website. The planning window of the forecast is 26 days. The proposed forecasting method incorporates two complementary models to predict the total number of returns per day. The first model classifies whether a return request will be returned and the second model predicts the timing between the registration and the actual return. Together, they provide a prediction of the total number of returns per day. All discussed models were validated using a 5-fold Cross-Validation.

For the first model, the classification of whether a return request will be returned is based on product characteristics of the return, time aspects and reason codes. Based on the literature review performed in this research, Logistic Regression and Random Forest are found to be the most appropriate methods for this purpose. Using the Recursive Feature Elimination Cross-Validation, we are able to apply these models using the ten most important features to predict the outcome of the response variable. The performance of the models is measured using a classification report containing the precision, recall, F1-score, the AUC score and the confusion matrix. Based on the results, we can conclude that the accuracy of the model is quite high, but the model is poor at predicting the true negatives which leads to an overestimation of the number of returned requests. The differences between the Logistic Regression and Random Forest with all and only ten features are small. The Random Forest model performs slightly better than the Logistic Regression model. Although, the Logistic Regression method is preferred due to the higher interpretability of the model. Using these models, we found the following explanatory features to be important for determining the total number of returns:

- Positive effect: *price, sources of registration, selling parties* and almost all *reason codes*.
- Negative effect: *hour of registration, day of the week, quantity*, reason codes *delivery too late* and *no reason provided*.

These results extend the findings in the literature for the time effects and combination of features.

The second model determines the timing of the return request, based on product characteristics, time aspects and reason codes. There is evidence in the literature that the LASSO Regression provides solid results to forecast returns. However, the LASSO Regression did not provide satisfying results in our research, which is indicated by a low R-squared value of 6% combined with a low Root Mean Square Error. Because the timing of a return is count data with a positive skew and non-negative numbers, we use also Poisson Regression and Negative Binomial Regression to get more promising results. The performance of the Poisson and Negative Binomial Regression is found to be much higher compared to LASSO, with a R-squared value of 25% for both models. Based on the AIC values, the Negative Binomial Regression shows a better fit of the model. Although, since Poisson Regression requires less

parameter estimation and updating, both models were tested in the prediction of the total number of returns per day. Based on the outcome of the models, we found the following explanatory features to be important for determining the timing of returns:

- Positive effect: *hour of registration, sources of registration, selling parties* and *reason codes*.
- Negative effect: *day of the week* and *quantity*.

Based on the literature, less research about the important features is conducted regarding the timing of a return. The only comparison is the non-significant importance of the price and the significant importance of reason codes. The reason codes that positively influence the timing of the return the most are *delivery too late*, *wrong article received* and *no reason provided*.

The output of the Logistic Regression is used as an input for the Poisson and Negative Binomial Regression. For each positively classified return request, the timing is determined. In this way, the total number of predicted return requests per day is calculated. However, due to direct returns without registration, this number is increased using a day of the week and month specific percentage. The overall performance of the short-term return forecast is measured using Mean Absolute Percentage Error (MAPE). The proposed forecasting model using Logistic Regression and Poisson Regression reduce the current MAPE of 15.1% to 13.3%. Using Negative Binomial Regression, the MAPE reduces to 13.5%. In both cases, the overall performance of the short-term return forecast increases.

Although Bol.com requests a short-term return forecast on item level, we also test the models with the aggregate of the return requests per day, instead of per request. The main goal of this aggregation is to show the predictive power of the models when adding additional data and to show an alternative modelling choice by using aggregate returns instead of a prediction on item level. From these observations, we see that although the number of days between the registration of the return and the processing of the return reaches 26 days, our findings with aggregate measures show that a possible resource planning based on aggregate measures does not necessitate a planning window of 26 days, 11 days would be sufficient. This decrease in planning window leads to a major increase in the R-Squared value from 25% to 56% of the Poisson Regression model and in the overall performance of the MAPE from 15.1% to 11.2%.

To conclude the findings of this research, we advise Bol.com to implement the proposed forecasting model based on Logistic Regression for the classification and Poisson Regression for the timing. The proposed method significantly increases the performance of the return forecast. Based on the aggregate results, we strongly advise Bol.com to integrate additional data which decreases the planning window. Integrating the transport data would decrease the planning window from 26 to 5 days, which we believe will have a major positive impact on the accuracy of the forecast. Based on the current dataset, we advise Bol.com to keep track of the individual items but use the aggregate forecast.

We recommend Bol.com to improve the accuracy of the model by increasing the number of explanatory features. Currently, the model predicts customer behavior without any personal information regarding the customer. Product characteristics, time aspects and reason codes are the only criteria of the explanatory features. We believe that adding additional information regarding the customer, their past behavior and the transport process would have a positive impact on the accuracy of the return forecast. Furthermore, the predictions of direct returns, the weekends and the aggregate of the return requests could be investigated in more detail.

Preface

With this thesis, I finish my master Industrial Engineering & Management at the University of Twente. Finishing my master thesis marks the end of my time as a student, which was a wonderful time with many accomplishments and experiences. I look forward to my next adventure.

In this preface, I would like to take the opportunity to express my gratitude to the people who helped me realizing this thesis. All members of the committee helped accomplishing this success. First of all, I would like to thank Joost for the opportunity and guidance of conducting my thesis at Bol.com. Furthermore, I would like to thank Engin and Karin for their valuable feedback and support to bring my thesis to a higher level. You made this thesis as it is today.

I would also like to thank my family and close friends for their encouragement and the good times we spent together. In particular, I would like to thank my parents and Jens for their support, understanding, and love.

I hope you enjoy reading this thesis!

Moniek Maljaars

Utrecht, August 2020

Table of Contents

1. Introduction

The purpose of this master thesis is to provide Bol.com advice on the short-term return forecast. Ingram Micro is also an important stakeholder, since they organize the logistic process at the warehouse. They are both eager to increase the accuracy of the short-term return forecast. In this chapter we discuss the organizational context, the problem statement, the research goal and finally the scope of this research.

1.1 Organizational context

Bol.com is founded in 1999 by the German multinational Bertelsmann. They started as the first online bookstore, selling 140,000 different types of books. In 2012, Bol.com became part of Ahold and sold products in several categories. Nowadays, Bol.com has more than 22 million articles in more than 40 product categories with 10.5 million active clients from Belgium and the Netherlands. On average, Bol.com has more than 7000 visits per minute (Bol.com, 2020).

This research is conducted at Bol.com at the logistics-MaX department in a team dedicated to the Outbound & Returns processes. The organizational chart is visualized in Figure 1.1.

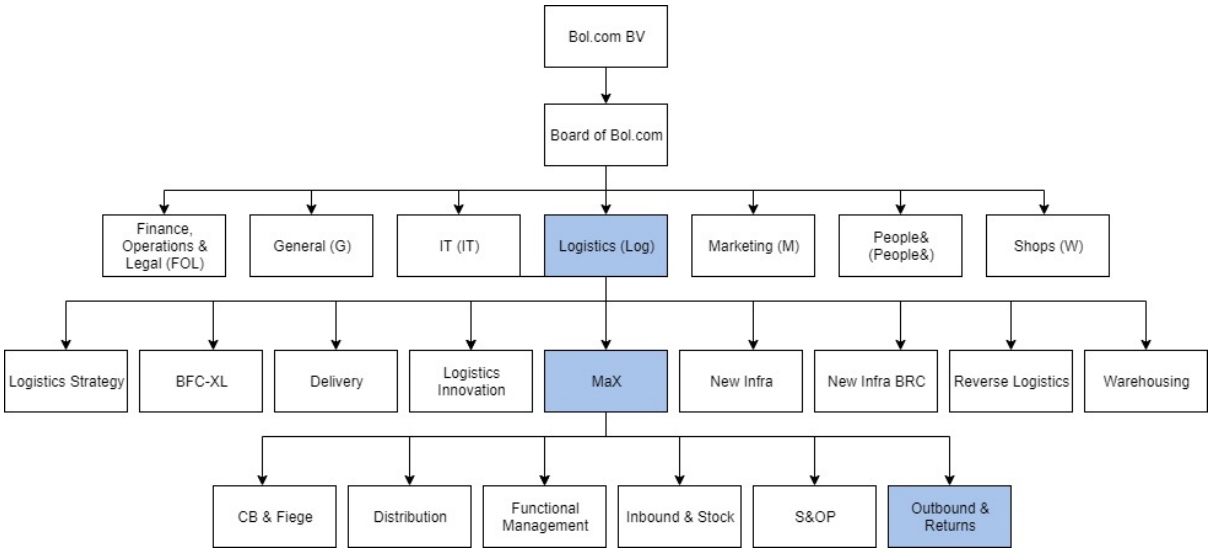


Figure 1.1: Organizational chart Bol.com.

Bol.com has three different streams included in their processes. Those contain:

- Own products: products that are owned, stored and delivered by Bol.com.
- Plaza Logistics via Bol.com (LvB): products from partners, but stored and delivered by Bol.com.
- Plaza without LvB: products from partners, which are stored and delivered by the partners.

This research is based on the own- and Plaza LvB-products. Plaza without LvB-products are excluded from this research, since those products are not returned to the warehouses from Bol.com. This research focuses on the warehouse in Waalwijk at the Veerweg, where Ingram Micro arranges the workforce.

1.1.1 Ingram Micro

Ingram Micro (IM) is one of the main logistics providers worldwide that deals with the logistics of several webshops. They have a workforce of around 21,800 to give partners the appropriate service. IM represents around 1700 suppliers worldwide (Micro, 2020). One of its biggest partners in the Netherlands is Bol.com. Therefore, close cooperation is required between Bol.com and IM. IM arranges the logistic processes at the warehouse in Waalwijk.

1.1.2 Warehouse operation

The Supply Chain of Bol.com consists of several suppliers, warehouses, transporters and customers. This research focuses only on the return process at the warehouse at the Veerweg in Waalwijk. Figure 1.2 visualizes both the forward flow and the return process at the warehouse. The forward flow is represented as follows: products from own suppliers and Plaza LvB are send to the warehouse and are input for the inbound process of the warehouse. Subsequently, products are stocked and prepared for the outbound process. The products are either send to Pick Up Points (PUP) or directly to the customers. The return process always starts with the request of a customer. The product is returned to the PUP by the customer, or directly send to the PostNL sorting center by select-members. Select-members pay an extra fee and receive extra services in return. From the PUP, the product is either send by PostNL and then send to the sorting center or the product is sent by BPost. In both cases, the products are returned to the warehouse.

Bol.com has arrangements with several transporters for their own and Plaza LvB products. The most important transporters are PostNL, Dynalogic, BPost, RedjePakketje and PartsExpress. PostNL delivers the largest part of the products in the Netherlands and also in Belgium. Dynalogic is mainly responsible for the large and heavy products. BPost is only transporting in Belgium and lastly both PartsExpress and RedjePakketje are responsible for the 'same-day' delivery. For the return process, PostNL and BPost are the main transporters and are in this research considered as the only transporters for the returns. The other transporters are excluded for the remainder of this research.

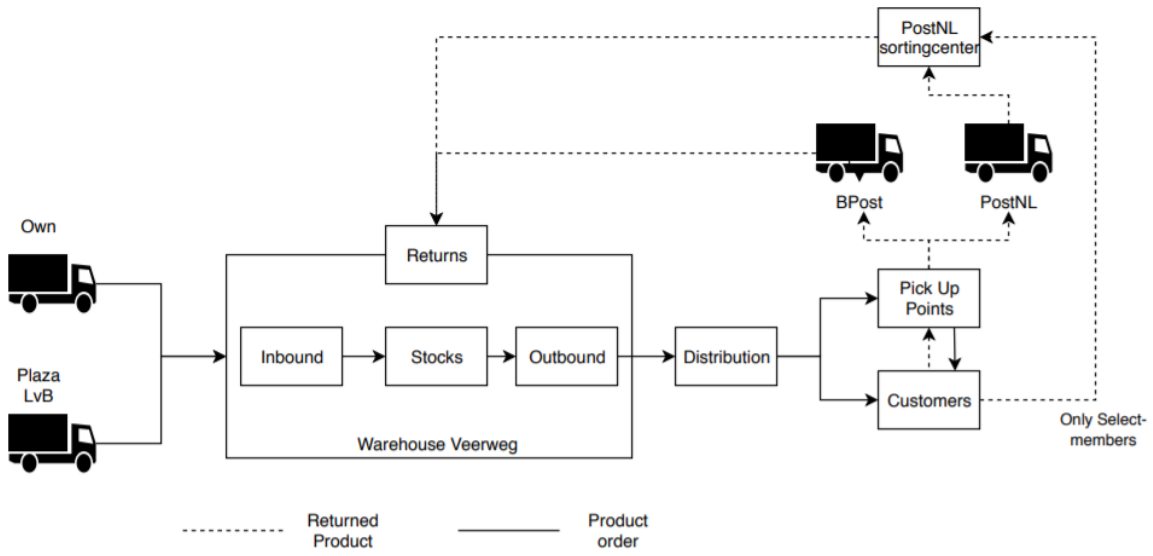


Figure 1.2: Process of the warehouse at the Veerweg.

The customers of Bol.com are 10.5 million active clients from the Netherlands and Belgium. Whether the forward flow is free of charge for the customer depends on the product and a minimum order. The return process, on the other hand is always free for Bol.com's own and LvB Plaza's products.

In general, there will be no deliveries in the weekend from the transporters to the Veerweg. That is why, there will be no returns processed in the weekends at the Veerweg.

1.1.3 Return forecast

Sales data is often integrated in the return forecast. Because the time between selling a product and the actual delivery is not equal, the expected deliveries are taken as an input for the return forecast instead of the sales forecast. This forecasted delivery data is referred to as the *hold data*. Which implies information that is based on the physical delivery instead of the online sales of the product. Currently, Bol.com makes a long-term return forecast for the entire year based on hold data forecast and return percentages. Bol.com assumes that using the hold data instead of the sales data increases the accuracy of the return time forecast.

The mid-term return forecast consists of an eight-week forecast, which is equal to the long-term return forecast, adjusted with the actual hold data. This eight-week return forecast is updated every week for the remaining weeks. This weekly update is equal to the short-term return forecast. However, there is no clear distinction in the data between the mid- and short-term return forecast, because both are updated equally. The only difference between de mid-term and short-term forecast is the forecast window. The short-term forecast is only for one week, compared to the mid-term forecast of eight weeks. We do not investigate the mid-term return forecast separately, since the outcome is equal to the short-term return forecast.

1.2 Problem statement

Bol.com has grown significantly in the recent years and is still growing. Figure 1.3 shows the increasing sales of the last three years. Due to confidential regulations, the exact numbers are excluded from this report. The increasing sales puts more pressure on the existing resources. Compared to 2018, the number of returns increased approximately 35%. Product returns present one of the largest operational challenges in internet retailing, which is due to the volume and cost of returns (Mollenkopf, Rabinovich, Laseter, & Boyer, 2007). Forecasting return logistics is more difficult than forward logistics, since more uncertainty is involved in terms of quantity, time and quality of the returned product (Flapper, 1995).

Bol.com indicates that the long-term return forecast is good enough, while the short-term return forecast is not. Furthermore, they encounter problems regarding the dissatisfaction of employees of Ingram Micro and of the customers. Petersen & Kumar (2009) state that the return process is part of the post purchase-experience and herein influences customer satisfaction and retention. Furthermore, higher costs are visible due to the varying workloads and return lead times. The following section elaborates on the problem formulation.

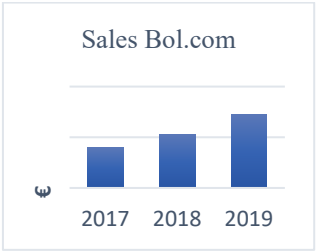


Figure 1.3: Growing sales Bol.com

1.2.1 Problem cluster

In order to investigate what can be improved on the short-term return forecast, a problem cluster is created to identify the cause and effect relationships that lead to the core problem(s). Determining the core problem is useful for identifying the action problem, which is defined as the result of the reality that differs from the norm (Heerkens & Van Winden, 2012). Figure 1.4 visualizes the problem cluster associated with the return forecast at the warehouse. Three action problems are identified together with Bol.com. First, dissatisfied employees of Ingram Micro are identified as an action problem. Second, high costs that are related to the varying workloads and return lead times. Those longer return lead times will also lead to dissatisfied customers as a third action problem. The root-causes of those action problems are visualized in the problem cluster.

One core problem that is identified using the problem cluster is the **lack of a separate short-term return forecast**. Currently, the short-term return forecast is equal to the weekly updated mid-term return forecast, which is called the 8-week planning at Bol.com. Because there are no adjustments to the short-term forecast compared to the mid-term forecast, the daily return forecast is currently based on the weekly demand multiplied by a fixed day index, which is only revised at most quarterly. This revision is not performed each quarter and sometimes this index is only updated once a year. This fixed multiplier index is explained in more detail in Section 2.1. Furthermore, the short-term return forecast is only updated once a week with actual data.

Another core problem that contributes to this gap is the **lack of an unified storage of data** regarding the return process. Several data can be useful for the short-term return forecast, which will be explained in Section 2.3. Because there is no central storage, information regarding the registered returns and information from the PUP as well as information from PostNL is not included in the current return forecast, as will also be explained in Section 2.3. Therefore, the short-term forecast is not adjusted with this extra information, with inefficient capacity use as a result.

Both core problems contribute to the mismatch between the workforce and the actual work on a daily basis. Because the actual hold data is only updated once a week in the short-term return forecast, varying workloads as well as varying return lead times are a result. The return lead times vary because the return forecast is currently inaccurate. On the one hand, the return lead times are influenced by the workforce and on the other hand by the Work in Progress (WIP) at the warehouse. Bol.com currently incorporates a high WIP at the warehouse to cope with overestimated days, to have enough work for the workforce. Human effort is needed in the return process at the warehouse, because packages are wrapped, sorted and investigated by humans. As a result, the return lead time would be longer if the return forecast is underestimated and shorter if overestimated due to the planned workforce. The internal longer lead times of Bol.com influence the customer return lead time, because the customers' money is only returned after the return is processed at the warehouse. The longer lead times can lead to dissatisfied customers but in addition high costs because of the high WIP, which involve stocking costs. Next to this, the varying workloads result in dissatisfied employees if their workload and job varies each day.

The current short-term capacity of the warehouses is used inefficiently for processing returns due to short-term forecast models which are not updated frequently and do not incorporate alternative recent source of data. This results in high costs and dissatisfied employees and even customers.

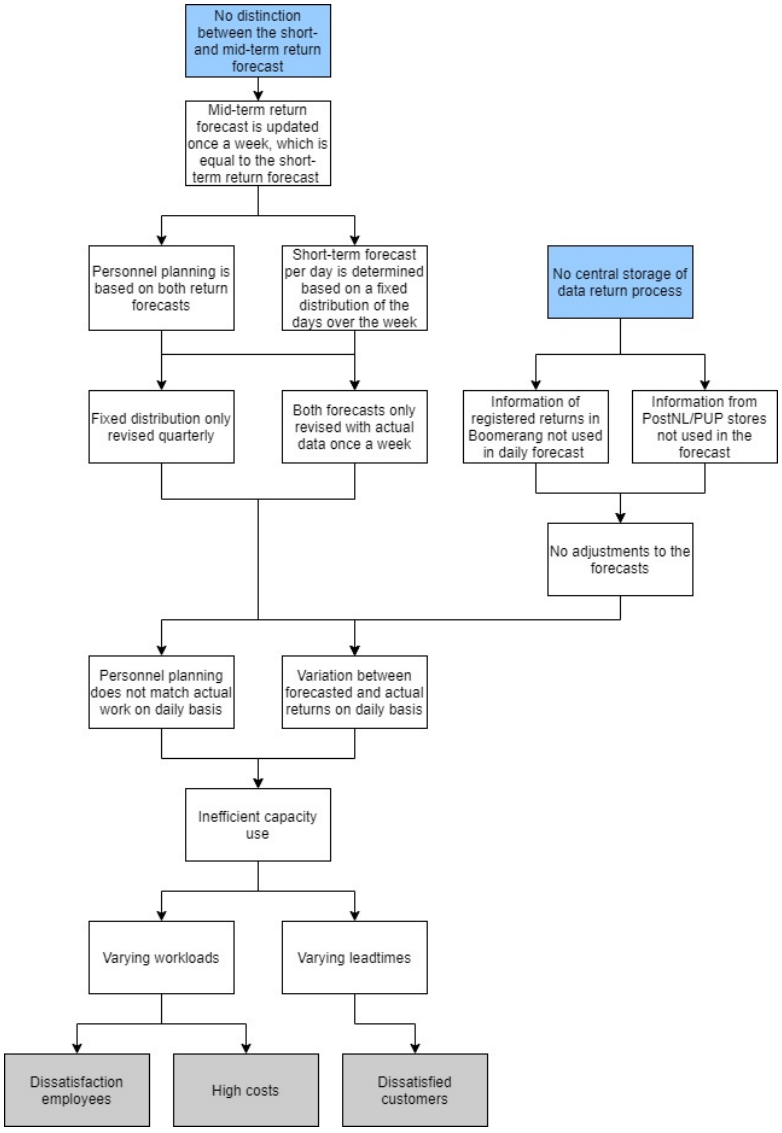


Figure 1.4: Problem cluster.

1.3 Research goal

1.3.1 Main research goal

Bol.com desires to increase the accuracy of the daily short-term return forecast. The two core problems described in the previous section are likely to be the cause of the variation between the forecasted and actual number of returns on short-term. However, designing a central storage of data is out of scope. Despite, additional data will be integrated in the short-term return forecast to increase the accuracy. Because the mid-term and short-term return forecast are updated equally and not stored separately, there is no clear distinction between them in the data. Therefore, with evaluating the current short-term return forecast, we indirectly evaluate the mid-term return forecast as well. That is why we do not investigate the mid-term return forecast individually, as will be visible in the research questions. The weekly updated short-term forecast is not accurate and leads to inefficient capacity use. Therefore, a model should be developed that is updated each day and incorporates additional data regarding the return

process. Bol.com wishes a forecast based on item level, to gain more insight in the product mix of the arrived returns at the warehouse. The research goal is formulated as follows:

'*Develop a model that forecasts the number of return items for the short-term to improve the accuracy at the warehouse, which leads to a better efficiency and satisfied personnel and clients'*

This research goal results in the following main research question:

'*How should the short-term return forecast-model for Bol.com be constructed, such that the difference between the forecasted and actual number of return items on daily basis is mitigated?'*

The aim of the research is to answer this question, using insights in the following aspects:

- Analysis of the current way of forecasting the demand side of the returns.
- Provide insight in the existing forecasting methods for returns (on daily basis), as described in the literature.
- Data-analysis of the available data regarding the return forecast and managing returns.
- Developing and implementing a model that reduces the variation between the forecasted and actual number of return items.

1.3.2 Research questions

The following research questions are formulated to answer the central research question mentioned in the previous section. The different research questions are divided on chapter basis.

Research question 1: What does the current forecast of the returns look like? -*Chapter 2*

1.1) What is the difference in long and short-term return forecast, and how are they performed at Bol.com?
1.2) What is the current performance of the long and short-term return forecast?
  1.2.1) Evaluated per week and day?
  1.2.2) Evaluated per month and weekday?
1.3) Which data from Bol.com is available that could be used by a return forecasting model?
  1.3.1) How can the Boomerang data be used?

Research question 2: Which methods are described in available literature regarding the (daily) forecast of the number of returns? -*Chapter 3*

2.1) Which methods for short-term forecasting are proposed in the literature?
2.2) How to identify and minimize overfitting?
2.3) Which methods for return forecasting were used in the literature and are suitable for our research?

Research question 3: How can we develop a short-term return forecast that produces more accurate results? -*Chapter 4*

3.1) How can this method be put into a model to increase the accuracy for the given input data?
3.2) How to collect, process, analyze and synthesize the data for the inputs of the model?

Research question 4: How should the model be validated? -*Chapter 5*

4.1) How well does the proposed forecasting method perform?

4.2) How can the model be validated and verified?

Research question 5: How should the model be implemented at Bol.com? -*Chapter 6*

5.1) How should Bol.com implement the new return forecast?

5.2) What is needed to implement the new return forecast-model?

Finally, the conclusions and recommendations are presented in Chapter 7.

1.4 Research approach

The core problems are already defined in Section 1.2.1. Only the core problem regarding the lack of a separate short-term return forecast is considered. For the problem-solving approach, a model to forecast the demand of returns for the next day(s) should be developed. The uncertainty of the current return forecast influences the workloads, lead times and costs. The result of this research is a short-term return forecast model for Bol.com, which is also used as an input by Ingram Micro for the workforce planning. The forecasting model should predict the number of return items for the next day with a higher accuracy. The request from Bol.com is to develop a forecasting model on item level, which can be used to predict not only the number of returns, but for example also the number of returns per product group.

1.5 Scope

This research is based on the return process of the own and LvB Plaza-products of Bol.com. This implies that the Plaza non-LvB products are excluded from this research. Only the warehouse at the Veerweg in Waalwijk is considered in this research. The forecast window will be short-term and should be updated on a daily basis to increase the accuracy of the forecast for the next day.

As explained above, the current short-term planning is just a simple function of the mid-term planning, which creates confusion in distinguishing the terms. Therefore, we do not investigate the mid and short-term return forecasts individually. The remainder of this paper will therefore refer to this weekly updated return forecast as the short-term forecast and the mid-term forecast will be left out of this research. The workforce planning and the core problem regarding the central data storage are out of scope. Furthermore, since the forecast window is short, sales are not included in the forecast-model. The literature review is restricted to the most widely known quantitative forecasting methods and machine learning methods.

The forecast should be on item level as requested by Bol.com to gain inside in the product mix of the forecasted returns. This is requested due to the arrival of a new warehouse in which the returns are partly processed automatically for specific product groups. The return forecast will be based on the registered returns. However, in practice, some returns are not registered and directly send to the transporters. Those returns are called *direct returns* and should be considered in the research for Bol.com to implement the forecast. However, the direct return forecast is not the major goal of this research and should be considered as an approximation and needs more concern in further research.

2. Analysis of the current situation

To evaluate whether the proposed return forecasting method increases the accuracy, the current forecast should be thoroughly investigated. Therefore, we describe the current return forecast in Section 2.1 and evaluate the current forecasting method in Section 2.2. However, we do not only look at the current short-term return forecasting method, but also at the long-term return forecasting method, since Bol.com states that the long-term forecasting method performs good in contrast to the short-term return forecast. We evaluate both forecasting methods per day and per week and search for data patterns for the weekdays and months. This chapter answers thereby the first research question: '*What does the current forecast of the returns look like?'.*

2.1 Current return forecast

First, we investigate the current process of forecasting the number of return items, which is visualized in Figure 2.1. As mentioned before, the return forecast is made by using forward hold-data. Based on actual data from previous year, the long-term sales forecast is created. This sales demand forecast is made by the department of Sales & Operations Planning (S&OP) for the entire year. From this long-term return forecast of one year, the sales forecast in items is retrieved. However, sales are not immediately sent to the customers. Therefore, the data of sending the product to the customer is used, namely the hold-data. The aggregate hold data forecast is disaggregated over the six different clusters Bol.com uses, namely:

- Sport, style and baby;
- House and garden;
- Electronics;
- Daily care and animals;
- Toys and entertainment;
- Reading and learning.

Furthermore, the percentage of the actual return of last year is also disaggregated over those clusters. This percentage together with the item forecast per cluster determines the mid-term forecast of the number of return items per week and day. However, the return forecast per day is adjusted. The day forecast of each week is summed, and this represents the week forecast. The return forecast per day is however adjusted by multiplying weekly demand by a day index, which is fixed. They try to update this index multiplier each quarter, but this is not always the case. Sometimes this index is not even updated each year.

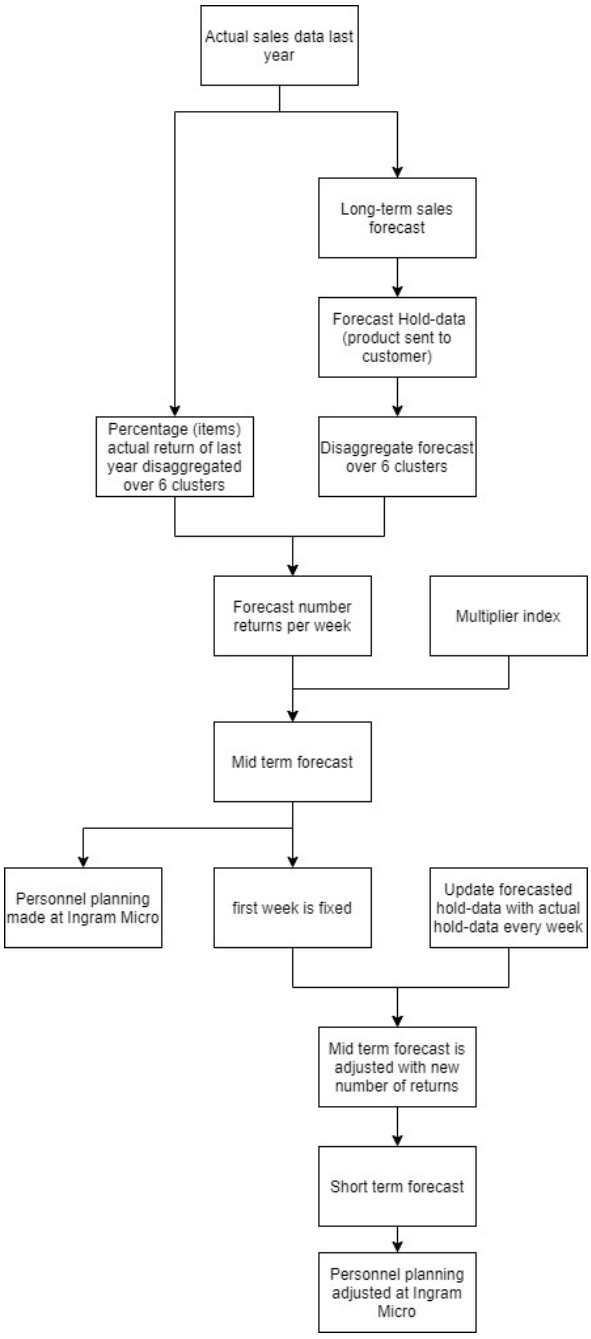


Figure 2.1: Return forecasting process.

The mid-term return forecast is created every eight weeks. Currently, there is no distinction between the mid-term and short-term return forecast. The weekly updated mid-term return forecast is equal to the short-term return forecast. Bol.com makes no distinction in the data between those forecasts, because both are updated equally and not stored separately. As mentioned before, this is the reason why we do not evaluate the mid-term and short-term separately, since the outcome would be equal. Because our aim is to increase the accuracy of the short-term return forecast, we investigate the current short-term return forecast instead of the mid-term return forecast. Therefore, the mid-term return forecast is excluded from the remainder of this research.

2.2 Current performance

We use the accuracy as a measurement to analyze the current performance. Currently, Bol.com uses different measurements to determine the accuracy. Therefore, we propose to use the Mean Absolute Percentage Error (MAPE) and Mean Absolute Deviation (MAD) measurements as indicators for the accuracy of the forecasts. The MAPE does not meet the validity criteria due to the distribution skewness to the right, but is probably the most widely goodness-of-fit measurement (Moreno, Pol, Abad, & Blasco, 2013), (Kim & Kim, 2016). In contrast to the MAPE, the MAD has the absence of bias in method selection and is suitable for series with intermittent and near-zero demands (Kolassa & Schütz, 2007). In this research, we rely on the values of the MAPE, since the values of the MAD are confidential. We analyze the current performance based on the long and short-term return forecast.

2.2.1 Long-term return forecasting

The first created version of the long-term return forecast is taken as the actual data input for the analysis of the demand return forecast. This is only applicable to 2019, since the long-term return forecast of 2018 was adjusted to the short-term return forecast and not stored separately. Table 2.1 visualizes the MAPE per week and per day for the long-term demand return forecast. The MAPE values show that weekly return forecasts are better. Therefore, we can assume that the day return forecast deviates more from the actual number of returns than the return forecast per week and has a lower accuracy for 2019. This is in line with the experience from Bol.com.

Long-term	**MAPE**
Per week	0.093
Per day	0.154

Table 2.1: Accuracy results long-term return forecast 2019.

The averages of the MAPE of the year return forecast per day and week in 2019 are shown in Figure 2.2 and Figure 2.3 respectively. From the figures, the peak moments are visible, namely parts of January, May, September, November and December. Especially January, November and December are the months with the highest sales. From the figures, we can see that the deviation is also high during those peak moments. Which can be explained by the higher deviation in sales forecast or by a changing return percentage in those peak months. Based on the results, we cannot exclude seasonality and should take the difference per month into account for the return forecast.

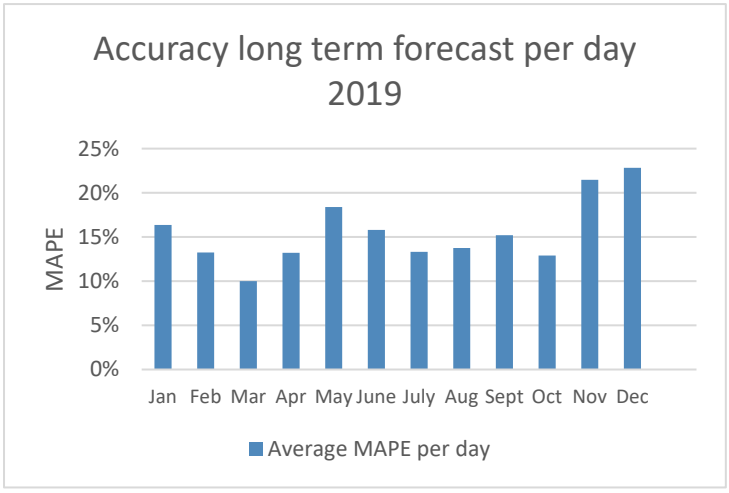


Figure 2.2: Result long-term return forecast per day.

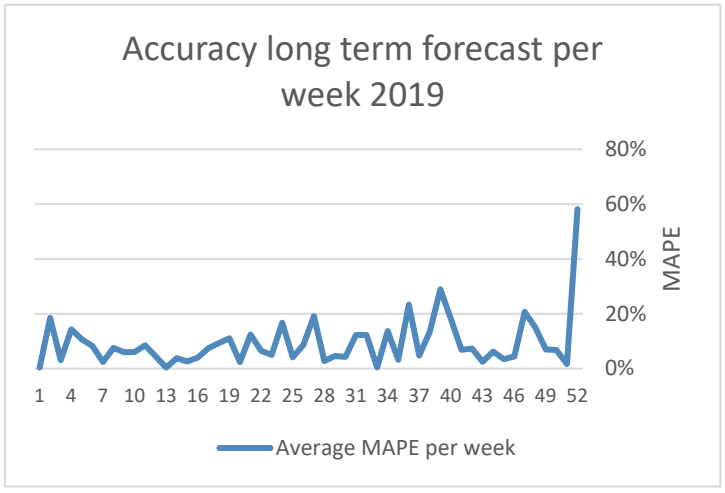


Figure 2.3: Result long-term return forecast per week.

2.2.2 Short-term return forecasting

For the short-term return forecasting, both data from 2018 and 2019 are available. We did not add the data of previous years, since those would not represent the current situation due to the large increase in sales and difference in return percentages. The accuracies per week and day for the short-term return forecast of 2018 and 2019 are shown in Table 2.2. From the results we can conclude that the forecast had a higher accuracy in 2018. The MAPE of 2019 per day is 20.98% higher compared to 2018 and the MAD 54.13% higher. Figure 2.4 visualizes the accuracy of the short-term return forecast per week for 2018 and 2019. Figure 2.5 zooms in on the difference between the MAPE for 2018 and 2019. However, we cannot conclude a relation from the figures between the forecasting errors of 2018 and 2019. Figures 2.4 and 2.5 show differences in the MAPE per week and month for 2018 and 2019. The differences per week do not follow a clear pattern for both years. Therefore, week numbers and the year could have an impact on the return forecast.

Short-term	2018	2019
Per week	0.073	0.091
Per day	0.125	0.151

Table 2.2: MAPE result short-term return forecast 2018 and 2019

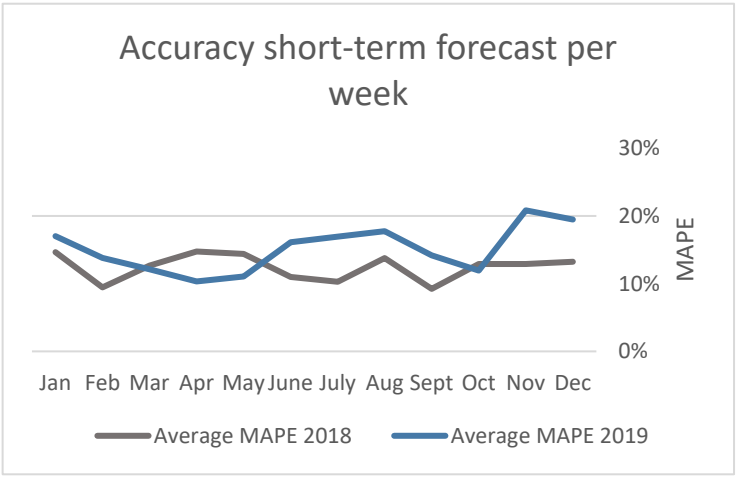


Figure 2.4: Result short-term per week for each month.

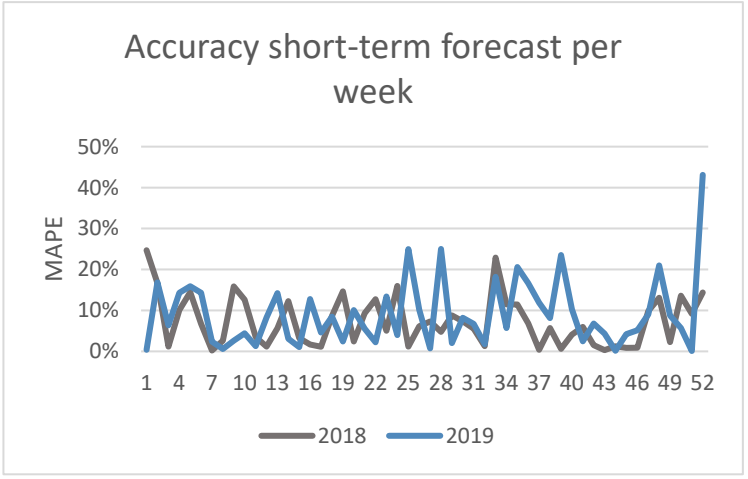


Figure 2.5: Result short-term return forecast per week.

2.2.3 Long versus short-term return forecasting

We analyze whether the short-term return forecast has a higher accuracy compared to the long-term return forecast. Tables 2.1 and 2.2 show the differences. The accuracy of the short-term return forecast is 2.08% higher per week and 2.33% higher per day. Although, the increase in accuracy is small, we investigate this difference in more detail. Figures 2.6 and 2.7 represent the MAPE of the long and short-term return forecast per day of 2019. The difference is on the x-axis, where the figures represent respectively the months and the week numbers. Around May, the accuracy difference is the largest. In the months April, May, September, October, November and December, the long-term return forecast adjustment to short-term return forecast increased the accuracy. In the other months, this was not the case. However, from the results that are shown, we can conclude that the difference between the total average MAPE of the long and short-term return forecast is only 0,2 percent point per week and 0,3 percent point per day, from which we conclude that this difference is small. In order to see the differences in more detail, and to analyze whether data-patterns are visible, we analyze the differences per month and per weekday in the next sections.

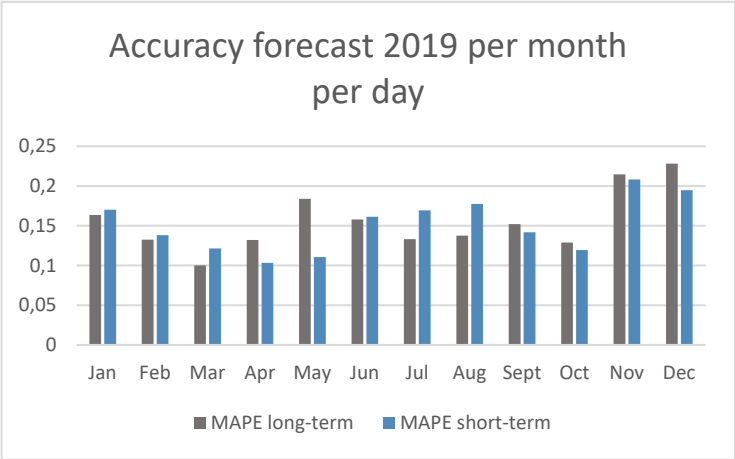


Figure 2.6: Accuracy long vs. short-term forecast per month.

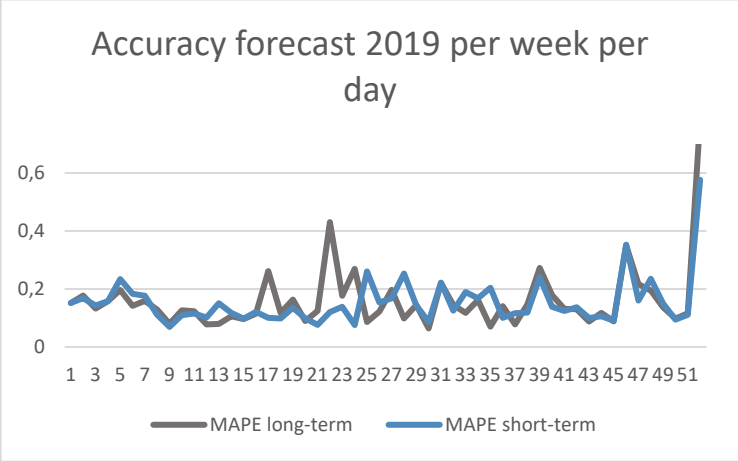


Figure 2.7: Accuracy long vs. short-term forecast per week.

2.2.4 Performance per month

Since the total average has minor difference, we also analyze the over- and underestimation of each day for the long and short-term return forecast of 2019. We investigate whether data-patterns are visible during the months. Table 2.3 shows the total over- and underestimated number of returned items of 2019 for the short and long-term return forecast. Based on those results, we can conclude that the long-term return forecast is rather underestimated than overestimated and the opposite holds for the short-term return forecast. This could be explained by the reaction of the short-term return forecast to the long-term return forecast. An underestimation is noted during the weeks, and the forecast is adjusted with a higher forecast, but this adjustment happens later than it actually occurs, which results in an overestimation. The interaction between the long and short-term return forecast per month is visualized in Appendix A.

	Short-term	**Long-term**
Overestimated	9.19%	6.01%
Underestimated	5.89%	9.82%

Table 2.3: Results long and short-term return forecast per month over- and underestimation 2019.

From Appendix A and Table 2.4, we can conclude that the short-term return forecast is not correctly adjusted to the deviation of the long-term return forecast. Figure 2.8 shows the total deviation per month. It differs per month whether the return short-term forecast has a smaller interval or not. From the results, we cannot indicate a specific month which is always under- or overestimated. However, as mentioned before, we should take the different months into account for the return forecast. In the next section, we look closer at the return forecast per weekday.

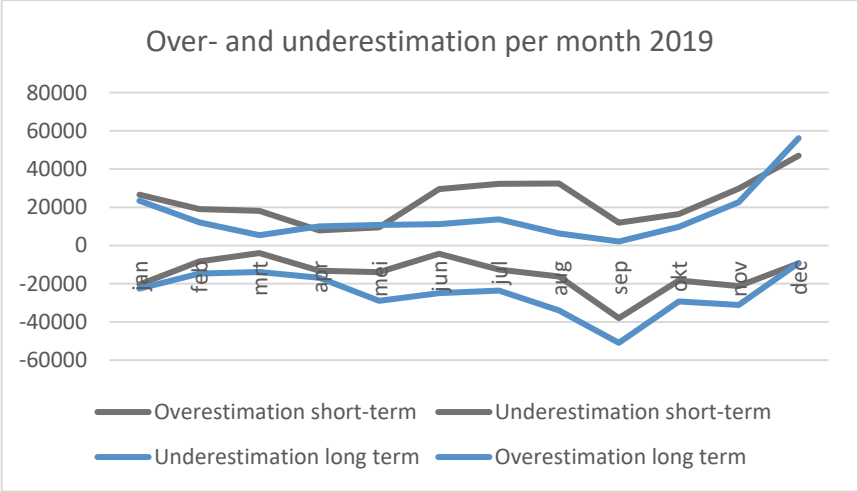


Figure 2.8: Over- and underestimation per month for long and short-term return forecast 2019.

2.2.5 Performance per weekday

Since the current return forecast incorporates a fixed multiplier index of the days over the week, we analyze the results also on weekdays to look for data-patterns. The deviation between the forecasted and actual number of returns for each weekday is determined in Appendix B. From the figures in Appendix B, we see that Monday is often overestimated in both the long and short-term return forecast. On the other hand, Wednesday is often underestimated in both forecasts. The total over- and underestimation per weekday is shown in Table 2.4. This over- and underestimation of the weekdays can be due to the day index with fixed multiplication of weekly demand as explained before. Table 2.5 visualizes the actual and forecasted percentages per weekday as well as the percentual difference. From those results, we can conclude that the return forecast of Monday was on average too optimistic and Wednesday on average too pessimistic in 2019.

	Short-term					**Long-term**				
	Monday	Tuesday	Wednesday	Thursday	Friday	Monday	Tuesday	Wednesday	Thursday	Friday
Overestimated	**17,69%**	8,36%	1,74%	5,38%	**13,17%**	**12,79%**	5,82%	0,41%	5,38%	8,90%
Underestimated	0,88%	3,41%	**11,08%**	7,23%	5,32%	1,44%	6,71%	**17,68%**	7,23%	8,67%

Table 2.4: Results over- and underestimation of the return forecasts per weekday 2019.

The day index used in the fixed multiplication of weekly demand is unequal to the actual day index. Therefore, we investigate whether this day index comes from the actual index of 2018. Further, we verify our observations based on the data of 2018. To investigate whether those observations are not a result of randomness. The total over- and underestimation per weekday of 2018 is shown in Table 2.6. From those results we cannot see a structural over- or underestimation per weekday. However, we can see that the return forecast on Wednesday is almost never overestimated.

	Actual Percentage	Forecast	%Difference
Monday	20.04%	23.80%	18.74%
Tuesday	20.66%	20.90%	1.17%
Wednesday	20.56%	17.70%	-13.93%
Thursday	19.46%	18.47%	-5.11%
Friday	19.27%	19.13%	-0.73%

Table 2.5: Percentages of number of returns per weekday 2019.

	8-week				
	Monday	Tuesday	Wednesday	Thursday	Friday
Overestimated	1.61%	1.28%	0.23%	0.47%	0.80%
Underestimated	0.94%	1.25%	2.20%	2.58%	1.32%

Table 2.6: Result over- and underestimation of the return forecasts per weekday 2018.

The actual return day index and the percentual difference compared to 2019 are given in Table 2.7. Based on these results, we would advise Bol.com to use the actual day index of the previous year as the fixed day index in the next year. The percentual difference was lower than 1.28% for each day, which is better than the current estimation of 2019. Based on the results we should also take the weekday into consideration for the return forecast model.

	Actual Percentage	%Difference 2019
Monday	19.64%	0.41%
Tuesday	20.80%	-0.14%
Wednesday	20.24%	0.32%
Thursday	20.74%	-1.28%
Friday	18.58%	0.69%

Table 2.7: Percentages of number of returns per weekday 2018.

2.2.6 Conclusion current performance

From the data we can conclude that the MAPE of the long-term return forecast and short-term return forecast only differs 0.3 percent point on daily basis and 0.2 percent point on weekly basis. Therefore, the adjustments of the short-term forecast do not have the desired impact on the performance of the forecast. The MAPE on daily basis is around 15%, compared to 9% of the weekly forecast for the short-term as well as the long-term forecast. The higher deviation on daily basis is partly due to an incorrect disaggregation of the returns over the days.

The forecasted multiplier index of the days over the week is overestimated on Monday and underestimated on Wednesday for 2019. If the actual multiplier index of 2018 was taken as the index for 2019 instead of the estimated index, the accuracy would be higher. Therefore, we advise Bol.com to use the indexes of the previous year as the current index multipliers. However, we do not investigate those index multipliers in more detail in this research. Despite, we will increase the accuracy of the return forecast by integrating the weekdays.

Based on the results, we cannot exclude seasonality for the return forecast. The following variables can have an impact on the return forecast model: year, month, week number and weekday.

2.3 Dataset

Bol.com has a large database including several datasets that can be used for the short-term return forecast. Therefore, available data is analyzed in this section. Currently, Bol.com uses Tableau to visualize information regarding the return processes. Data from Tableau is retrieved from BigQuery, which is a web service that enables interactive analysis of massive datasets. BigQuery is used to help convert big data into informed business decisions. The raw data is retrieved from BigQuery.

The current short-term return forecast is updated once a week. Every week, the actual hold-data and return percentage of the past week are included in the short-term return forecast. Figure 2.16 shows the process of the demand side of a return from the customer.

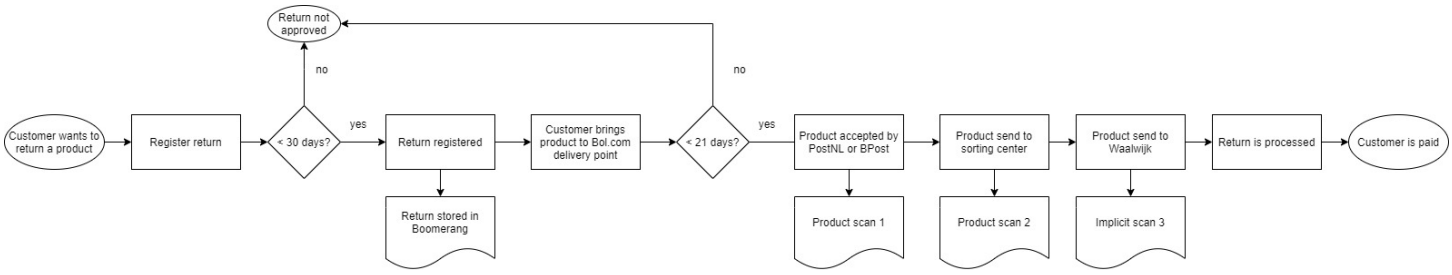


Figure 2.9: Process of a customer return to the Veerweg.

There are several datasets currently unused in the short-term return forecast. We indicate the main uncertainties and opportunities below, followed by a conclusion for each bullet whether we use this information, or leave it out of scope:

- *In most cases, the return is registered in Boomerang.* The customer registers the return at the website and if the item is registered within 30 days, the return is approved. This approval is registered in a database called Boomerang. However, the registration in Boomerang is not deleted if the customer cancels the return.
  - ✓ The registration of the return in Boomerang will be used as the starting point of our return forecast.
- *In other cases, the return is not registered in Boomerang.* In some cases, the customer does not register the return and sends it directly back to the warehouse, which is referred to as a *direct return*. This is not the regular way but happens in some cases. If there is no registration in Boomerang and the return is received at the warehouse, the warehouse registers the return in Boomerang. Therefore, the timing between the registration and processing is equal to zero.
  - ✓ The return forecast model should include direct returns.
- *The registered return should be returned to a PUP within 21 days.* Once the customer requests the return, the registered return should be returned to a Pick Up Point (PUP) within 21 days. If this requirement is not fulfilled, the return is not approved by PostNL or BPost. For the select members as shown in Figure 1.2, it is not necessarily to bring the item to a PUP, but can be send to PostNL directly. However, this contains only a small percentage of the total number of items and has no impact on the data from Boomerang.
  - ✓ The maximum return time after registration of 21 days is used as a constraint in our model.
- *A first product scan is performed once the item is returned to the PUP.* The approval happens during the scan of the product. Therefore, this is the second time that data is available regarding returned items.

- This information is useful for Bol.com but is out of scope for our research, due to unreliable data storage.

- *Return is sent to the sorting center.* PostNL and BPost send the returns to the sorting center, where the items are scanned for the second time.
  - This information is out of scope for our research, also due to unreliable data storage.
- *There is no scan performed at the warehouse in Waalwijk.* After the items are sorted, the returns are delivered to the warehouse in Waalwijk. However, there is no scan performed in Waalwijk. We would strongly advise to implement this scan, to improve the accuracy of the estimated number of returns that arrived at Waalwijk and to enlarge the insights of the return process.
  - This information is out of scope for our research.
- *The number of received returns is estimated.* Because there is no scan at the warehouse, the actual number of returned items per day is unknown. The number of processed returns per day is known, but this is unequal to the received items. Currently, the number of items is estimated using a fixed number of items within a package, multiplied by a fixed number of items on a roll container. This fixed number of items on a roll container varies during the seasons, due to different sizes of seasonal products. Once the item is processed, the item is registered. However, due to a high Work-in-Progress (WIP), this number of processed items is not equivalent to the actual number of items that entered the distribution center on that day.
  - ✓ This problem is a major drawback of the research and decreases the accuracy. The model will be based on the processed returns per day.
- *The Track & Trace code of the customer is an estimation.* PostNL uses Track& Trace codes for the returned items for the customers. This Track& Trace code is not exact, because the code is based on the scan of the returned items to the PUP plus three extra days. PostNL assumes that once the product is accepted by a PUP, the product is at least returned to Waalwijk within three days.
  - This information is out of scope for our research.
- *Received information is not a perfect information.* The return lead time, so the time between a return request and the arrival at the warehouse, depends on many other aspects. For example, it depends on the external parties, such as PostNL and BPost, but also on the customer. The customer can return the item within 21 days or keep the item. Those uncertainties imply that the exact timing of a return is unknown, which means that the received information is not a perfect information. Therefore, forecasting effort is still required.
  - ✓ Because of imperfect information, we will use a forecasting model to determine the number of returns.
- *The short-term return forecast will rely on the data from Boomerang.* Since the current data from the product scans of PostNL and BPost is inaccurate, the model will not rely upon those product scans, but only on the Boomerang data. Since most returns are registered in Boomerang, the short-term return forecast should not necessarily rely upon the sales data. Therefore, data from Boomerang replaces the input of the sales data.
  - ✓ We will only use the Boomerang data for our return forecast model.
- *The number of days between the registration of the return and the processing is not certain and varies between 1 and 26 days.* The maximum of 26 is determined due to a promised maximum return lead time of Bol.com to the customers of 5 days plus the maximum return time of 21 days. However, sometimes this maximum return lead time is not met, since customer service can give permission to the customer to return the item. This contains only a small percentage of the total amount of returns and is left out of scope.

- ✓ The actual return of a return request varies between 1 and 26 days, which is a constraint for the return forecast model.

- *Not all registered returns in Boomerang will be returned by the customer*. Despite all returns are registered in Boomerang, not all registrations are actually returned. Because cancellations or delays are not deleted from Boomerang. Therefore, forecasting the return percentage is still required.
  - ✓ Therefore, an additional model should be developed to determine whether a registered return will be returned.

To summarize, the most important information is given in Table 2.8. Based on the information stated above, two different aspects should be covered in the proposed model:

1. Classify whether the return request will actually be returned.
2. Forecast the timing between registration and arrival at the warehouse.

The two forecasting models will be based on the Boomerang data. Boomerang is the database which stores every return registration. All characteristics of the concerned return registration are registered and useable for data analysis. Some examples of characteristics of the return requests are for example the shop group, price, quantity and reason of the return. More details regarding the used dataset are explained in Chapter 4.

REGISTRATION IN BOOMERANG	Starting point
FORECAST WINDOW	Return request needs to be returned within 26 days
PROCESSED RETURNS	Timing of return will be based on the day of processing, which can deviate from the actual return date.

Table 2.8: Summary of data information.

2.3.1 Analysis of the Boomerang data

The distribution of the duration from the Boomerang dataset of 2019 is shown in Figure 2.10. The duration in days between the return registration and processing is stacked. From the figure we can conclude that the distribution is positively skewed to the right for each month individually, but also together. Even if the zero values are not included, the distribution is still positively skewed as shown in Figure 2.11.

The zero values in the dataset represent returns without registration, which are called the *direct returns* as described in Section 1.5. If the return is processed at the warehouse without registration, the registration will be done at the warehouse. Hence, the timing between registration and processing will be zero. These zero values should be forecasted upfront and should be left out of the registered return forecast. The zero values represent around 9% of the data, which is shown in Table 2.9. These direct returns are mainly a result of an error in the smart returns system of Ingram Micro, by which some items cannot be read correctly and therefore not connected to the associated customer. Because the zero values represent around 9 % of the data, we cannot ignore this data. We adjust the following two aspects that should be covered in the models:

1. The zero values should not be incorporated in the timing of registered returns.

2. An additional prediction of direct returns should be calculated upfront and added to the total forecasted number of returns.

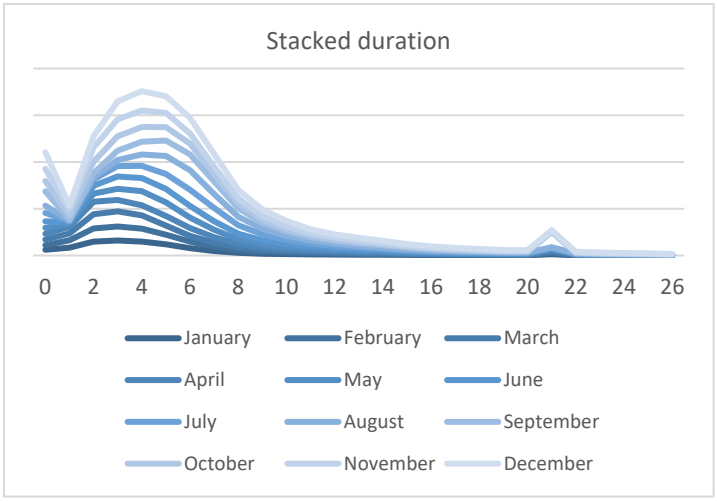


Figure 2.10: Distribution duration.

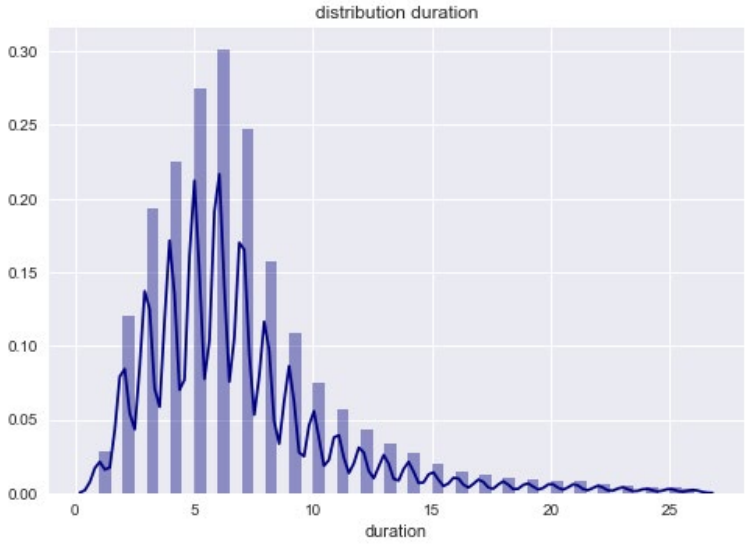


Figure 2.11: Distribution of timing window 1-26 days.

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	AVERAGE
JANUARY	11.21%	8.31%	9.74%	9.33%	9.94%	5.98%	4.47%	**8.43%**
FEBRUARY	8.68%	9.47%	8.17%	8.67%	7.90%	2.97%	2.57%	**6.92%**
MARCH	9.74%	9.44%	10.00%	9.70%	9.00%	2.92%	3.00%	**7.69%**
APRIL	7.82%	9.03%	10.15%	9.58%	9.15%	2.40%	3.06%	**7.31%**
MAY	8.43%	9.29%	9.23%	8.41%	9.53%	2.68%	2.38%	**7.13%**
JUNE	7.72%	9.63%	10.18%	12.02%	10.93%	3.89%	3.27%	**8.23%**
JULY	10.13%	10.29%	9.95%	10.64%	10.49%	4.41%	3.66%	**8.51%**
AUGUST	10.22%	10.29%	9.90%	10.41%	10.79%	5.66%	3.69%	**8.71%**
SEPTEMBER	11.05%	11.44%	10.37%	12.12%	12.85%	5.61%	3.29%	**9.53%**
OCTOBER	13.93%	13.52%	12.04%	13.92%	13.53%	3.81%	3.66%	**10.63%**
NOVEMBER	11.97%	14.02%	14.37%	14.78%	14.04%	3.14%	2.97%	**10.75%**
DECEMBER	15.10%	14.56%	12.06%	10.89%	14.36%	3.19%	2.95%	**10.44%**
AVERAGE	**10.50%**	**10.77%**	**10.51%**	**10.87%**	**11.04%**	**3.89%**	**3.25%**	**8.69%**
MAX DEVIATION	**4.6%**	**3.8%**	**3.9%**	**3.9%**	**3.3%**	**2.1%**	**1.2%**	**2.1%**

Table 2.9 Result average percentage zero values per weekday.

Based on the results of Table 2.9, a prediction of the zero value percentage could be made. Since the averages of the average percentages per weekday deviate, a prediction per weekday is required. Furthermore, the maximum deviation is 4.6%. Hence, a prediction per month is needed for a sophisticated prediction of the zero values.

The impact of the hour of registration on the zero values is shown in Table 2.10. We cannot see a clear pattern from this table for the registration hour. Each month, the percentage of zero values differ a lot per hour. For example, the percentage of zero values at 11:00 pm, is 17.79% in January, compared to

2.59% in February. Therefore, the percentages of zero values are not stable for each month. In addition, from the averages we can see that the hour of registration influences the zero value percentages and should therefore be taken into consideration.

Registration hour	January	February	March	April	May	June	July	August	September	October	November	December	Average
0	5.24%	4.82%	3.99%	4.46%	5.62%	8.62%	6.68%	7.28%	6.67%	8.78%	5.79%	10.70%	**6.55%**
1	6.18%	7.09%	2.54%	5.75%	3.10%	5.15%	5.75%	5.13%	7.24%	9.24%	5.51%	6.56%	**5.77%**
2	7.97%	2.74%	4.94%	5.62%	5.73%	11.07%	5.33%	10.83%	8.10%	6.06%	6.00%	4.43%	**6.57%**
3	6.79%	6.01%	2.42%	1.91%	4.15%	6.02%	3.91%	6.49%	3.61%	12.16%	6.01%	6.67%	**5.51%**
4	8.50%	5.81%	4.94%	3.85%	3.26%	3.17%	3.59%	3.89%	4.31%	4.71%	4.58%	4.07%	**4.56%**
5	4.81%	2.63%	2.64%	2.69%	4.83%	6.13%	6.55%	4.14%	5.58%	20.74%	3.41%	3.24%	**5.62%**
6	3.21%	2.87%	3.01%	10.18%	9.50%	13.44%	14.00%	11.02%	13.42%	21.38%	9.70%	14.02%	**10.48%**
7	11.25%	3.37%	5.25%	14.31%	12.82%	15.19%	15.10%	16.96%	17.07%	14.46%	19.91%	21.00%	**13.89%**
8	11.51%	11.94%	11.35%	10.03%	11.03%	13.04%	12.76%	11.72%	13.10%	15.23%	13.80%	10.82%	**12.19%**
9	12.36%	11.23%	12.21%	14.87%	12.88%	12.07%	12.13%	13.49%	14.26%	12.59%	13.53%	11.71%	**12.78%**
10	9.47%	9.57%	12.31%	11.24%	10.53%	12.79%	13.13%	11.94%	14.52%	12.50%	14.18%	9.51%	**11.81%**
11	9.58%	12.80%	14.05%	9.63%	8.76%	8.93%	9.29%	8.71%	11.19%	12.70%	8.28%	9.46%	**10.28%**
12	8.27%	6.03%	7.24%	13.89%	12.15%	12.04%	13.62%	12.79%	14.38%	11.39%	11.66%	10.04%	**11.13%**
13	8.79%	11.34%	12.62%	9.88%	8.58%	9.92%	10.74%	9.93%	11.45%	11.82%	10.83%	8.92%	**10.40%**
14	8.50%	9.54%	9.09%	9.48%	9.34%	10.39%	11.39%	11.86%	13.37%	9.33%	10.63%	9.71%	**10.22%**
15	7.72%	8.32%	8.80%	3.34%	4.28%	7.92%	7.48%	4.91%	5.73%	8.97%	10.41%	10.79%	**7.39%**
16	8.75%	5.26%	5.29%	2.97%	2.81%	3.37%	3.45%	4.58%	4.36%	9.41%	11.06%	11.87%	**6.10%**
17	6.61%	3.08%	3.47%	2.60%	2.80%	3.18%	3.20%	3.74%	3.86%	8.57%	6.60%	8.07%	**4.65%**
18	7.13%	3.15%	2.95%	2.49%	2.31%	3.11%	3.42%	3.21%	3.36%	8.36%	9.49%	11.81%	**5.07%**
19	6.45%	2.76%	2.88%	2.71%	2.53%	2.79%	3.00%	3.29%	3.98%	9.46%	8.08%	10.53%	**4.87%**
20	6.60%	2.87%	3.09%	2.61%	2.37%	2.73%	2.86%	3.32%	3.82%	9.93%	9.57%	11.34%	**5.09%**
21	8.49%	3.01%	3.42%	2.93%	2.22%	2.65%	2.73%	3.44%	3.38%	11.51%	11.19%	15.87%	**5.90%**
22	10.09%	3.61%	2.81%	1.94%	2.21%	2.54%	2.74%	2.59%	4.02%	12.57%	15.00%	24.42%	**7.05%**
23	17.79%	2.59%	2.22%	4.01%	3.50%	3.35%	4.06%	2.98%	3.56%	10.18%	22.61%	24.14%	**8.42%**

Table 2.10: Impact Registration hour on zero values per month.

Based on the information mentioned above, we determine the zero value percentage based on the following:

- Month;
- Day of the week;
- Registration hour.

We need to predict the number of direct returns upfront. Because the registration hour is unknown upfront, we cannot use the same percentages as the zero-values. However, the month and day of the week are known for the entire planning window. Therefore, the number of direct returns is predicted using a percentage of the month and day of the week.

3. Literature review

Demand forecasting has been the subject of research in multiple fields, which contrasts with return forecasting. Numerous studies of demand forecasting have focused on time series forecasting, which is an essential area of forecasting in which historical observations of the dependent variable are obtained and analyzed to develop a model which describes the underlying process. Any time series can be thought of as being composed of five components, namely level, trend, seasonal variations, cyclical movements and irregular random fluctuations (Silver, Pyke, & Thomas, 2016). Mentzer & Cox Jr. (1984) found that Moving Average (MA), Exponential Smoothing (ES) and regression were well-known and widely used approaches for demand forecasting. There are often external features that affect time series. Machine learning techniques can integrate those features. In this chapter, we answer research question 2: *'Which methods are described in available literature regarding the (daily) forecast of the number of returns?'*. Preliminary literature research is given for quantitative forecasting in Section 3.1 and Machine Learning in Section 3.2. Furthermore, Cross-Validation is described in Section 3.3 as a method to overcome overfitting in most methods. To summarize this chapter, a taxonomy of the described methods is shown in Figure 3.1 for the found literature regarding return forecast as described in Section 3.4. A conclusion is provided in Section 3.5.

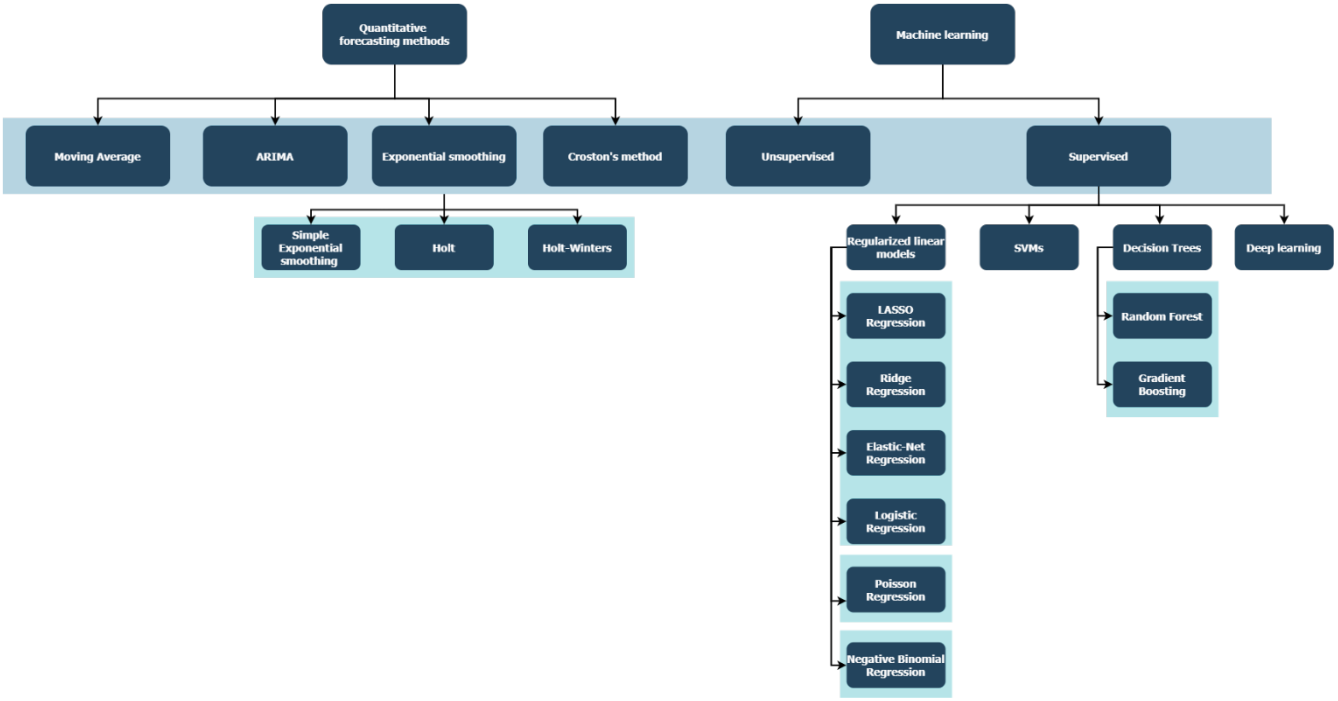


Figure 3.1: Taxonomy.

3.1 Quantitative demand approaches

Examples of traditional quantitative approaches are Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ES). Hamilton (1994) describes those methods in detail. Those approaches assume time series to be stationary, which means neither the mean nor the autocovariances depend on the date. Those approaches focus on the lagged values of the dependent variable.

3.1.1 MA and ARIMA

The MA and ARIMA rely on finding cyclical patterns to predict the volumes of the dependent variable. The ARIMA includes six parameters which determine the behavior of the model. The notation of the model is ARIMA(*p,d,q,P,D,Q).* The six parameters are divided over three techniques with and without seasonality. The three techniques are autoregressive, integrated and Moving Average. Those approaches usually do not allow for explanatory variables. The ARIMAX model is an exception, which is a widely used extension of the ARIMA model and has better prediction results. ARIMAX is just an ARIMA with additional explanatory variables. The model can be viewed as a multiple regression model with one or more autoregressive terms and one or more moving averages. A drawback of both models is that time series should be stationary. Since our data shows nonstationary time series, we do not describe this method in more detail.

3.1.2 Exponential Smoothing

Exponential smoothing (ES) is another representative quantitative approach. ES gives gradually declining weights to historic data. Simple Exponential Smoothing (SES) is a time series forecasting method for univariate data without a trend or seasonality, which only requires a single smoothing factor. SES is probably the most widely used statistical procedure for short-term forecasting (Silver, Pyke, & Thomas, 2016). Babai, Ali, Boylan, & Syntetos (2013) found that univariate time series of high sales volumes can be handled successfully using different ES methods. However, ES is less appropriate when demand is intermittent, because ES places more weight on the most recent data, which generates biased estimates when there is a mass around zero value observations. In our case, we have also zero values in the days between the registration in Boomerang and processing date, therefore this method is less suitable for our research.

However, according to the studies of Makridakis, Spiliotis, & Assimakopoulos (2018) and Crone, Hibon, & Nikolopoulos (2011), the two best performing methods are ARIMA and a variation of Exponential Smoothing, namely Error, Trend and Seasonal (ETS) in case of time series data. Basically there are three base models of ETS, which are divided based on the criterion of having trend and/or the seasonal component. Those models are Simple Exponential Smoothing (SES), Holt's linear method (Holt) and Holt-Winter's method (Holt-W). Table 3.1 visualizes the Exponential Smoothing methods. Two different errors are distinguished, namely additive and multiplicative errors. Additive errors are calculated by the difference between the forecasted and actual value. Multiplicative errors are calculated by the difference between the forecasted and actual value, divided by the forecasted value.

Trend/Seasonality	**No**	**Additive**	**Multiplicative**
No	SES	Holt-W	Holt-W
Additive	Holt	Holt-W	Holt-W
Multiplicative	Holt	Holt-W	Holt-W

Table 3.1: Comparison Exponential Smoothing methods.

SES is best suited for a short-term forecast, with no clear trend or seasonal pattern. SES is a type of weighted moving average and requires only an estimation of one parameter, namely the smoothing constant alpha. Contrary to SES, the Holt model has a trend component which need to be updated. The Holt-Winters model is an extension of exponential smoothing and Holt for using in trended and seasonal time series. The model requires to update two parameters, the trend and level components. Exponential smoothing techniques are simpler than many other forecasting techniques and can produce good results

with less computation time. Exponential Smoothing techniques provide better results than MA, but are still simple and inflexible in terms of using fewer data for the prediction of the future values. Based on Table 3.1, Holt-Winters includes both trend and seasonality and would be the most extensive method for using Exponential Smoothing. However, Exponential Smoothing is not the best forecasting method, because our data shows intermittent demand.

3.1.3 Croston's method

An approach to forecast intermittent demand is Croston's variant of Exponential Smoothing. This method was developed to provide the mean demand per period with a higher accuracy. Similar to Exponential Smoothing, Croston's method assumes a normal distribution (Willemain, Smart, & Schwarz, 2004). The method forecasts on the size of a demand and the time period between demands. The method is mainly beneficial in case of low demand. The main disadvantage of this method is the lack of accurate estimates of demand per time period (Synthethos & Boylan, 2001). Because we are interested in the accuracy of the daily forecast, we do not investigate this method in more detail.

3.2 Machine Learning

Besides quantitative methods, demand forecasting can also be based on Machine Learning. Machine Learning techniques can be divided into supervised and unsupervised Machine Learning applications. The primary objective of supervised learning is to learn the mapping between input and output variables, such that the system can compute predictions. Supervised learning is the case where you have input variables (x) and an output variable (y) and you use an algorithm to learn the mapping function from the input to the output. This technique can be useful for our research since we have many input data that can predict the output variables. Supervised learning problems can be divided into regression and classification problems. The problem is called a regression problem if the output variable is a real value, while classification problems contain a category.

Unsupervised learning holds when you only have input data and no corresponding output variables. The main goal is to model the underlying distribution or structure in the data to learn more about the data. Because we have a corresponding output variable, we do not focus on this technique in the remainder of this research.

Supervised methods can be divided into four types of methods, namely regularized linear models, SVMs, Decision Trees and Deep learning (Khosla, Jamison, Ngo, Kuceyeski, & Sabuncu, 2019). Since our goal is to have a return forecast with a higher accuracy, but also with a high interpretability, regularized linear models and Decision Trees are preferred over SVMs and Deep learning methods. In the remainder of this section, we describe in more detail the most common methods for regularized linear models, namely LASSO regression, Ridge regression, Elastic-Net regression, Logistic regression Poisson regression and Negative Binomial regression and for the Decision Trees the two most commonly used methods Random forest and Gradient Boosting.

Linear regression is a relatively inflexible approach, because it can only generate a linear function. However, since we are mainly interested in inference, we prefer this inflexible approach because it is more interpretable. The tradeoff between flexibility and interpretability for different learning methods is represented in Figure 3.2.

Figure 3.2: Trade-off between interpretability and flexibility, retrieved from (James, Witten, Hastie, & Tibshirani, 2013).

3.2.1 LASSO Regression

The relationship between dependent and independent variables is used to predict future values instead of using an historical time pattern. Several approaches are known in the literature. The most straightforward statistical method is the linear regression model. A linear regression model minimizes the residual sum of squares (RSS). If the relationship is between a dependent variable and multiple explanatory variables, the process is called multiple linear regression. There are several modifications of linear regression, of which LASSO regression is one.

The LASSO of Tibshirani (2011) is a generalization of a linear regression, in which not only the RSS is minimized, but also a linear penalty function of the coefficients is included. The choice of the right explanatory variables is central for high-dimensional datasets. The response variable can follow different types of probability distributions. If this variable is normally distributed, we consider the linear model. For an explanatory variable that is not normally distributed, a generalized linear model can be used. The goal of a linear regression model is to fit a straight line to several points, while minimizing the sum of squared residuals.

LASSO is an acronym for Least Absolute Shrinkage and Selection Operator in order to gain statistical insights in variable selection and provide recommendations for short-term forecasting. Model selection is a crucial part for further analysis of any multiple regression model and its forecasting performance. The variance of the model increases if too many regressors are picked but can be biased and inconsistent if fewer regressors are included in the model. Both negatively affect the accuracy of the forecast (Savin & Winker, 2013).

In our case, more than two explanatory variables are involved. Therefore, we use a Multiple Linear Regression instead of the Simple Linear Regression. A Linear Regression Model describes the relationship between response variable Y_i and explanatory variables X_{ij}. The model is assumed to follow a normal distribution with mean μ_i and variance σ^2. Linearity of the model is another assumption, which is the linear relationship between the response variable Y_i and explanatory variables X_{ij} with random error ϵ. The random error is assumed to have a mean of zero, has a constant variance and is independent.

$$Y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ik}\beta_k + \epsilon_i \qquad i = 1, \ldots, n \qquad (1)$$

Parameters $\beta_0, \beta_1, \ldots, \beta_k$ represent the regression coefficients and k shows the number of explanatory variables.

The minimization formula of the LASSO is shown in equation 2.

$$\beta^{lasso} = \sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta_j + \beta_0)^2 + \lambda \sum_{j=1}^{p}|\beta_j|, \quad (2)$$

Where $\lambda \geq 0$ is the tuning, or regularization parameter. If $\lambda = 0$, the LASSO estimator is equal to the standard OLS estimator. On the other hand, if λ is large, all coefficients $\beta_j s$ become zero. If λ approaches infinity, the solution approaches the global mean. Since you are shrinking all the model parameters, they go to zero. So, we are only left with minimizing the squared difference between the training data and a constant value, which is equal to the mean. Features are removed sequentially in order of least important to most important.

For intermediate values of λ, there is a balance between minimizing the first term, the Residual Sum of Squares (RSS) and the second part, the shrinkage of the coefficients towards zero. Therefore, it is important to obtain good values for λ for successful model selection. A good value of λ is different for each model but should minimize the error. This constraint relating to the sum of the regression coefficients, is a penalty for adding too many variables to the model. Therefore, overfitting is constrained (Zhang, Minchin, & Agdas, 2017).

3.2.2 Ridge Regression

A regression model that is similar to LASSO is the Ridge regression. Ridge regression has a disadvantage compared to LASSO, because it includes all predictors in the final model. The difference is within the penalty. The formula for the Ridge regression minimization function is shown in equation 3.

$$\beta^{Ridge} = \sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta_j + \beta_0)^2 + \lambda \sum_{j=1}^{p}\beta_j^2, \quad (3)$$

The penalty within equation 3 of the Ridge regression will shrink the coefficients to zero, but not exactly to zero. Only if the tuning parameter $\lambda = \infty$, the Ridge regression can shrink coefficients to zero. This may not be a large problem for the accuracy, but it can create less interpretability since all coefficients are used for generating the model. The LASSO overcomes this disadvantage by variable selection. LASSO yields sparse models, which means models that involve only a subset of the variables (James, Witten, Hastie, & Tibshirani, 2013). The tuning parameter λ controls the strength of the penalty. If λ is sufficiently large, coefficients are forced to be exactly equal to zero, in which dimensionality can be reduced. The higher the tuning parameter λ, the more coefficients are shrinked to zero. On the other hand, if $\lambda = 0$, the LASSO and the Ridge regression are equal to the Ordinary Least Square (OLS) regression. The difference between the LASSO and Ridge regression model can be explained using Figure 3.3.

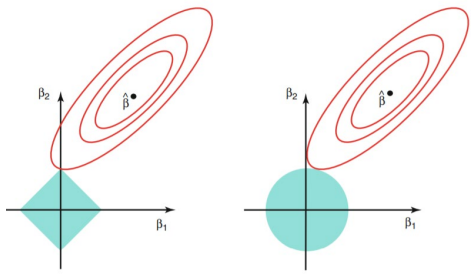


Figure 3.3: Contours of the error and constraint functions of the LASSO (left) and Ridge regression (right), retrieved from (James, Witten, Hastie, & Tibshirani, 2013).

The red ellipses present the contours of the RSS, while the blue areas are the constraint regions. We can rewrite equations 2 and 3 to solve the problems for the LASSO and Ridge regression respectively.

$$Minimize\ \beta^{lasso}\ \{\sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta_j + \beta_0)^2\}\ \ subject\ to\ \ \sum_{j=1}^{p}|\beta_j| \leq s, \tag{4}$$

and

$$Minimize\ \beta^{Ridge}\ \{\sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta_j + \beta_0)^2\}\ \ subject\ to\ \ \sum_{j=1}^{p}\beta_j^2 \leq s\,, \tag{5}$$

In other words, for every value of λ, there is a *s,* for which equation 2 and 4 give the same LASSO coefficient estimates. The same holds for the Ridge regression in equation 3 and 5. Based on the equations 4 and 5, the blue areas are created. The intersection of the RSS and the constraint function of the LASSO lies on the axis, which implicates that β_1 will be excluded from the model. This does not hold for the Ridge regression, since the blue area has no sharp points, the intersection will not generally occur on an axis. The situation for higher dimensions is more complicated but is based on the same principle. The diamond becomes a polyhedron if more variables are included in the model. However, this polyhedron also has sharp points, therefore variable selection is still possible for the LASSO model.

LASSO cannot identify all 'true' predictors in a dataset if the correlation is high between the regressors. The LASSO tends to choose only one of the highly correlated variables instead of the whole group and consequently misleading results are obtained. Therefore, the LASSO is only consistent if the correlation settings are low, however it can still provide good approximations for large sample sizes (Savin & Winker, 2013). Furthermore, LASSO is not able to select more than *n* variables, which does not cause any problems if k< *n*, but if this does not hold, we do not obtain the right model. LASSO is more robust, so less sensitive to outliers in data compared to Ridge regression. The Ridge regression has always a unique solution, which is not the case for LASSO, which may have multiple solutions.

James et al. (2013) show that in general, they expect that LASSO performs better in a setting with a relatively small number of predictors that have substantial coefficients. Ridge regression performs better when many predictors, with roughly equal sizes of coefficients, are included. However, they have a qualitatively similar behavior, if lambda increases, variances decreases and bias increases. However, the number of predictors related to the response cannot be known upfront for real data. Therefore, Cross-Validation can be used to determine which approach results in lower variation.

3.2.3 Elastic Net Regression

A method that combines the LASSO and the Ridge regression is Elastic Net regression as proposed in Zou and Hastie (2005). The function is shown in equation 6.

$$\beta^{Elastic\ net} = \sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta_j + \beta_0)^2 + \lambda\sum_{j=1}^{p}\left(\alpha|\beta_j| + (1-\alpha)|\beta_j{}^2|\right), \tag{6}$$

Where $\alpha \in [0,1]$ is the parameter that can be varied. When $\alpha = 0$, the equation is reduced to Ridge regression, and with $\alpha = 1$, the equation is reduced to the LASSO. The penalty parameter α determines how much weight should be given to either the LASSO or the Ridge Regression. It is a strictly convex problem, for $\alpha > 0$, due to the Ridge Regression part. So regardless of the correlation between the independent variables, a unique solution exists. The Elastic Net Regression overcomes the problems of

selecting more than *n* predictors if *p>n* and not only pick one predictor out of a group. Elastic Net can result in lower mean squared errors compared to LASSO (Bühlmann & van de Geer, 2011). Furthermore, Elastic Net produces a higher number of correctly identified influential variables compared to LASSO (Tutz & Ulbricht, 2009). However, the computational costs are higher. In the research from Cui et al. (2020), the optimal value for α was equal to 1, which means that the outcome was equal to the LASSO.

3.2.4 Logistic Regression

Logistic Regression is used in binary problems in which we predict class membership, which is often called classification. It is an extension to the linear regression, where the dependent variable is binary. For each observation, it determines the probability that the dependent variable will take the value of 1 (Hastie, Tibshirani, & Friedman, 2009). According to the research of Asdecker et al. (2018), the binary logistic regression is easy to conduct but also allows for detailed analysis regarding the factors that affect consumer return behavior. They show that the price is significant on a 0.05 level for the number of returns, together with the number of articles and delivery time. Logistic Regression is not applicable to forecast het timing of the return, but it can identify whether a return request will be returned.

The Logistic Regression model establishes a relationship between the binary dependent variable and a group of explanatory variables. It models a logit-transformed probability of the linear relationship with the explanatory variables, using the following equation:

$$logit(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \quad (7).$$

However, we are interested in the probability of the success, which is the inverse of the logit function. This probability is called the Sigmoid function and is calculated using the following equation:

$$p = \frac{1}{1+\exp\left(-(\beta_0+\beta_1 x_1+\cdots+\beta_k x_k)\right)} \quad (8).$$

Compared to linear regression, the outcome of Logistic regression has only a limited number of possible values. The outcome of linear regression is continuous. Furthermore, Logistic regression uses a maximum likelihood method to arrive at the solution, while linear regression uses the ordinary least squares (OLS) method to minimize the errors.

3.2.5 Poisson Regression

Because one of our response variables is a counted number, namely the number of days between the registration and arrival, the distribution is discrete and limited to non-negative numbers. Problems occur when applying linear regression to our dataset. First, the distributions of count data are mostly positively skewed with many observations containing a zero value. Secondly, the regression model will be likely to produce negative values, which are theoretically impossible. An often used solution is to use a Poisson Regression model. A Poisson Regression has a discrete distribution, a skew and a restriction of non-negative numbers. The regression model follows a Poisson distribution of the errors instead of a normal distribution. Furthermore, it models the natural log of the response variable as a linear function of the coefficients instead of modelling the linear function of the regression coefficients (Gardner, Mulvey, & Shaw, 1995). For Poisson Regression, the incidence rate μ is determined by a set of *k* regressor variables, which is calculated the following:

$$\mu = exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k) \ , \tag{9}$$

where β_0 is called the intercept. The regressor coefficients $\beta_1, \beta_2 \ldots \beta_k$ are the unknown parameters that are estimated using the dataset. Those coefficients are estimated using the method of maximum likelihood (MLE). The fundamental Poisson Regression model for an observation *i* is given by

$$\Pr(Y_i = y_i | \mu_i, t_i) = \frac{e^{-\mu_i t_i} (\mu_i t_i)^{y_i}}{y_i!} \ , \tag{10}$$

where μ_i represents the risk of a new occurrence of the event during a specified exposure period *t*. The Poisson distribution has the property that its mean and variance are equal.

3.2.6 Negative Binomial regression

Negative Binomial distribution has one parameter more than the Poisson Regression, which adjusts the variance independently from the mean. Therefore, Negative Binomial Regression is more flexible than Poisson. If the variance is higher than the mean, over-dispersion is likely to be present and indicates that Negative Binomial Regression would be appropriate. The confidence intervals for the Negative Binomial are often narrower as compared to those from a Poisson Regression model. The Negative Binomial distribution describes the probabilities of occurrence of whole positive numbers, which is the same for the Poisson distribution. Therefore, it might be useful for modeling counts. The variance of the Negative Binomial distribution is given by

$$\text{var}(Y) = \mu + \frac{\mu^2}{k}, \tag{11}$$

where *k* is called the dispersion parameter. If this dispersion parameter increases to infinity, the variance converges to the same value as the mean and turns the Negative Binomial into a Poisson distribution.

3.2.7 Random Forest

Random Forest is a variant of a bagging method. A bagging method implies that *N* learners (decision trees) are created and produce *N* new training data sets by random sampling with replacement from the original set in a parallel and independent way. The final prediction is equal to the average of those *N* decision trees. This bagging method overcomes the sensitivity of specific data in the training set. Random Forest includes an extra step, namely random selection of features instead of using all features to grow trees (Alonso, Torres, & Dorronsoro, 2015). Generally, two tuning parameters should be tuned, namely the number of variables that are chosen from the input variables and the number of trees to grow. If more trees are used, variance will be less. Random Forest can handle thousands of input variables without deletion and runs efficiently on large databases. Furthermore, interactions are automatically tested, which could provide better results.

3.2.8 Gradient Boosting

In contrast to the bagging method, the boosting method trains the individual models in a sequential way. Which implies that each individual model learns form the previous mistakes. The Gradient Boosting method learns from the residual error directly, rather than update the weights of the data points. Data points that have a high residual error are more likely to be included in the new training set. Therefore, it introduces leaf weighting to penalize those that do not improve the predictability of the model. The number of parameters that should be tuned are the shrinkage parameter, depth of the tree and the number of trees. In contrast to Random Forest, tuning parameters are harder to fit, because increasing the

number of trees does not necessarily lead to a better fit, since overfitting can be a result. However, if the tuning parameters are correctly used, Gradient Boosting generally provides somewhat better results.

3.3 Test on overfitting through K-fold Cross-Validation

A forecast model can under- or overfit the data, which can result in a high forecast error. If a forecast model extracts repetitive behavior in a simple way and estimates future demand in a general way, the model will be underfitted. Contrary, if the forecast model captures every repetitive behavior, the model would be overfitted. Underfitting is detected and solved by the model by minimizing the error, but overfitting is much harder to detect and solve. A way to identify overfitting is to do Cross-Validation. K-fold Cross-Validation divides the data into a training and testing set. K-fold Cross-Validation randomly divides the set of observations in *k* folds (groups) of approximately equal sizes. The first fold is treated as the validation set and the method is fit on the remaining *k*-1 folds. The resulting validation is typically assessed using Mean Squared Error (MSE). The procedure is repeated *k* times, and each time a different group of observations is treated in the validation set and the MSE is computed. The k-fold Cross-Validation is eventually computed by the average of those MSE's (Efron & Tibshirani, 1994).

$$K - fold\ CV = \frac{1}{k}\sum_{i=1}^{k} MSE_i\ , \tag{12}$$

Typically, a *k*-fold of *k*=5 or *k*=10 is used, due to computational effort (Kohavi, 1995), (Breiman & Spector, 1992). A grid of values for the tuning parameter lambda are chosen, and the Cross-Validation error is computed for each lambda. The tuning parameter value which has the lowest Cross-Validation error is chosen as the lambda.

3.4 Research done

In this section, we describe previous literature found regarding return forecast. In total, twelve different researches are investigated and described below.

In the past years, research have been done regarding the forecast of returns by several authors. Goh & Varaprasad (1986) made the first initial effort to study the product returns in a statistical way. They proposed a Box-Jenkins transfer function model, relating returns to previous sales. The model estimates the return probability using a proportion of the total product returns.

Kelle & Silver (1989) used return proportions for forecasting demand to determine the quantities of reusable containers that will be returned. They used the return proportions introduced by Goh and Varaprasad. They developed four different models to calculate the return demand forecast, depending on various factors. They started with a model which included the probability of return. The second model analyzed each time bucket separately, resulting in probabilities of returns per time bucket. The third model was an addition of conditional probabilities to the second model. The last model was the second model plus aggregated return data. They state that the model would be more accurate if individual information was available.

Guide & Srivastava (1997) introduced the idea of using intrinsic forecasting for estimation of return quantities and return rates by using time series. Their efforts were mainly focused on the capacity planning in remanufacturing, by calculating the product recovery rate.

However, Hess and Mayhew (1997) were one of the first few researchers who developed a statistical forecasting model to estimate the number of commercial returns. They used a direct marketing model instead of a traditional marketing model, which involves higher probability of returns compared to the traditional market. Two key components of the return phenomenon should be modeled to understand returns, namely the timing of return and the probability of the return. To gain insight in when the return will occur, a simple historic average time-to-return or a linear regression model can be developed, which includes factors that may affect the time-to-return. They showed that the higher the price, the earlier the return. A dependent variable to be regressed on could be the time between the sale and return. This model attempts to explain the variation of the return times. The probability of return is determined using a logit model. However, data is censored because both approaches have the problem that not all items that eventually will be returned are already returned. Furthermore, the regression model has an arbitrary assumption of normally distributed random errors, which has a negative tail.
Therefore, they developed the hazard rate model, since this is a pure function of time. The split hazard model explains not only the returns, but also nonreturns. Probability that item would be returned, multiplied by the probability that it would have been returned by that point in time. The probability of observing a nonreturn is the sum of two probabilities, defined by the probability that the item will never be returned plus the probability that the item will be returned multiplied by the probability that it would not have been returned by that time.

Toktay, van der Laan, & de Brito (2003) assume that the return process can be modeled by the probability that a product will ever be returned multiplied by the probability that the product will be returned after a number of periods, conditional on ever being returned. This type of relation is indicated as a 'distributed lag model'. They showed that using a geometric lag model for single use cameras makes practical sense, since most purchases are impulsive and returned quickly after sales. They state that take back price and trade in offers influence the returns.

De Brito (2004) evaluated the impact of (mis)information of the four methods to forecast the number of returns proposed by Kelle et al. (1989). The first method which has only knowledge about average behavior, performs in general very poorly and is not recommended for practical implementation. The second method that includes information on the return distribution provides a sufficient level of sophistication. It is in general better to underestimate the return rate rather than overestimate, because stockouts are usually more expensive than overstocks. The third method uses a periodic record of returns and the fourth method needs to track back the period each individual product was sold. Although the fourth method gives the best results, the second model is exceptionally robust given misinformation compared to the other models.
Potdar (2009) uses a combination of two appraoches, namely central tendency, which uses a moving average and an extreme point approach, which is based on data envelopment analysis (DEA) together with linear regression. They forecast product returns using reason codes based forecasting for the Consumer Electronics industry.

Potdar & Rogers (2012) incorporate reason codes in their forecasting model as stated by Potdar (2009). They try to understand the data pattern for each reason code, to apply an appropriate method for each code to predict the future. The reason codes are divided over three categories, namely:

- Product is defective or delivered with damage;
- Return is without any reason;
- Product does not have the desired features or is not worth the price.

They use a DEA-CCR model to analyze the performance indices and compare multiple items based on their in- and output. A correlation method is used to find the correlation between the variables rank and percent returns. They believe that those reason codes can effectively translate consumer behavior into meaningful data, which can be integrated into the return forecast.

Clottey, Benton, Jr, & Srivastava (2012) use a distributed lag model (DLM) to capture the dependence of returns on sales in previous periods, as considered by Toktay, Wein, & Zenios (2000). An advantage of this model is that it requires less data. Clottey et al. propose a model that provides an alternative to the geometric delay model.

Liang, Jin, & Ni (2014) utilize an effective characterization of three influence factors sales, life expectancy and return behavior. Other factors were not proven to have significant influence on return quantity. However, the reasons of product returns are limited to failure-induced return and end-of life return. Other reasons are not considered. They state that the return function is usually heavily skewed to the left, which indicates that customers return the products in a short period of time. This can be modeled by inverse Gaussian functions, due to its skewness, positive support, relatively ease expression and flexibility in modeling. The inverse Gaussian function, retrieved from Liang et al. (2014) has the general probability distribution function:

$$C\ t \qquad \left[\frac{i}{2\pi t^3}\right]^{1/2} e^{\frac{-i\,t-j\,^2}{2j^2 t}};\ i,j>0 \tag{13}$$

With C(t) as the probability that the customer returns the product *t* times after it has failed. The results were verified using Monte Carlo simulation.

Asdecker & Karl (2018) considers five approaches to forecast returns. Binary Logistic Regression is the simplest method which is taken into account. For each observation, the binary Logistic determines the probability of receiving a value of '1' for the dependent variable. Secondly, they consider the linear discriminant function analysis. The main idea is to create a linear combination of independent variables, which classifies the available data in the best way. Further, they consider the artificial neuronal network. Connected neurons can exchange signals with each other to find a function that best assigns input data to the correct output. Next to this, they also consider the decision tree learning C5.0 algorithm. They state that the C5.0 algorithm is the faster and more efficient successor of the widely-employed C4.5 algorithm. Lastly, they consider the ensemble learning technique, which uses several algorithms to improve the predictive performance. The three techniques retrieved from the training set were the decision trees C5.0, CHAID and QUEST. According to the results, ensemble learning technique provides the highest accuracy of 68.45%. The binary Logistic Regression performs surprisingly well, with an accuracy of 66.79%. Therefore they state that the simple models might be a better choice in business practice.

Cui, Rajagopalan, & Ward (2020) propose one of the high-dimensional methods, namely Least Absolute Shrinkage and Selection Operator (LASSO) or also known as L1 regularization to use as a predictive model, achieving best accuracy prediction for future return volume. They analyze a car manufacturing company, with a large product variety. This company also handles the returns by themselves. Though aggregate return volume in each period is of first-order interest, they are also interested in predicting return volume by each product type and each retailer. They use a dataset containing 331,390 products that are sold. Their focus is partly to identify the variables that can help predicting the volume of returns. They found that LASSO was effective in selecting the variables that

are useful in predicting the future return volume out of multiple machine learning methods. Their predictor variables are:

- Sales effect;
- Time effect;
- Product effect;
- Retailer effect;
- Production process and resources;
- Multi-product effect;
- Historical returns.

They found similar results for the Elastic-Net method, since the method was reduced to LASSO, because the alpha was equal to 1. The decision trees Random Forest and Gradient Boosting outperform LASSO in the training set but are worse in the test set.

3.5 Conclusion

Based on previous research described above, a conclusion of our studied methods is given in Table 3.2. For each research, it is stated which method shown in Figure 3.1 is used. According to the table, not all researches include one of the methods. However, the other proposed methods are to our knowledge not proven to increase the accuracy of our forecast. Therefore, we do not investigate them in more detail and focus on the researches that did include our described methods.

Goh & Varaprasad (1986)	ARIMA, Box-Jenkins model
Kelle & Silver (1989), De Brito (2004)	Other: Naïve estimation
Guide & Srivastava (1997)	Other: Rough cut capacity planning
Hess and Mayhew (1997)	Regression model, Split hazard model
Toktay, van der Laan, & de Brito (2003)	Other: Distributed lag model
Potdar (2009), Potdar & Rogers (2012)	Regression model, MA, DEA
Clottey, Benton, Jr, & Srivastava (2012)	Other: Distributed lag model
Liang, Jin, & Ni (2014)	MA, Method based on influential factors
Asdecker & Karl (2018)	Regression model, plus 4 other models
Cui, Rajagopalan, & Ward (2020)	Regression model, Random Forest, Gradient Boosting

Table 3.2: Conclusion of the described models from the literature review.

The regression model is most often used as shown in Table 3.2 and has proven to provide good results. As mentioned in section 2.3, we need to classify whether a return request will be returned and secondly when the return request will be returned to the warehouse. Different types of regression models are proposed. Logistic Regression performs well if it contains a binary variable, which is the case for forecasting whether a return request will be returned. According to the research of Asdecker et al. (2018), the binary Logistic Regression is easy to conduct but also allows for detailed analysis regarding the factors that affect consumer return behavior. Therefore, we will use the Logistic Regression to determine whether a return request will be returned. In addition, decision trees can be used for classification and could provide better results due to the automatic inclusion of interaction effects. Random Forest is preferred over Gradient Boosting, due to the lower computational time and lower risk of overfitting.

Linear regression is used by Hess and Mayhew (1997), Potdar (2009) and Potdar & Rogers (2012). However, more advanced methods of linear regression are used by Cui, Rajagopalan, & Ward (2020). The best performing methods were LASSO and Elastic Net, which outperform the other regression and classification models. The classification models, by which we refer to the decision trees, perform better in the training set, but worse in the test set. The decision trees can be more difficult to understand compared to the regularized linear models as shown in Figure 3.2, but have a higher flexibility. As mentioned before, we prefer an inflexible approach, because it is more interpretable. Therefore, we will use the LASSO method to forecast the timing of the return.

However, since our explanatory variable regarding the timing is a count variable, we also investigate the Poisson Regression model, which is to our knowledge not used for return forecast in previous research. Furthermore, Negative Binomial Regression is used since our data shows over-dispersion as shown in Section 2.3.1. We will compare the outcomes of the LASSO, Poisson Regression and Negative Binomial Regression models.

To conclude, we will use the following two complementary models to determine the number of returns.

1. To determine whether the return request will be returned:
    - Binary Logistic Regression and Random Forest.
2. To predict the timing of the return request:
    - LASSO, Poisson and Negative Binomial Regression.

4. Proposed model

Our focus in this research is to identify the variables that can help predict whether a return request is returned and how long this will take. In this chapter, we answer the third research question: '*How can we develop a short-term return forecast that produces more accurate results?'.*

First, we describe the forecasting method. Followed by a description of how to collect, process, analyze and synthesize the information for the inputs of the model.

4.1 Forecasting method

We need to perform two complementary forecasting models to determine the number of returns per day. The two models both have a specific purpose. Since we are interested in the quantity and time between the registered return in Boomerang and the arrival at the warehouse, we are interested in two parts:

1. Classification of whether an individual registered return request will be true (actually returned).
    - Logistic Regression and Random Forest, which are explained in Section 3.2.4 and 3.2.7 respectively.
2. The timing of the return request, which implies the time between registration and arrival at the warehouse.
    - LASSO, Poisson and Negative Binomial Regression, which are explained in Sections 3.2.2, 3.2.5 and 3.2.6 respectively.

The method of predicting the number of return requests that will actually be returned is shown in Figure 4.1. First, we need to determine the number of return requests that are returned based on the Boomerang registrations. Because the registered requests in Boomerang are not always returned, we use a binary Logistic Regression and a Random Forest tree to classify whether a registration will be returned. Based on this classification, we predict for those requests that are classified as true, the timing of the return. In Chapter 3, we described different statistical and time series models to predict the timing of the return. From the taxonomy we concluded that the regression model was used most commonly. The best performing method among those described Machine Learning methods is LASSO (Cui, Rajagopalan, & Ward, 2020), but also Elastic-Net Regression showed the same results. However, we only investigate LASSO because less variables are chosen by the model, which increases the interpretability. On the other hand, Poisson Regression could provide better results because our dependent variable is a count variable. In case of count data, distributions are often positively skewed with many observations containing zero values and non-negative values. A Poisson Regression has this skew and restriction of non-negative numbers and has a discrete distribution. However, the mean and variance are assumed to be equal in the Poisson Regression, which is not the case for the Negative Binomial Regression. Therefore, we investigate LASSO, Poisson and Negative Binomial Regression to determine which method should be implemented by Bol.com in order to predict the timing of the returns. However, the number of registered returns should be adjusted with the zero values and direct returns.

Based on the zero values and direct returns as explained in Sections 1.5 and 2.3.1, an adjustment to the return requests should be made. This adjustment is needed to decrease the number of registered returns for the zero values, but also for an increase in returns per day to cope with the direct returns without registration. The determination of the number of returns per day is explained in Section 4.5.

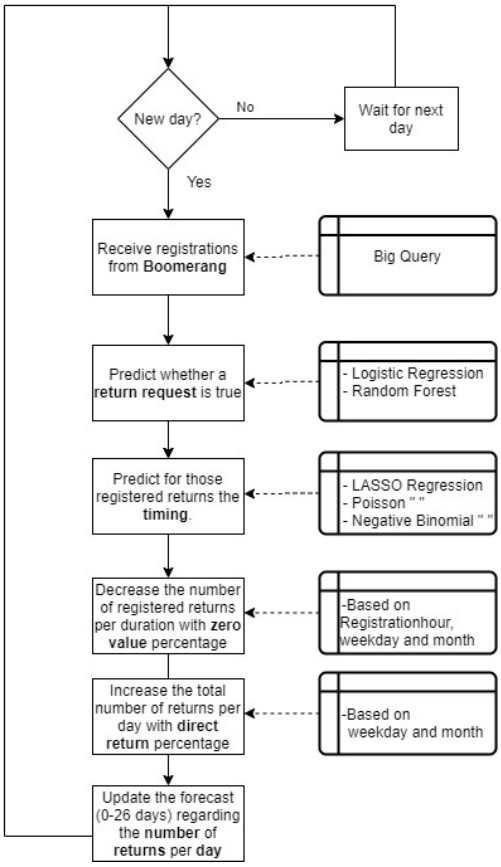


Figure 4.1: Process of forecasting the number of returns per day.

Since we have a large dataset for 2019 (millions of rows), we divide the dataset over the months. We will come up with a forecast for each value for each month based on the data of 2019. For each month, different coefficients will be predicted. Because we look at the short-term, we do not think seasonality will play an important role. For both the *quantity* response variable as well as the *timing* response variable, a detailed explanation of the models is provided in the remainder of this chapter.

We evaluate our models based on K-fold Cross-Validation as described in Section 3.3 to prevent overfitting. The procedure divides the set into *K* subsets of roughly equal sizes. It considers training on all but the *k*th part, and then validating on the *k*th part. Typically, a *K*-fold of 5 or 10 is used, due to computational effort (Kohavi, 1995), (Breiman & Spector, 1992). We use a 5-fold Cross-Validation to reduce the computational effort for our large dataset. The methods will be tested and evaluated individually.

4.2 Input of the model

For each model, except the LASSO model, we need to select upfront which factors have a significant impact on the demand side of the return forecast. For the LASSO method, this is of less importance due to the integrated feature selection. The determination of important features for the models are partly based on the previous researches described in Section 3.4, but also based on knowledge from Bol.com.

Based on previous research of Cui et al. (2020), the following predictors were selected by LASSO Regression for the prediction of returns:

- *Sales:* We do not use sales as an input, because our forecast window is on the short-term. Sales is more likely to be explanatory for the long-term instead.
- *Historical return data:* We will use historical return data of 2019 to fit the regression models and to determine the coefficients.
- *Time:* We take the time component as an input for our model. We create a forecast for each month but implement additional time components like day of the week and the registration hour.
- *Retailer:* From the research of Cui et al. (2020), retailers have proven significant impact on the return process. Therefore, we will select the retailer as a variable. Besides the retailer, we look also at the *source* of the registration, namely how the return is registered in Boomerang. Because in contrast to the research of Cui et al. (2020), we have a source of registration in addition to the source of retailer.

Research from Potdar et al. (2012) and Brito (2004) showed that *reason codes* influence the returns. They state that the reason why a customer wants to return the item has impact on the return process. Since this information is known in our case, we also include those reason codes as input variables. Furthermore, Hess et al. (1997) and Toktay (2003) showed that the higher the *price*, the higher and earlier the chance of return. Therefore, we will also take the price as an input variable.

In Chapter 2, we concluded that the year, month, week number and weekday influence the returns. Here we will only use data from 2019 to train the models. Therefore, year is fixed. However, we would advise to update the parameters every year to increase the accuracy of the forecast. This can be done by fitting the models with new data and adjust the coefficients of the models. The month is included in the forecast, since there will be a forecast for each month. We do not include week numbers since those are associated to the month and are not mentioned in any of the previous models described in Chapter 3. The *days of the week* on the other hand were also not mentioned in the literature, but have shown to influence the number of returns in Chapter 2. Therefore, we include the weekdays as a variable into our model.

Besides the variables mentioned in previous research, we think that more variables can influence the timing and chance of the return. Below, we give an overview of all input variables we think are relevant to forecast the *quantity* and *timing* of returns in Table 4.2. The response variable timing is excluded from the Logistic Regression and Random Forest tree, since this variable is not known upfront. Furthermore, the product effect variable Product group is included in the models, but the shop group is not included to avoid multicollinearity in the models. The variable processed return is excluded from the response variable timing, because this is a constraint for the input. The return should be processed, otherwise we cannot test the timing of the return.

Category	Input Variable	Explanation
Response variable quantity	*processedReturn*	This response variable indicates whether the return request was returned to the warehouse. We need this response variable to forecast the return requests that will be returned
Response variable timing	*daysBetweenRegistrationAndProcessing* *	This response variable indicates the timing of the return request.

Time effect	*dayOfWeek*	Since there are no returns delivered in the weekend, we think the day of the week influences the timing of a return.
	registrationHour	We assume that the hour of registration influences the timing of a return. Because an item can only be accepted at a PUP during opening hours.
Source effect	*sellingParty*	Since Cui et al. (2020) showed that retailers have a significant influence, we assume that it would also have a significant influence in our research. Therefore, we make a distinction between own and Plaza LvB-products
	source	In addition to the selling party, we consider the source of the registration. The registration can go via the Web shop, via customer service or the warehouse. As mentioned in Section 2.3, sometimes the customer does not register the return. In this case, the warehouse registers the return.
Multi-product effect	*quantity*	We think that if a customer returns multiple items, the time between registration and processing will be smaller.
Reason-code effect	*code*	According to the research of Potdar et al. (2012) and Brito (2004), reason codes have a significant influence on returns. Therefore, we take the codes into account.
Product effect	*Price*	Hess et al. (1997) and Toktay (2003) showed that the higher the price, the earlier the return.
	*Shop *, ***	We think some products will be returned quicker than others. Therefore, we consider the shops as another input variable.
	productGroup	As an extension to the shop, the product group is more specific.
Excluded from quantity response variable, ** Excluded from timing response variable		

Table 4.2: Overview of response and explanatory variables.

The data will be retrieved for each month of 2019 for the input variables stated in Table 4.2 from BigQuery. BigQuery uses the syntax code SQL and the written code can be found in Appendix C.

4.2.1 Data cleaning

Before we can use the models, the data should be analyzed and cleaned by converting categorical variables into numerical variables, deal with missing values and aggregate some categories.

Categorical

Because some input variables are not numerical but categorical, we need to convert these categorical variables into numerical variables. Each object visualized in Table 4.3 represents a categorical variable. We could translate each category to a number, but this allocation could influence the weight of the variable. Therefore, we create dummy variables with only two values: zero and one. In this way, each category is transformed to a dummy variable. For example, the *source* consists of Webshop, Docdata or Blue. Instead of using the categorical variable source, we create three dummy variables, namely *source_BLUE, source_DOCDATA* and *source_WEBSHOP* and use those variables instead of only source. In this way, the categorical variable is transformed into three binary variables, containing zero and one values. We converted each category into a dummy variable, which means our dataset consists of 84 variables instead of 9 as shown in Table 4.4.

Registration Hour	Selling Party	quantity	source	dayOf Week	code	Price	ProductGroup	Processed Return
int64	object	int64	object	int64	object	int64	object	int64

Table 4.3: Datatypes per variable.

INPUT VARIABLE	VARIABLES
PROCESSEDRETURN	ProcessedReturn – *Dependent variable*
DURATION	duration – *Dependent variable*
DAYOFWEEK	DayOfWeek
REGISTRATIONHOUR	registrationHour
SELLINGPARTY	sellingParty_Lvb sellingParty_Own
SOURCE	source_BLUE source_DOCDATA source_WEBSHOP
QUANTITY	quantity
CODE	code_ARTICLE_BROKEN code_ARTICLE_DAMAGED code_ARTICLE_DELIVERY_TOO_LATE code_ARTICLE_INCOMPLETE code_INCORRECT_PRODUCT_INFORMATION code_NO_REASON_PROVIDED code_OTHER code_WRONG_ARTICLE_ORDERED code_WRONG_ARTICLE_RECEIVED code_WRONG_SIZE_ORDERED
PRICE	Price
PRODUCTGROUP	productGroup_Algemene_Internationale_Boeken_PG productGroup_Algemene_Nederlandstalige_Boeken_PG productGroup_Auto productGroup_Baby_Hardwaren_PG productGroup_Baby_Verzorging_PG productGroup_Baby_en_Kindermode productGroup_Beauty_PG productGroup_Beeld_en_Geluid_Accessoires productGroup_Been_en_Ondermode productGroup_Brick_voor_tijdelijke_productclassificaties productGroup_Cadeaukaarten_PG productGroup_Camera productGroup_Dames_en_Herenmode productGroup_Desktop_Monitor_en_Beamer productGroup_Dierbenodigdheden_en_Ruitersport_PG productGroup_Diervoeding_PG productGroup_Drank_PG productGroup_Ebooks productGroup_Educatief_Internationaal productGroup_Educatief_Nederlandstalig productGroup_Ereaders_en_Accessoires productGroup_Erotiek_PG productGroup_Fiets productGroup_Film_PG

productGroup_Games_Accessories
productGroup_Games_Consoles
productGroup_Games_Software_Physical
productGroup_Gereedschap_en_Verf_PG
productGroup_Gezondheid_PG
productGroup_Groot_Huishoudelijk_PG
productGroup_Heat_en_Air
productGroup_Hobby_Spellen_en_Buitenspeelgoed_PG
productGroup_Home_Entertainment
productGroup_Household_Appliances
productGroup_Huishouden_PG
productGroup_Kamperen_en_Outdoor_Hardwaren
productGroup_Kern_Speelgoed_PG
productGroup_Kitchen_Machines
productGroup_Koken_en_Tafelen_PG
productGroup_Laptop_Computers
productGroup_Meubelen_PG
productGroup_Motor
productGroup_Muziek_PG
productGroup_Opslag_en_Netwerk
productGroup_PC_Accessoires
productGroup_Personal_Audio
productGroup_Personal_Care
productGroup_Persoonlijke_Verzorging_en_Huishoudmiddelen_PG
productGroup_Printen_en_Inkt
productGroup_Sanitair_en_Veilig_Wonen_PG
productGroup_Schoenen_PG
productGroup_School_en_Kantoor_PG
productGroup_Sieraden_en_Horloges_PG
productGroup_Sport_Hardwaren_PG
productGroup_Sport_en_Outdoor_Kleding_en_Schoenen_PG
productGroup_Tassen_Reisbagage_en_Modeaccessoires_PG
productGroup_Telefonie_en_Tablets
productGroup_Telefoon_en_Tablet_Accessoires_PG
productGroup_Televisie
productGroup_Textiel
productGroup_Tuin_en_Kerst_PG
productGroup_UNDEFINED
productGroup_Verlichting_PG
productGroup_Wearables
productGroup_Woondecoratie

Table 4.4: Input variables.

Missing values

Furthermore, the input variable *code* has missing values. In some cases, the customer does not motivate the return reason. Hence, there are empty cells for the *code* reason. Only less than 5 percent of this input variable has missing values. A common method to deal with missing values for categorical variables is

imputation using most frequent values (Galli, 2020), which is also known as MCI (Huang, Cao, & Srivastava, 2011). Therefore, we replace those missing values by the most often chosen reason code. In this way, no missing values occur in the dataset and enables the model to run without debugs.

Aggregate data of the sources
For consistency we fit the models based on the same categories. The categories are equal for each month, except for the input variable source. Sometimes, a distinction is made within the customer service department between the source. However, we do not think this distinction would add extra knowledge to the regression model. That is why we aggregate the sources of customer services into one category.

Additional data cleaning for each response variable individually will be explained in the next sections.

4.2.2 Feature selection

Often, some irrelevant and sometimes insignificant and unimportant features remain after data cleaning. The contributions of these types of features is often small and prevent the process from efficient prediction. Therefore, feature selection is a good solution to enhance the performance of a model by selecting the most important and relevant features of the dataset. The most commonly used general feature selection methods are Filter, Wrapper and Embedded methods.

- *Filter methods* are generally used as a data preprocessing step, because the selection of features is independent of a Machine Learning algorithm. Based on statistical scores, the correlation with the outcome variable is determined. This method filters irrelevant features out before classification starts.
- *Wrapper methods* need a Machine Learning algorithm and uses the corresponding performance as evaluation. Most commonly used methods are:
  a. *Forward feature selection*, in which the procedure starts with an empty set of features. In each iteration, the best remaining feature is added to the set.
  b. *Backward feature elimination*, in which the procedure starts with a full set of features. In each iteration, the feature with the smallest Pearson correlation with the predicted parameter is deleted.
  c. *Recursive feature elimination*, is a type of Backward feature elimination, but works on a feature ranking system. This method is not limited to linear regression and can incorporate resampling.
- *Embedded methods* select features that contribute the most to the training set. Regularization methods are often used, which penalize features given a coefficient threshold. LASSO is a well-known example of a Regularization method.

The recursive feature elimination (RFE) method provides good results and has also the option to include Cross-Validation. The Recursive Feature Elimination, Cross-Validated (RFECV) feature selection selects the best subset of features using RFE. Based on the Cross-Validation score of the model, the best subset is chosen. We will use this method for feature selection for the models with response variable quantity returns.

4.3 Forecasting whether a request becomes a return

First, we look at the response variable quantity of the returns. The percentage of registrations that are returned is shown in Table 4.5. On average, 81% is actually returned after registration.

Month	January	February	March	April	May	June	July	August	September	October	November	December	Average
%Returned	76.6%	79.9%	80.4%	81.1%	80.7%	83.8%	83.4%	83.2%	79.9%	81.7%	81.8%	79.9%	81.0%

Table 4.5: Percentage of returned requests.

There are different metrics to evaluate and/or compare classifier performance. We describe the most widely used performance measurements for classification.

- The *accuracy* is given by the formula: $\frac{(True\ Positive+True\ Negative)}{total\ number}$. In which *True Positive* is equal to the return requests that are forecasted to be returned and are actually returned. *True Negative* contains the number of return requests that are forecasted not to be returned and are not returned. In other words, the accuracy is calculated by the number of return requests forecasted correctly, divided by the total number of return requests. However, this metric is less suitable, since it represents only the total percentage correctly classified instances but no information about the number of the following classes: True Positives, False Positives, True Negatives and False Negatives. Therefore, low performance of a minority class can be easily unnoticed.
- The *precision* is given by the formula $\frac{True\ Positive}{True\ Positive+False\ Positive}$. Precision gives the percentage of correctly predicted returned requests divided by the total number of predicted returned requests.
- The *recall* is given by the formula $\frac{True\ Positive}{True\ Positive+False\ Negative}$. Recall gives the percentage of correctly predicted returned requests divided by the total number of actual returned requests.
- The *F1-score* is given by the formula $\frac{Precision*Recall}{Precision+Recall}$. The F1 score entails the harmonic mean of precision and recall. The F1 score is most often used, because there is no trade-off needed between precision and recall with this method.
- A *Confusion matrix* summarizes the performance of the classifier in a two-dimensional matrix. The observed class labels are presented in the top and the predicted class labels on the bottom.
- The *AUC* curve represents degree of measure of separability. Which means the model is tested on how capable the model is in distinguishing between being returned or not. The higher the AUC, the better prediction of True Positives and True Negatives. AUC stands for Area Under the ROC Curve. If the score is 1, the model can perfectly distinguish between classes and with a score of 0.5, the model cannot differentiate the classes. The curve is plotted with the *True Positive Rate* (recall) against the *False Positive Rate* $\frac{False\ Positive}{True\ Negative+False\ Positive}$.

We investigated the required number of features for both Logistic Regression and Random Forest in Figure 4.2 using the RFECV. The RFECV is based on the F1-score, to include both the recall and precision. The shaded area represents the variability of Cross-Validation, which is one standard deviation below and above the mean accuracy score drawn by the curve. From the results we can conclude that having 10 features for both methods result in similar results as having all features included. Therefore, we will also test the methods with only having 10 features. Those 10 features are

determined using the importance scores. The code for the response variable quantity is provided in Appendix E.

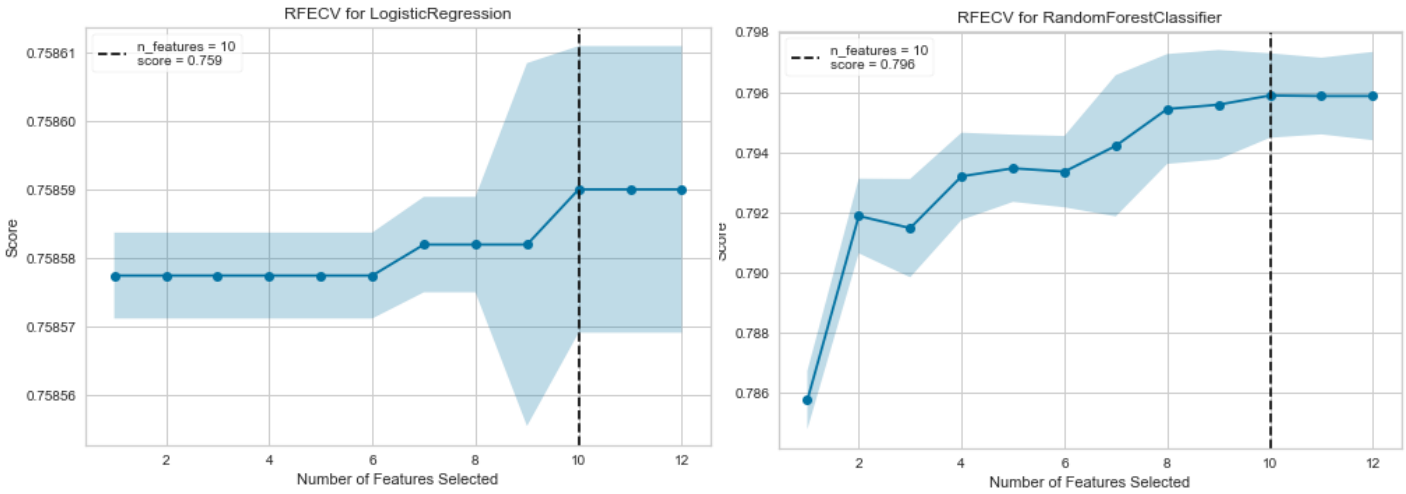


Figure 4.2: RFECV outcome of the number of features for the response variable quantity.

4.3.1 Predicting the quantity of returns using Logistic Regression

The Logistic Regression model is used to predict our dependent variable, namely whether the return request will be returned. This variable is a binary variable, that represents 1 (returned) or 0 (not returned). A visual explanation of the method is given in Figure 4.3. In total, 84 variables are used as an input for the regression model.

Due to the feature selection as described in section 4.2.2 and in this section, only the 10 most important features will be selected instead of all 84 features. Those 10 features are, stated from most important to least:

- *registrationHour,* which implies the rounded hour of registration.
- *dayOfWeek*, which states the day of the week.
- *Price*, which states the rounded price.
- *source_BLUE*, which implies the registration is done via Customer Service.
- *source_WEBSHOP*, which implies the registration is done via the website.
- *source_DOCDATA*, which implies the registration is done via the warehouse.
- *quantity*, which shows the number of return requests for the customer.
- *sellingParty_Lvb,* which shows that the product is not from Bol.com, but from a partner.
- *sellingParty_Own*, which shows the product is from Bol.com.
- *productGroup_Kamperen_en_Outdoor_Hardwaren*, which states the product group is 'Kamperen en Outdoor Hardwaren'.

Those 10 features represent more than 80% of the total importance. The importance scores can be found in Appendix D. The outcome of the regression model is a list with coefficients for each explanatory variable and a value of the intercept. The probability for an item to be returned is calculated according to the coefficients that are transformed to probabilities using the Sigmoid function: $f(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2 \dots \beta_{84} X_{84})}}$. Variable X represents the input variables of the function and β_0 is the

intercept and β_i the slope. The outcome is equal to the probability a return request will be returned. If the probability is equal to or higher than 0.5, we classify the outcome as 1 and 0 otherwise.

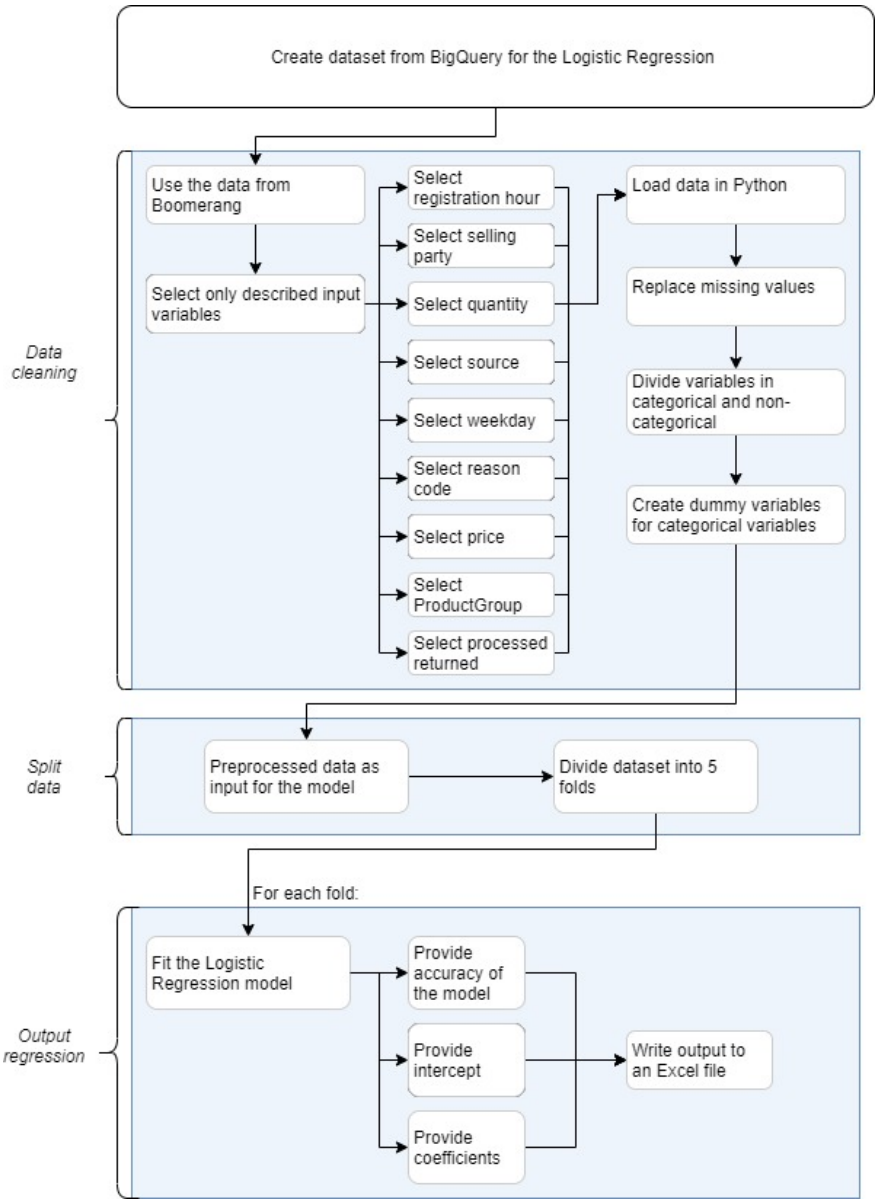


Figure 4.3: Visualization of the Logistic Regression model.

4.3.2 Predicting the quantity of returns using Random Forest

For the Random Forest, the data cleaning is the same as for the Logistic Regression. Instead of creating just one tree, multiple trees are built. The number of trees that are used are referred to as the number of estimators. Multiple trees are built and the average is taken to improve the predictive accuracy and controls overfitting. The number of estimators needed is determined using Figure 4.4. From the figure we can conclude that including more than 10 estimators has no added value. The AUC score does not increase and remains stable after 10 estimators. Similar to Logistic Regression, we will also evaluate the Random Forest model using only the 10 most important features as shown in Figure 4.2.

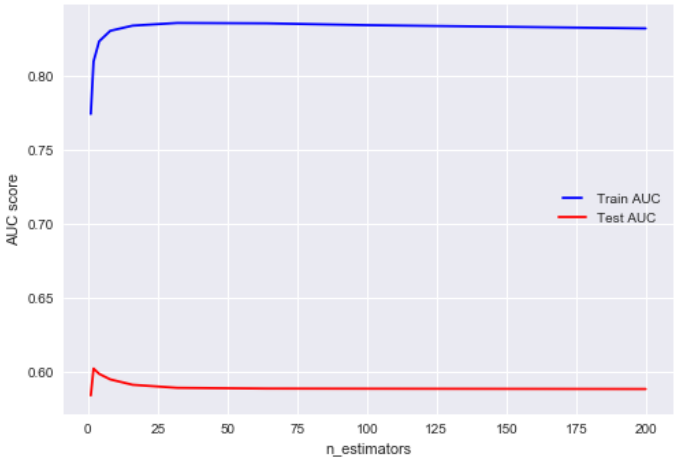


Figure 4.4: Number of estimators Random Forest.

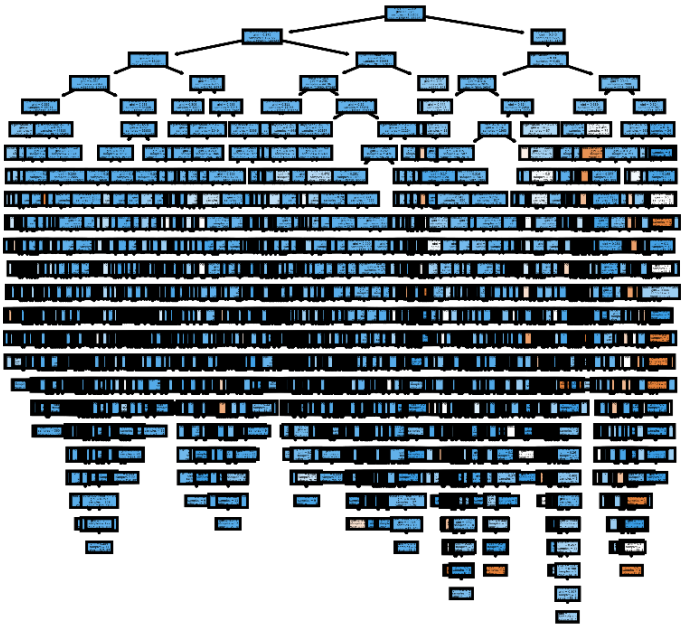


Figure 4.5: Example of one decision tree.

The outcome of the Random Forest is an average of the number of estimators. One of these trees is visualized in Figure 4.5 using only 10 important features. In each node, a decision is made.
The tree is relatively big, which makes it less interpretable compared to the Logistic Regression. Figure 4.6 shows the first branches of the left side of the tree to give an impression of the decisions. The Gini index is a statistical measurement of the inequality of the distribution. It calculates the amount of probability of a specific variable that is classified incorrectly when it is randomly selected.

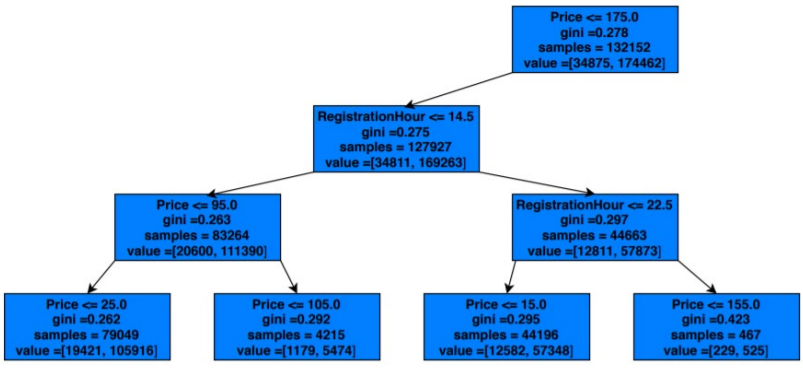


Figure 4.6: First branches of the left side of the Random Forest tree.

The index varies between 0 and 1, where 0 denotes that all elements belong to a certain class and 1 denotes that elements are randomly distributed among different classes. The samples represent the number of observations in the node and the value shows the number of samples in each class (0, 1). The results of the Random Forest tree are discussed in the next chapter.

4.4 Forecasting the timing of a return

After explaining the response variable quantity, we continue with the response variable timing. First, we describe additional data cleaning, followed by the performance measurement and finally the three forecasting methods.

Data cleaning

The input variable *daysBetweenRegistrationAndProcessing* is truncated at 26 days, since it is technically impossible that the time between the request and arrival at the warehouse is higher, we assume this is noise. Therefore, all items with more than 26 days between registration and processing are deleted from the dataset, which represents only a small percentage of the total dataset (less than 1%). Furthermore, only processed items are selected, otherwise the timing of the return has no added value. Next to this, items with a timing of zero are not included in the model.

Performance measurement

The models can be evaluated based on the following often used metrics:

- *R-squared* is the proportion of variation in the outcome that is explained by the predictor variables. The higher the value of R-squared, the better the model.
- *RMSE* measures the average error, by taking the square root of the MSE. The lower the RMSE, the better the model.
- *MSE* measures the average squared difference between the observed actual outcome values and the values predicted by the model.
- *AIC* penalizes inclusion of additional variables to the model. The lower the AIC, the better the model.
- *BIC* is a variant of AIC, with a stronger penalty for including additional variables to the model.

As mentioned in Section 3.3, the MSE is most commonly used for Cross-Validation. Therefore, we will also use this performance measurement to prevent overfitting. We will evaluate the overall performance of the models based on the R-squared values, because we want the highest proportion of the outcome that can be explained by the predictor variables. In addition, we will also compare the Poisson Regression and Negative Binomial Regression on the AIC value. The AIC measures the relative quality of the models, by which a comparison between the models can be made. The code for the models can be found in Appendix F.

4.4.1 Predicting the timing of the return using LASSO Regression

One advantage of using the LASSO Regression for predicting the timing of the return is its automated feature selection. Feature selection is done using a 5-fold Cross-Validation to find the best tuning parameter (alpha), for which the MSE is the lowest. Each curve represents a coefficient in the model. Figure 4.7 visualizes for July how coefficients become non-zero if alpha changes. The higher the tuning parameter, the more coefficients are shrinked towards zero. The tuning parameter regularizes the coefficients such that if the coefficients take large values, the optimization function is penalized. So, if the tuning parameter is equal to zero, the function becomes similar to the Linear Regression cost

function. The feature selection for each month is provided in Appendix H. The weights represent the value of the coefficients. A negative weight suggests that as the independent variable increases, the dependent variable tends to decrease. A zero weight suggests that the variable has no effect on the dependent variable.

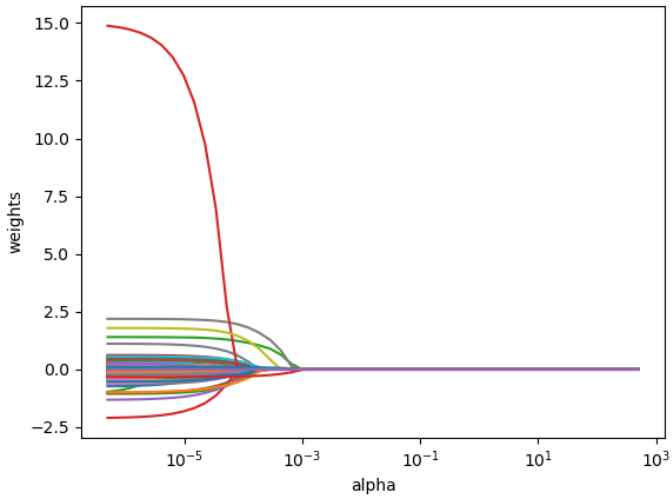


Figure 4.7: Feature selection of LASSO Regression in July.

4.4.2 Predicting the timing of the return using Poisson Regression

The Poisson Regression is also used to predict the timing of the return. The Poisson distribution has the property that its mean and variance are equal. However, this is not true in our case, which is shown in Section 2.3. Therefore, we will also look at a Negative Binomial Regression model. Since the skewness is not large, we will also look at the results of the Poisson Regression. We investigated two extensions of the Poisson Regression, namely Generalized Poisson model 1 (GP1) and Generalized Poisson model 2 (GP2). However, they did not converge with our dataset and are therefore left out of the remainder of this research. Similar to LASSO Regression, we use a 5-fold Cross-Validation to prevent overfitting.

The feature selection from LASSO is also used for both Poisson and Negative Binomial Regression. Which implies that the excluded features mentioned above will also be excluded in those models. The results are discussed in Section 5.1.2.

4.4.3 Predicting the timing of the return using Negative Binomial Regression

A way to prevent overdispersion is to use the Negative Binomial Regression. The variance of the Negative Binomial is equal to the following:

$$Variance = mean + \alpha * mean^2, \tag{13}$$

Where α is automatically set to 1. However, we use a technique called auxiliary OLS regression without constant, to determine the correct value of alpha. The formula that we used is the following:

$$\frac{(y_i-\lambda_i)^2-y_i}{\lambda_i} = \alpha * \lambda_i, \tag{14}$$

Where the Poisson outcomes give the vector of fitted rates λ. Using those rates, the auxiliary OLS Regression model is fitted to the dataset, which provides us the value of α. This optimal value of α will

be used as an input for the predictions of the Negative Binomial Regression. Similar to the other methods, we use also a 5-fold Cross-Validation to prevent overfitting. The results can be found in the next chapter.

4.5 Forecasting the total number of returns per day

This section describes the determination of the total number of returns per day. As shown in Figure 4.1 in Section 4.1, the first step is to classify whether a registered return will be returned. For those truly classified returns, the timing is determined. The outcome of the model for the response variable timing provides at item level the duration of each registered return. Items with equal predicted return dates are summed and a total number of returns will be provided.

However, this total number of returns does not incorporate the zero values and direct returns.
Therefore, the following two modifications should be done:

1. The number of returns should be decreased with a percentage of *zero values*, to incorporate the return requests that will be processed on the same day. This percentage should be based on the registration hour, weekday and month, which is shown in Table 4.5.
2. Besides the zero values, we also need to determine the number of *direct returns* that are returned to the warehouse without registration. Those direct returns are not included in the response variables quantity and timing and need to be calculated to increase the total number of returns. To determine those direct returns, we use historical percentages based only on the month and weekday as shown in Table 2.9 and as explained in Section 2.3.1, due to the unknown registration hour upfront. The number of returns based on the response variables quantity and timing will be multiplied with this direct return percentage per weekday and month to determine the overall number of returns per day.

Registration	*January*							*February*							*March*						
hour	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0	9.09%	1.78%	3.37%	4.05%	3.57%	11.49%	14.71%	20.00%	5.80%	3.14%	2.68%	3.42%	3.54%	12.77%	23.81%	3.10%	1.61%	4.32%	3.64%	3.88%	4.00%
1	11.76%	0.97%	5.26%	6.25%	8.70%	9.30%	10.53%	11.11%	7.04%	6.58%	9.86%	0.00%	7.69%	23.53%	10.53%	1.19%	0.00%	0.00%	1.85%	5.08%	27.27%
2	0.00%	1.61%	6.67%	11.54%	8.33%	15.09%	9.09%	15.38%	0.00%	0.00%	2.25%	0.00%	8.57%	14.29%	25.00%	2.27%	0.00%	4.55%	2.70%	13.33%	7.14%
3	0.00%	4.26%	8.57%	2.78%	4.55%	6.06%	27.78%	60.00%	0.00%	0.00%	3.57%	9.30%	0.00%	16.67%	25.00%	0.00%	0.00%	6.52%	0.00%	0.00%	20.00%
4	0.00%	2.53%	7.14%	7.41%	14.47%	8.33%	33.33%	0.00%	2.38%	5.19%	7.46%	5.68%	8.33%	16.67%	45.45%	4.08%	1.20%	2.99%	1.20%	6.38%	25.00%
5	35.29%	3.02%	3.65%	5.34%	3.70%	5.24%	11.11%	17.65%	0.77%	2.98%	3.25%	2.23%	1.63%	16.00%	16.00%	1.74%	2.26%	2.43%	2.16%	2.36%	16.67%
6	16.13%	1.43%	2.93%	2.90%	3.78%	3.90%	9.21%	13.04%	2.52%	2.86%	2.08%	2.07%	3.76%	9.23%	11.86%	4.34%	1.76%	2.03%	2.18%	2.67%	18.67%
7	17.05%	9.24%	5.42%	8.39%	14.53%	15.87%	30.30%	10.58%	2.33%	2.76%	2.62%	2.57%	5.28%	14.86%	17.65%	5.89%	3.19%	4.15%	3.86%	7.21%	21.67%
8	15.08%	6.92%	7.85%	14.54%	15.43%	12.29%	20.48%	16.27%	13.04%	9.82%	11.40%	9.73%	15.55%	15.52%	11.39%	12.18%	8.77%	7.54%	11.78%	16.70%	18.51%
9	21.50%	8.97%	9.47%	13.64%	15.89%	11.94%	21.15%	17.80%	8.71%	10.81%	11.51%	9.49%	15.33%	16.91%	18.15%	16.04%	8.26%	9.95%	10.95%	14.79%	19.11%
10	19.64%	8.16%	5.35%	11.00%	11.84%	9.54%	16.16%	13.15%	7.99%	7.14%	7.57%	13.03%	12.15%	16.86%	18.12%	11.32%	7.49%	14.22%	11.98%	17.39%	13.65%
11	21.69%	6.86%	8.02%	9.62%	10.28%	11.21%	17.56%	16.26%	10.77%	11.82%	10.82%	12.83%	17.98%	17.92%	18.88%	13.37%	10.57%	13.25%	13.68%	20.67%	17.23%
12	18.10%	6.16%	5.93%	8.82%	9.85%	8.76%	17.59%	14.91%	5.59%	4.89%	4.87%	5.86%	7.74%	14.89%	17.65%	8.03%	5.45%	6.91%	5.25%	8.52%	17.20%
13	20.91%	6.62%	6.43%	9.73%	7.67%	12.01%	15.20%	13.06%	10.82%	10.10%	9.43%	10.69%	16.38%	12.67%	19.54%	13.63%	10.14%	11.76%	11.22%	16.07%	16.18%
14	16.71%	8.37%	4.70%	10.33%	9.05%	8.53%	15.57%	22.34%	11.52%	8.21%	8.08%	9.27%	8.44%	18.56%	16.30%	12.04%	7.14%	7.26%	8.35%	9.27%	15.06%
15	15.29%	6.56%	7.31%	8.40%	6.62%	7.39%	15.77%	14.77%	8.94%	9.94%	8.11%	5.83%	7.39%	12.44%	14.48%	7.98%	9.07%	9.77%	8.48%	6.93%	14.81%
16	14.79%	7.10%	6.92%	10.19%	9.01%	9.14%	15.89%	12.62%	4.81%	5.64%	4.06%	5.21%	4.33%	20.06%	18.95%	4.93%	6.16%	3.94%	4.69%	3.24%	19.41%
17	17.74%	3.43%	6.04%	7.24%	7.70%	6.26%	18.70%	9.09%	2.01%	3.73%	2.16%	2.96%	2.72%	16.54%	18.53%	3.10%	3.93%	2.14%	2.88%	2.49%	14.24%
18	12.01%	5.38%	5.81%	8.61%	8.55%	6.76%	10.75%	14.83%	1.96%	3.47%	2.77%	2.94%	2.77%	12.13%	16.96%	2.18%	3.33%	2.13%	2.31%	2.51%	12.58%
19	17.67%	4.64%	6.14%	6.77%	5.97%	7.08%	13.50%	11.97%	1.89%	2.60%	2.37%	2.29%	3.06%	12.11%	18.14%	2.57%	2.59%	2.51%	2.38%	2.35%	10.60%
20	16.98%	5.65%	5.54%	6.75%	5.05%	8.10%	13.97%	12.18%	1.92%	3.42%	2.35%	2.72%	2.10%	14.38%	16.60%	2.59%	2.45%	2.22%	3.36%	2.40%	10.92%
21	17.62%	7.47%	7.40%	10.28%	5.51%	10.47%	10.16%	10.27%	2.56%	3.05%	2.22%	3.06%	2.59%	10.96%	19.75%	2.39%	3.54%	2.17%	4.10%	1.60%	13.48%
22	19.13%	6.80%	9.76%	13.62%	7.79%	9.50%	17.93%	11.96%	2.37%	2.93%	3.21%	3.39%	4.58%	11.11%	15.58%	2.38%	1.36%	3.08%	2.22%	1.96%	14.46%
23	7.14%	15.37%	11.92%	19.05%	27.61%	19.13%	5.13%	13.04%	0.93%	1.18%	2.22%	1.67%	4.98%	10.42%	14.29%	2.19%	1.15%	1.22%	0.94%	2.20%	7.50%

Registration hour	April							May							June						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0	11.76%	6.00%	0.00%	6.15%	0.00%	7.69%	17.65%	31.58%	1.45%	2.08%	3.77%	3.23%	10.34%	8.33%	12.50%	8.57%	18.39%	8.25%	1.28%	4.55%	7.41%
1	25.00%	2.22%	5.56%	9.26%	0.00%	6.98%	15.38%	16.67%	0.00%	1.75%	1.41%	4.00%	4.08%	9.52%	12.50%	2.78%	3.70%	1.89%	1.96%	3.92%	31.58%
2	0.00%	2.33%	7.14%	8.33%	0.00%	9.38%	11.11%	7.69%	5.08%	4.62%	8.62%	6.78%	2.17%	7.14%	56.67%	2.56%	2.08%	6.56%	1.85%	11.48%	21.43%
3	25.00%	2.63%	0.00%	2.53%	2.99%	0.00%	0.00%	33.33%	0.00%	3.03%	0.00%	6.00%	10.00%	7.69%	30.00%	10.98%	3.19%	3.64%	5.94%	1.85%	9.68%
4	7.69%	1.01%	3.03%	6.33%	3.32%	5.59%	0.00%	20.00%	2.03%	0.99%	4.23%	3.65%	4.26%	13.64%	30.77%	1.65%	3.01%	2.36%	3.08%	2.59%	5.36%
5	9.52%	2.04%	2.13%	3.46%	1.82%	2.41%	18.87%	7.41%	1.43%	3.48%	11.08%	2.30%	4.05%	10.75%	21.21%	1.88%	7.14%	8.65%	4.19%	6.19%	10.67%
6	19.32%	8.83%	9.89%	11.35%	8.10%	12.24%	12.88%	16.18%	9.18%	6.20%	12.25%	10.59%	8.48%	15.79%	20.18%	6.32%	12.80%	10.59%	15.54%	16.75%	29.46%
7	16.52%	16.47%	11.89%	16.11%	11.14%	16.61%	15.60%	14.50%	7.03%	13.84%	11.18%	16.43%	15.83%	14.94%	22.95%	18.42%	12.31%	15.31%	13.55%	14.97%	22.22%
8	14.43%	9.81%	8.77%	11.32%	8.48%	11.53%	12.37%	17.56%	13.72%	5.67%	10.87%	11.38%	13.28%	16.75%	23.70%	16.77%	9.83%	12.56%	10.90%	15.25%	13.87%
9	15.38%	16.82%	15.69%	15.27%	13.17%	13.60%	13.17%	14.83%	10.84%	11.56%	11.65%	13.81%	16.88%	16.52%	19.75%	12.75%	11.60%	10.95%	9.14%	13.95%	21.19%
10	17.14%	11.29%	12.91%	12.26%	7.74%	11.58%	12.62%	11.50%	8.76%	7.84%	10.13%	12.00%	14.19%	14.51%	17.12%	15.06%	9.78%	11.98%	14.29%	11.31%	17.60%
11	18.49%	9.25%	8.58%	11.24%	9.17%	8.86%	13.58%	14.57%	9.32%	7.87%	7.57%	8.45%	9.61%	16.57%	24.63%	9.68%	8.12%	6.96%	8.20%	10.88%	10.12%
12	16.02%	14.74%	15.72%	15.70%	11.35%	10.98%	15.61%	14.86%	10.18%	11.83%	11.30%	13.44%	13.63%	17.82%	19.83%	15.00%	12.09%	8.67%	10.54%	14.89%	9.44%
13	17.90%	12.72%	10.22%	12.67%	6.21%	6.34%	13.11%	14.40%	6.78%	6.67%	9.18%	8.77%	10.67%	13.93%	22.68%	8.90%	10.79%	10.53%	9.31%	8.90%	9.57%
14	17.47%	12.09%	10.99%	11.21%	6.64%	5.61%	10.47%	15.63%	8.51%	8.79%	8.69%	10.65%	9.71%	12.78%	20.56%	12.98%	10.49%	10.70%	10.00%	7.05%	10.76%
15	12.36%	2.64%	4.51%	2.96%	2.44%	2.49%	15.02%	17.86%	3.10%	4.14%	3.74%	4.91%	3.67%	12.96%	22.27%	13.97%	4.61%	6.30%	8.31%	5.23%	8.58%
16	25.63%	2.19%	3.44%	2.29%	2.45%	2.12%	13.36%	13.66%	2.28%	2.26%	2.01%	3.17%	2.74%	13.61%	18.09%	3.48%	3.48%	2.87%	2.43%	2.44%	8.11%
17	17.79%	2.73%	2.17%	2.92%	2.12%	1.47%	10.89%	14.29%	2.93%	1.90%	2.26%	3.52%	1.83%	11.72%	19.70%	3.53%	2.84%	2.48%	3.02%	1.69%	7.90%
18	13.26%	2.25%	2.94%	2.51%	2.15%	1.26%	9.73%	9.45%	1.86%	2.33%	1.95%	2.41%	1.72%	11.33%	23.35%	2.78%	2.99%	2.78%	2.67%	1.73%	6.71%
19	21.25%	2.57%	2.48%	3.48%	1.94%	1.13%	9.55%	13.73%	1.93%	1.93%	2.19%	2.92%	2.37%	10.25%	16.82%	3.23%	1.94%	2.25%	2.49%	2.36%	5.64%
20	19.51%	2.55%	2.78%	2.81%	1.86%	1.27%	5.15%	13.92%	2.38%	1.63%	2.39%	2.22%	1.48%	8.09%	21.35%	2.94%	2.24%	1.62%	2.36%	1.61%	7.58%
21	13.75%	3.40%	3.74%	1.85%	2.53%	1.51%	6.74%	8.54%	1.78%	1.65%	1.78%	2.17%	1.00%	17.76%	20.95%	2.77%	2.21%	1.53%	1.50%	2.72%	5.62%
22	14.29%	2.16%	2.36%	1.28%	1.66%	0.83%	3.45%	14.29%	0.60%	1.02%	2.18%	2.85%	2.16%	8.93%	11.39%	3.62%	0.98%	1.65%	2.15%	1.76%	6.19%
23	21.43%	0.95%	3.64%	2.42%	1.40%	5.93%	14.29%	2.70%	4.70%	0.66%	0.99%	3.10%	4.64%	21.05%	13.33%	7.58%	1.17%	0.00%	2.60%	2.26%	11.76%

Registration hour	July							August							September						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0	60.00%	8.54%	2.35%	2.97%	4.92%	0.00%	14.29%	31.25%	3.80%	5.19%	4.65%	7.53%	4.40%	20.51%	32.00%	0.00%	5.33%	4.60%	8.54%	7.22%	8.70%
1	12.50%	12.77%	4.76%	1.56%	1.64%	0.00%	30.00%	25.00%	3.03%	1.85%	3.77%	3.64%	9.62%	5.00%	18.75%	4.17%	9.09%	3.51%	1.92%	12.50%	10.00%
2	0.00%	0.00%	3.33%	4.62%	9.26%	10.17%	5.26%	50.00%	6.67%	12.50%	8.70%	6.98%	10.34%	16.67%	44.44%	4.69%	6.56%	8.33%	10.64%	6.67%	0.00%
3	15.38%	4.40%	4.17%	2.67%	4.40%	1.09%	7.14%	28.57%	1.49%	9.46%	1.06%	8.47%	5.80%	11.54%	0.00%	2.22%	4.41%	2.56%	1.19%	3.64%	17.86%
4	30.77%	2.10%	4.60%	1.75%	4.89%	1.79%	9.33%	35.00%	1.48%	1.57%	1.60%	5.80%	4.59%	12.73%	10.53%	5.11%	5.56%	3.82%	2.02%	3.30%	7.25%
5	21.43%	1.54%	1.88%	6.65%	6.60%	8.09%	30.43%	25.00%	2.38%	2.21%	2.10%	3.21%	3.99%	28.95%	13.16%	4.93%	10.22%	3.44%	1.54%	3.11%	18.18%
6	20.22%	9.94%	12.98%	14.71%	14.85%	15.45%	20.62%	11.11%	8.10%	7.96%	10.34%	8.71%	12.90%	35.52%	17.16%	13.39%	12.11%	8.79%	12.86%	12.40%	38.54%
7	16.00%	17.26%	12.79%	15.89%	15.52%	14.02%	14.53%	14.22%	17.06%	15.38%	14.63%	15.97%	19.24%	26.89%	20.09%	18.34%	16.71%	15.98%	16.61%	17.12%	19.02%
8	21.18%	15.81%	14.56%	11.06%	10.82%	9.40%	17.04%	19.09%	12.84%	10.40%	8.44%	11.17%	12.49%	22.31%	16.82%	15.60%	16.39%	9.53%	9.75%	11.04%	23.44%
9	23.93%	10.38%	14.08%	12.30%	13.17%	9.69%	12.16%	23.79%	9.57%	13.59%	11.54%	12.65%	15.23%	27.29%	20.16%	14.94%	16.14%	12.19%	13.78%	13.58%	14.70%
10	24.11%	14.87%	12.69%	12.99%	14.24%	10.95%	8.87%	23.23%	10.88%	12.20%	8.05%	11.79%	13.47%	21.03%	24.22%	15.71%	15.05%	11.11%	11.33%	18.86%	14.26%
11	18.29%	10.75%	7.08%	10.66%	10.59%	6.51%	9.29%	20.99%	6.00%	6.09%	9.99%	9.77%	10.05%	9.62%	21.58%	12.37%	13.28%	9.17%	9.78%	9.27%	15.77%
12	28.37%	14.54%	15.02%	13.54%	13.50%	11.03%	10.61%	24.21%	10.18%	12.67%	13.03%	13.21%	14.40%	10.59%	16.55%	16.30%	14.88%	13.22%	13.08%	14.32%	14.27%
13	19.63%	11.77%	10.31%	12.85%	9.24%	9.34%	9.22%	28.99%	8.14%	8.71%	8.79%	10.48%	11.87%	9.60%	22.75%	12.81%	11.19%	10.52%	9.27%	11.99%	13.71%
14	18.45%	8.78%	13.19%	13.28%	12.32%	10.23%	5.47%	22.86%	17.10%	9.64%	8.31%	10.79%	12.05%	12.25%	19.07%	15.29%	12.96%	11.81%	12.39%	15.00%	7.94%
15	21.98%	13.32%	6.62%	3.97%	4.95%	6.59%	8.87%	22.02%	5.47%	3.96%	2.57%	5.95%	4.05%	9.21%	19.67%	6.18%	6.96%	3.83%	4.52%	4.89%	8.56%
16	18.78%	2.58%	4.11%	2.46%	3.71%	2.68%	5.76%	17.86%	3.65%	4.55%	3.18%	4.87%	3.52%	12.18%	18.79%	4.04%	6.42%	3.00%	3.99%	2.14%	7.11%
17	22.32%	3.33%	3.88%	1.89%	2.82%	1.39%	7.23%	21.48%	4.09%	3.50%	2.70%	3.47%	2.24%	8.32%	19.44%	4.39%	5.40%	2.63%	2.49%	1.77%	7.40%
18	23.46%	2.67%	3.69%	2.72%	3.87%	2.21%	5.01%	23.79%	2.26%	3.00%	2.01%	3.78%	2.24%	6.99%	14.96%	3.88%	4.27%	2.51%	1.72%	1.87%	8.44%
19	19.80%	2.50%	3.02%	2.82%	2.77%	2.13%	5.28%	22.78%	2.81%	3.00%	2.26%	3.06%	2.82%	5.68%	18.93%	3.96%	5.57%	2.95%	2.04%	2.68%	7.03%
20	21.91%	2.69%	2.77%	2.61%	2.54%	1.30%	5.38%	15.12%	2.20%	3.16%	3.13%	3.12%	2.97%	5.71%	9.68%	4.89%	4.52%	2.46%	3.13%	1.52%	10.05%
21	19.67%	2.45%	2.74%	2.05%	2.06%	1.28%	6.87%	17.74%	2.65%	3.10%	2.89%	3.23%	2.31%	7.52%	11.04%	2.79%	4.49%	2.88%	1.69%	2.04%	9.13%
22	22.64%	1.62%	2.67%	2.90%	3.04%	1.60%	0.79%	14.08%	2.61%	2.54%	1.74%	1.00%	2.54%	5.22%	14.47%	2.28%	2.78%	3.83%	2.90%	3.57%	14.44%
23	32.14%	4.66%	5.88%	1.88%	3.88%	0.00%	6.25%	8.77%	2.66%	0.55%	2.50%	2.91%	3.09%	6.02%	10.34%	1.37%	4.60%	5.00%	1.85%	2.17%	8.82%

Registration hour	October							November							December						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
0	50.00%	2.70%	4.62%	9.90%	3.70%	6.38%	37.84%	18.00%	3.83%	6.12%	3.65%	3.45%	7.50%	13.33%	18.03%	1.55%	25.07%	6.16%	15.02%	2.69%	17.24%
1	28.57%	5.80%	5.06%	7.35%	13.24%	13.46%	14.29%	18.52%	0.00%	6.11%	7.84%	3.57%	3.49%	22.73%	10.81%	7.18%	13.04%	4.26%	2.14%	4.32%	7.41%
2	11.11%	1.52%	4.11%	12.50%	2.99%	5.66%	28.57%	11.11%	3.49%	3.03%	0.00%	5.88%	11.11%	33.33%	16.13%	2.11%	7.32%	3.13%	1.25%	4.17%	4.00%
3	22.22%	3.92%	2.20%	8.64%	33.33%	4.05%	18.18%	16.67%	0.00%	1.69%	1.85%	5.88%	13.11%	25.00%	16.00%	8.33%	5.88%	3.08%	6.25%	3.88%	18.18%
4	33.33%	1.09%	5.65%	4.63%	2.76%	5.53%	28.57%	18.18%	4.03%	1.68%	5.41%	4.27%	5.49%	11.76%	13.33%	2.86%	5.98%	7.14%	2.46%	1.37%	8.00%
5	7.69%	12.95%	26.78%	18.12%	19.63%	25.42%	19.12%	11.43%	2.31%	2.90%	3.24%	2.34%	4.52%	13.16%	17.86%	2.31%	3.80%	2.60%	2.48%	2.49%	17.95%
6	20.48%	20.42%	19.67%	23.95%	21.38%	22.11%	21.85%	19.05%	4.73%	11.05%	14.58%	9.32%	7.34%	11.11%	14.49%	16.81%	13.76%	9.42%	12.35%	13.89%	24.82%
7	22.60%	13.97%	13.97%	15.56%	13.23%	13.99%	28.52%	19.59%	16.62%	19.85%	18.38%	22.58%	22.07%	20.73%	22.51%	24.36%	25.10%	24.94%	14.68%	15.74%	15.60%
8	18.33%	13.50%	14.76%	11.67%	18.28%	17.36%	18.26%	17.23%	12.85%	10.32%	12.27%	14.39%	19.18%	20.25%	15.30%	12.98%	16.27%	6.89%	6.58%	7.80%	21.01%
9	25.57%	11.89%	9.99%	15.38%	10.08%	13.40%	27.84%	24.11%	13.84%	13.49%	11.17%	9.91%	18.11%	18.99%	18.97%	10.92%	14.06%	14.79%	10.92%	8.10%	16.20%
10	20.34%	9.45%	12.15%	12.68%	13.59%	13.33%	23.98%	20.00%	10.15%	12.89%	15.10%	13.47%	18.38%	19.97%	18.47%	6.59%	12.10%	12.07%	7.37%	8.75%	15.67%
11	26.28%	10.51%	11.89%	12.67%	13.45%	13.36%	21.97%	16.70%	7.58%	6.12%	7.91%	8.73%	9.67%	16.95%	18.43%	12.29%	11.17%	9.85%	4.40%	6.78%	18.44%
12	18.22%	9.68%	11.01%	12.11%	12.39%	10.66%	19.72%	16.37%	9.75%	10.28%	10.78%	12.22%	14.31%	20.47%	15.97%	8.37%	13.36%	12.82%	5.78%	9.07%	17.85%
13	23.13%	5.84%	10.88%	12.67%	11.08%	17.54%	21.56%	17.33%	7.91%	7.48%	10.73%	10.11%	17.19%	17.83%	14.04%	11.27%	7.68%	10.03%	8.58%	5.71%	16.83%
14	15.96%	5.95%	9.75%	10.02%	9.19%	10.50%	20.91%	13.85%	8.69%	7.24%	11.86%	12.59%	11.25%	19.72%	18.40%	10.55%	10.14%	11.67%	6.04%	8.62%	16.75%
15	16.22%	5.59%	9.73%	7.17%	9.70%	11.04%	23.93%	17.63%	8.89%	7.61%	9.70%	11.08%	13.38%	19.37%	16.80%	12.54%	13.28%	9.58%	7.62%	9.47%	12.94%
16	19.92%	11.10%	8.94%	8.51%	8.75%	7.47%	20.46%	13.93%	8.57%	8.87%	12.92%	11.07%	12.59%	18.72%	14.38%	11.67%	9.93%	19.35%	9.42%	10.34%	13.96%
17	18.94%	5.43%	9.97%	7.29%	9.71%	9.16%	20.00%	18.62%	6.21%	5.06%	6.09%	6.50%	6.50%	19.19%	17.26%	9.54%	4.43%	10.61%	4.32%	9.13%	14.42%
18	16.99%	6.32%	8.19%	7.25%	8.79%	10.45%	16.96%	15.18%	8.18%	8.96%	8.59%	8.91%	12.08%	15.76%	13.71%	13.97%	12.31%	15.76%	10.16%	7.72%	12.33%
19	16.47%	6.95%	9.81%	10.75%	9.73%	8.98%	18.46%	16.40%	5.45%	8.33%	7.20%	9.07%	8.75%	16.91%	12.42%	12.49%	10.56%	11.97%	7.98%	9.18%	10.97%
20	13.85%	7.25%	12.04%	11.73%	9.90%	7.76%	16.89%	15.51%	7.14%	10.76%	11.71%	8.59%	7.80%	15.69%	16.38%	14.37%	7.13%	10.03%	7.74%	13.92%	13.67%
21	7.63%	9.54%	12.29%	11.79%	16.31%	7.17%	14.13%	12.19%	9.36%	12.10%	12.78%	10.29%	10.37%	16.25%	14.64%	17.69%	12.54%	22.89%	9.85%	16.83%	8.59%
22	16.98%	15.13%	11.11%	18.46%	12.42%	3.45%	13.79%	12.10%	12.09%	16.04%	16.41%	15.44%	15.98%	10.30%	12.67%	31.46%	30.18%	26.44%	8.67%	23.22%	12.70%
23	22.73%	15.69%	9.73%	8.19%	10.85%	3.55%	8.11%	14.81%	21.53%	26.13%	19.71%	22.89%	24.32%	21.54%	6.78%	22.06%	25.06%	31.43%	32.71%	15.58%	9.91%

Table 4.6: Zero value percentages per Registration hour, weekday and month.

To conclude, we will first predict for each request whether it becomes a return using Logistic Regression and Random Forest with all and only 10 features. After this step, we predict the timing of the return requests using LASSO, Poisson and Negative Binomial Regression. Based on those two predictions, the total number of returns can be retrieved. However, this number is adjusted using the direct return percentages.

5. Model validation

In this chapter, we answer the fourth research question: *'How should the model be validated?'*.

5.1 Performance of the proposed forecasting method

In this section, the results of previous described models are discussed for each response variable.

5.1.1 Response variable quantity returns

For each month, we used the following models for the dataset of 2019:

1. Logistic Regression with all features.
2. Logistic Regression with the 10 most important features.
3. Random Forest tree with all features.
4. Random Forest tree with the 10 most important features.

Model comparison

We will evaluate the models based on the classification's performance, the ROC and AUC curve and the Confusion Matrix as described in Section 4.3.

From Table 5.1 we see the classification's performance using the precision, recall and the F1-score. In the table, the best performing method is highlighted for each month. The scores of the performance are quite high, since 1 indicates a perfect fit. However, the results can be biased due to the high return percentage of return requests as shown in Table 4.5. On average, 81% of the return requests was actually returned. From the classification's performance we can conclude that the model performs better than a simple model that classifies all requests to be returned. Although, the differences are not large and we need a closer look at the Confusion Matrix and AUC curve to gain more insight in the performance of the models. But we first compare the models based on their classification's performance.

Despite the fact that the average precision of Logistic Regression with only 10 features in October is the highest, we can conclude that the Random Forest outperforms the Logistic Regression in all other cases based on the classification's performance. Table 5.2 compares the four methods and indicates the percentage of the number of times the model has the highest average or the best performance per fold for each month based on the F1-score. The Random Forest has the best F1-score on average in 78% of the cases. However, the differences are small in performance. Table 5.2 also visualizes the average F1-scores for each model. Based on these results, we can conclude that the Random Forest with all features performs best. Though, the differences are small.

January											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.822	0.832	0.822	0.820	0.826	0.820	0.838	0.848	0.838	0.828	0.836	0.830
February											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.830	0.846	0.830	0.820	0.826	0.822	0.858	0.868	0.856	0.842	0.854	0.844
March											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.832	0.850	0.832	0.820	0.828	0.824	0.862	0.868	0.858	0.844	0.856	0.846
April											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.830	0.846	0.826	0.822	0.828	0.826	0.852	0.864	0.852	0.836	0.852	0.838
May											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.824	0.842	0.824	0.802	0.820	0.810	0.854	0.866	0.850	0.836	0.850	0.838
June											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.798	0.838	0.774	0.782	0.836	0.762	0.826	0.850	0.822	0.818	0.848	0.818
July											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.780	0.832	0.760	0.716	0.832	0.760	0.798	0.834	0.800	0.800	0.840	0.796
August											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.778	0.830	0.766	0.756	0.830	0.754	0.812	0.840	0.810	0.810	0.842	0.802
September											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.778	0.808	0.752	0.760	0.802	0.752	0.792	0.816	0.788	0.784	0.812	0.782
October											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.778	0.820	0.768	0.832	0.820	0.746	0.820	0.842	0.814	0.818	0.840	0.810
November											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.772	0.818	0.738	0.728	0.818	0.736	0.830	0.848	0.820	0.840	0.852	0.818
December											
Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0.726	0.800	0.712	0.732	0.800	0.710	0.808	0.828	0.800	0.826	0.834	0.796

Table 5.1: Results classification report response variable quantity returns.

MODEL	# BEST AVERAGE	# BEST FOLD	F1-SCORE
LOGISTIC REGRESSION	0%	1%	0.784
LOGISTIC REGRESSION (10)	3%	1%	0.777
RANDOM FOREST	78%	53%	0.826
RANDOM FOREST (10)	19%	45%	0.818

Table 5.2: Comparison model performances classification report response variable quantity.

Figures 5.1 and 5.2 show the AUC curves for each month. The results of the AUC scores differ per month. For example, January shows a better fit compared to July with the highest respectively scores of 0.82 and 0.67. This makes sense due to the percentages of actual returns based on registrations as shown in Table 4.1. The return percentage is the lowest in January and the highest in July. An AUC score of 1 indicates that the model can perfectly distinguish between classes and a score of 0.5 cannot

distinguish. All models show the capability of distinguishing between classes and therefore the prediction of true positives and/or true negatives, but not as good as preferred. We expect this is caused by the prediction of true negatives. This will be clear from the Confusion Matrix. But we will first compare the models based on the results of these AUC scores. From the results, we can see that the Random Forest model with only 10 features outperforms the other models in each month. The difference with the Random Forest model with all features is only 0.01 percent point and the difference with the Logistic Regression models is at most 0.8 percent point. Furthermore, the difference between the Logistic Regression with all and only 10 features is small to nothing.

To look at the predictive power of the models for the true negatives, we investigate the Confusion Matrices. Based on the results from the detailed Confusion Matrix in Appendix G and the summarized Confusion Matrix in Figure 5.3, we can conclude that the model is barely predicting the true negatives. Which we believe is due to the fact that the dataset contains unbalanced data, because around 81% of the requests will be returned. Therefore, the model receives a good accuracy with mainly classifying the requests as 'true'. Which leads to an overestimation of returned requests. The numbers within the Confusion Matrix are stated in percentages due to confidentiality. From the Confusion Matrix, we can conclude that the Random Forest models are better at predicting the true negatives. The Random Forest (10) has a misclassification of 13.8%, compared to 14.1% for the Random Forest with all features, which are mainly caused by wrongly classifying the requests as being returned.

From the results we can conclude that the Random Forest is performing slightly better compared to the Logistic Regression based on the classification report, AUC score and confusion matrix. The Random Forest with all features performs better at the classification report compared to the Random Forest with only the 10 most important features. Although, the Random Forest with only 10 features performs better at the AUC score and the confusion matrix. The Logistic Regression with all features performs slightly better compared to only the 10 most important features. Although, the models are barely predicting the true negatives or false negatives, which implies that the response variable quantity will be overestimated.

Because the differences between the models are small, we prefer the model of Logistic Regression over the Random Forest tree due to the higher interpretability and lower computation time as preferred by Bol.com. The difference between the Logistic Regression with all features and only 10 is small to nothing. Because the outcome of the response variable quantity is complementary to the response variable timing, we prefer all features instead of only 10 for the implementation. Therefore, we will look at the results of the Logistic Regression with all features in more detail.

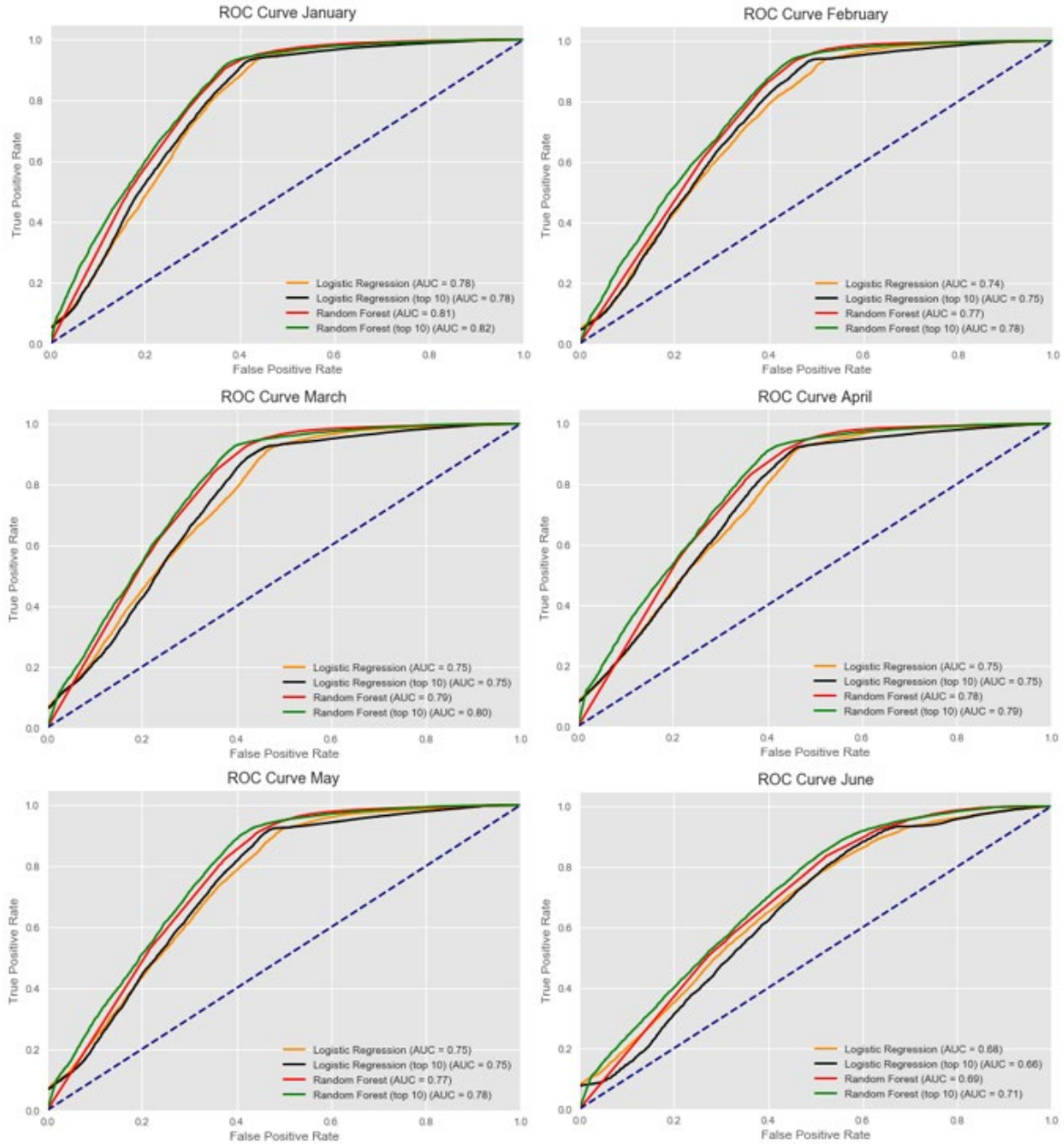


Figure 5.1: ROC and AUC curves report response variable quantity January-June.

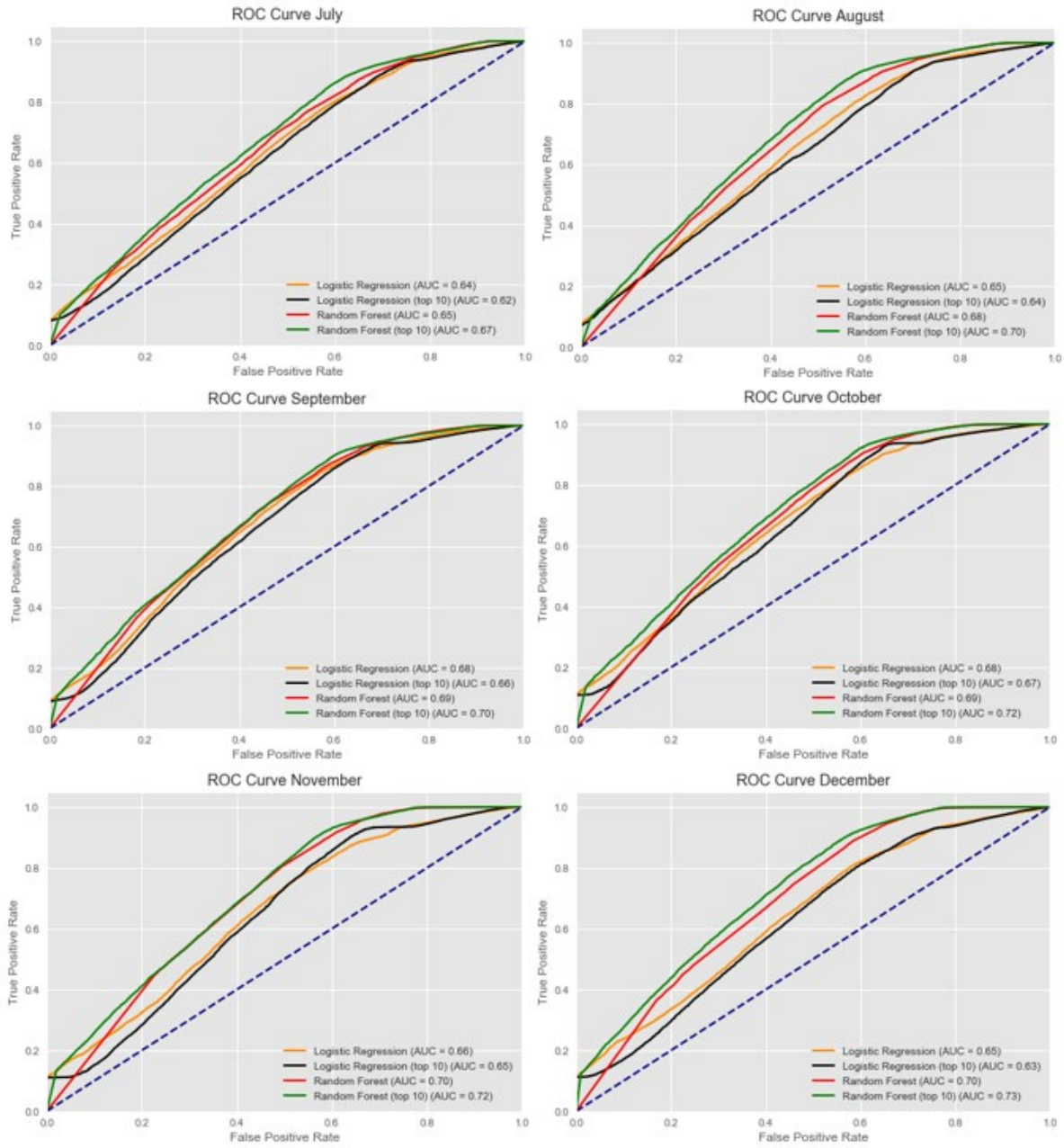


Figure 5.2: ROC and AUC curves report response variable quantity July-December.

Logistic Regression

	Predicted (0)	*Predicted (1)*
True (0)	3.2%	14.2%
True (1)	1.4%	81.2%

Random Forest

	Predicted (0)	*Predicted (1)*
True (0)	6.1%	11.3%
True (1)	2.8%	79.8%

Logistic Regression (10)

	Predicted (0)	*Predicted (1)*
True (0)	2.8%	14.5%
True (1)	1.6%	81.1%

Random Forest (10)

	Predicted (0)	*Predicted (1)*
True (0)	5.9%	11.6%
True (1)	2.2%	80.3%

Figure 5.3: Confusion Matrix report response variable quantity results (%).

Comparison to literature

As mentioned above, we prefer the Logistic Regression model for the response variable quantity. Based on the results of the Logistic Regression, the 10 most important features to classify whether a return request will be returned are:

- Hour of registration (-);
- Day of the week (- on average);
- Price (+);
- All three sources of registration (BLUE (+), DOCDATA (+) and Webshop (+));
- Quantity (-);
- Both selling parties (own (+) and LvB (+));
- The product group Camping and Outdoor hardware (+).

Besides the 10 most important features, reason codes also influence the quantity of returns:

- code_OTHER (+)
- code_WRONG_ARTICLE_ORDERED (+)
- code_WRONG_ARTICLE_RECEIVED (+)
- code_ARTICLE_DAMAGED (+)
- code_ARTICLE_BROKEN (+)
- code_WRONG_SIZE_ORDERED (+)
- code_INCORRECT_PRODUCT_INFORMATION (+)
- code_ARTICLE_DELIVERY_TOO_LATE (-)
- code_NO_REASON_PROVIDED (-)
- code_ARTICLE_INCOMPLETE (+)

We state for each variable whether the feature negatively (-) or positively (+) impacts the actual return. For example, at the end of the week, a return request is less likely to be returned. Besides, only the reason codes about incorrect product information and delay in delivery have a negative impact on whether a return request will be returned.

We compare our findings with the researches from the literature review, who used regression models to forecast returns. Based on Table 3.2 we compare the variables from Hess and Mayhew (1997), Potdar (2009) and Potdar & Rogers (2012), Asdecker & Karl (2018) and Cui, Rajagopalan & Ward (2020) in Table 5.3.

Features that were mentioned in other researches have also proven in our research to influence the returns. As stated by Cui et al. (2020), time components influence the returns. In our case, even more detailed time components influence the quantity of returns, namely the day of the week, but also the registration hour has an impact. Cui et al. (2020) also states the significant impact of the retailer, which can be translated to our selling parties. Even more specific, the source of registration significantly impacts the return process. As stated by Hess at al. (1997) and Asdecker et al. (2018), price positively influences the return quantity, which is also shown by our research. The main effect plot of price on whether a return request is returned is shown in Figure 5.4(a). Furthermore, the quantity of the return requests influences the return process. However, in contrast with Asdecker et al. (2018) the quantity negatively influences the number of returns. But, the negative impact is low. From Figure 5.4(b), we can see that this negative influence is mainly caused by a high quantity that is not returned. The boxplot in Figure 5.5 provides more insights in the quantity effect on the return.

QUANTITY	OWN	HESS AND MAYHEW (1997)	POTDAR (2009) AND POTDAR & ROGERS (2012)	ASDECKER & KARL (2018)	AND CUI, RAJAGOPALAN & WARD (2020)
HOUR OF REGISTRATION	(-)	-	-	-	-
DAY OF THE WEEK	(-)	-		-	-
PRICE	(+)	(+)	-	(+)	-
SOURCES OF REGISTRATION	(+)	-	-	-	Has influence, stated as process/ resources
QUANTITY	(-)	-	-	(+)	Has influence
BOTH SELLING PARTIES	(+)	-	-	-	Has influence, stated as retailer
SOME PRODUCTGROUPS	(+)	-	-	-	-
REASON CODES	(+/-)	-	Has influence	-	-
CUSTOMER CHARACTERISTICS	-	-	-	(-)	-
DELIVERY TIME	-	-	-	(-)	-
ACCOUNT AGE	-	-	-	(+)	-
YEAR	-	-	-	-	Has influence
MONTH	-	-	-	-	Has influence
SALES	-	-	-	-	(+)

Table 5.3: Results response variable quantity compared to the literature review.

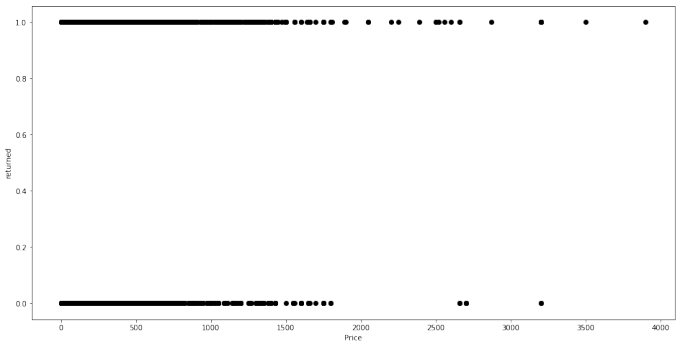


Figure 5.4(a): Effect plot response variable quantity: price.

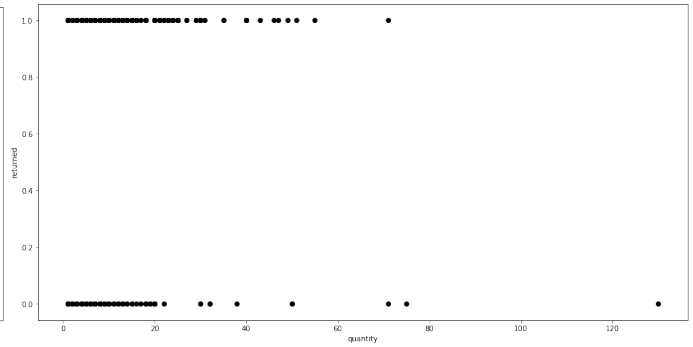


Figure 5.4(b) Effect plot response variable quantity: quantity.

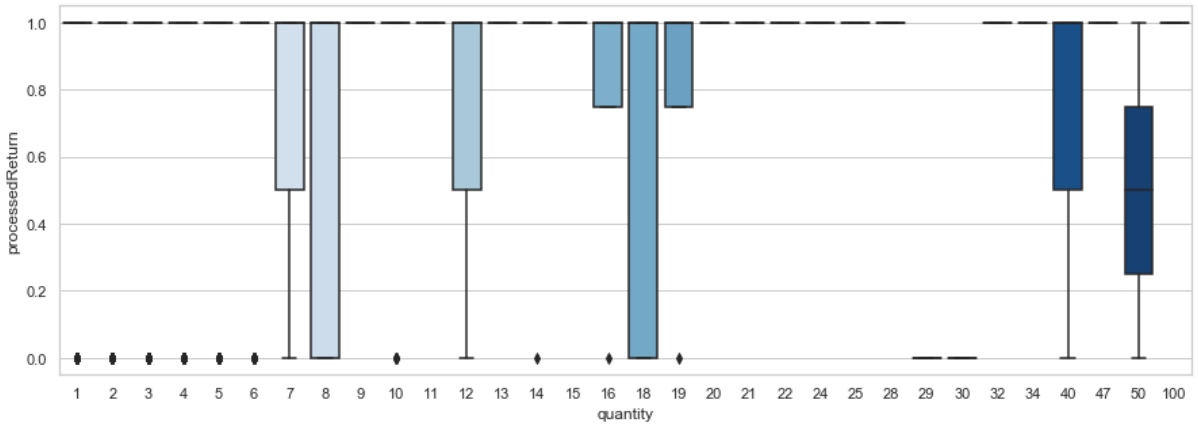


Figure 5.5: Boxplot response variable quantity: effect of the quantity.

Based on the boxplot, we can explain the negative impact of the quantity on whether a request will be returned. Most of the low quantities are returned. Only from 29 items, the chances for a request of being returned are smaller.

5.1.2 Response variable timing

First, we investigate the results of the LASSO Regression, followed by a comparison with the Poisson and Negative Binomial Regression models. The distribution of the duration of the timing is visualized in Figure 5.6 of the months January and July. Appendix I visualizes the histogram of each month. Based on this histogram, we can conclude that the timing does not follow a normal distribution. Because LASSO assumes a normal distribution, we expect LASSO to perform worse than the other two models. The coefficients per feature can be found in Appendix L. In the histograms we see a longer duration in the
summer compared to the winter. An explanation for this could be that people are on holiday and can take their time before returning their package.

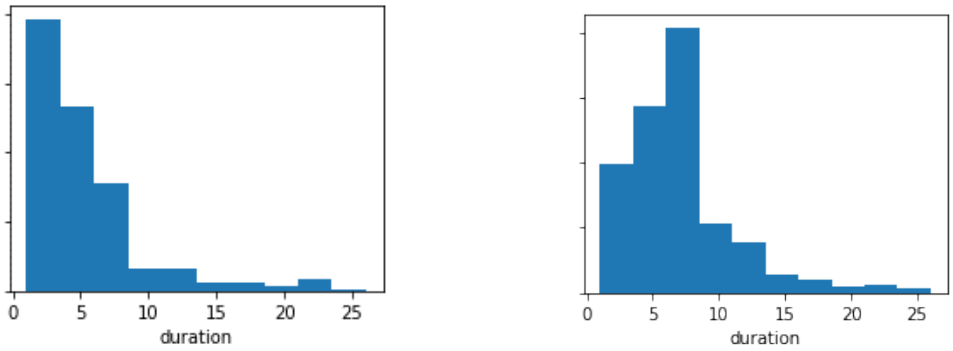


Figure 5.6: Histogram of the response variable timing of the months January and July respectively.

Results response variable timing using LASSO Regression

Based on the 5-fold Cross-Validation, the alpha parameter is tuned. This optimal tuning parameter is used as an input for the LASSO Regression model. Figure 5.7 shows a plot between the predicted and actual timing of returns for July. The black line would show a perfect prediction, which is not true in our case. Above 11 days, hardly any predictions are made. Figure 5.8 shows a residual plot of the training versus the testing data for July. The residuals are calculated by the difference between the predicted duration and the actual duration. The plot does not show major differences between the training and testing data but shows a large deviation of the predicted duration. The zero-base line indicates an exact prediction, but the models show a large deviation. The plots for the remaining months are shown in Appendix J. From those visualizations we see similar results, but small deviation in the forecasted duration per months are visible. For example, the duration in December is on average longer compared to July.

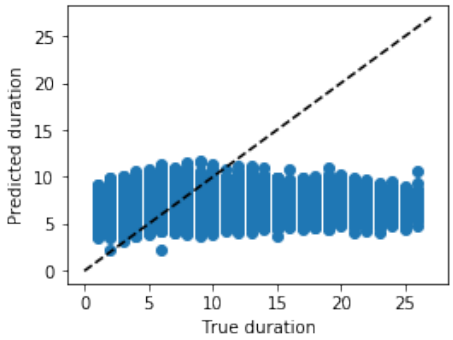


Figure 5.7: Predicted versus actual duration LASSO.

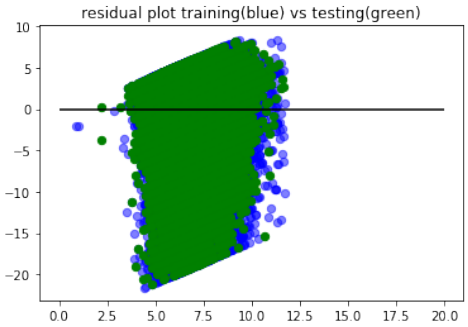


Figure 5.8: Residual plot training versus testing data LASSO.

If we look at the performance of the LASSO in terms of R-Squared, MSE and RMSE, we find the following results shown in Table 5.4. We can see that the performance differs per month, but no large deviation is shown. However, the performance is not good. The value of R-Squared is really low, which means that the timing is hard to predict using the predictor variables. We expect this is due to the assumption of normal distribution and lack of predictor variables regarding the transport and customer behavior.

	TRAINING		**TEST**	
	R-Squared	*RMSE*	*R-Squared*	*RMSE*
JANUARY	0.0615	4.0556	0.0574	4.0371
FEBRUARY	0.0455	4.1782	0.0429	4.1720
MARCH	0.0508	4.1535	0.0450	4.1899
APRIL	0.0626	4.3090	0.0629	4.3183
MAY	0.0607	4.1741	0.0612	4.1374
JUNE	0.0694	3.9031	0.0675	3.8929
JULY	0.0607	3.8648	0.0536	3.8696
AUGUST	0.0617	3.7084	0.0565	3.6734
SEPTEMBER	0.0742	3.9754	0.0744	3.9771
OCTOBER	0.0565	3.9848	0.0526	4.0210
NOVEMBER	0.0467	4.1999	0.0447	4.1751
DECEMBER	0.0667	4.3763	0.0711	4.3454
AVERAGE	0.0597	4.0736	0.0575	4.0674
MAX DEVIATION	0.014	0.365	0.017	0.394

Table 5.4 Results response variable timing using LASSO.

As shown in Figure 5.7 and Figure 5.8, the prediction window is small. The model only predicts the timing to be within 3 and 12 days. To enlarge this window, we use the natural logarithm of the timing. Logarithm transformation is a convenient means of transforming a skewed variable into a more normalized dataset. The histogram visualized in Figure 5.9 shows the transformed data in July. The prediction interval increased, which is shown in Figure 5.10. However, the performance of the model in terms of R-squared only increased with around 1 percent point.

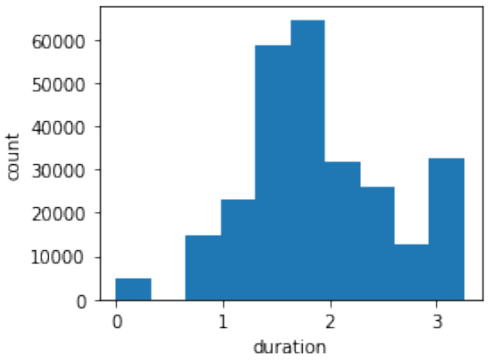


Figure 5.9: Histogram LASSO after transformation

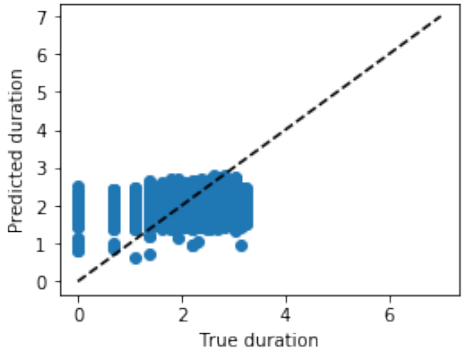


Figure 5.10: Predicted versus actual after transformation LASSO.

Results response variable timing using Poisson and Negative Binomial Regression

Besides the LASSO Regression, we also look at the results of the Poisson and Negative Binomial (NB) Regression results. Because the mean is not equal to the variance, we expect overdispersion in the data. The fraction of Pearson Chi over the degrees of freedom of the Residuals, which can be retrieved from Table 5.5, confirms this with a value higher than 1. If we fit a Negative Binomial Regression to the same dataset, we see a smaller value for the log-likelihood, deviance and Pearson Chi test. Therefore, the Negative Binomial Regression would be a better fit. However, the differences are small. The Degrees of Freedom (DF) of the models are equal, which means both models use the same number of predictors.

Generalized Linear Model Regression Results

Dep. Variable:	duration	No. Observations:	183979
Model:	GLM	Df Residuals:	183907
Model Family:	Poisson	Df Model:	71
Link Function:	log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-5.0998e+05
Date:	Tue, 09 Jun 2020	Deviance:	3.5075e+05
Time:	18:00:26	Pearson chi2:	4.09e+05
No. Iterations:	5		
Covariance Type:	nonrobust		

Table 5.5(a): Results Poisson model summary July 2019.

Generalized Linear Model Regression Results

Dep. Variable:	duration	No. Observations:	183979
Model:	GLM	Df Residuals:	183907
Model Family:	NegativeBinomial	Df Model:	71
Link Function:	log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-4.8214e+05
Date:	Tue, 09 Jun 2020	Deviance:	1.5641e+05
Time:	18:00:26	Pearson chi2:	1.90e+05
No. Iterations:	5		
Covariance Type:	nonrobust		

Table 5.5(b): Results Negative Binomial summary July 2019

If we compare the plots of the predicted versus actual counts per duration in Figure 5.11, only tiny differences are visible between the two models. The scatterplots do not show a straight line, which implies that the prediction is not always good. From the plots in Figure 5.12, we see that the model predicts the timing of returns mainly between 4 and 11 days. The plots represent the month July, the plots of the remaining months are shown in Appendix K. There are no large differences visible within the months.

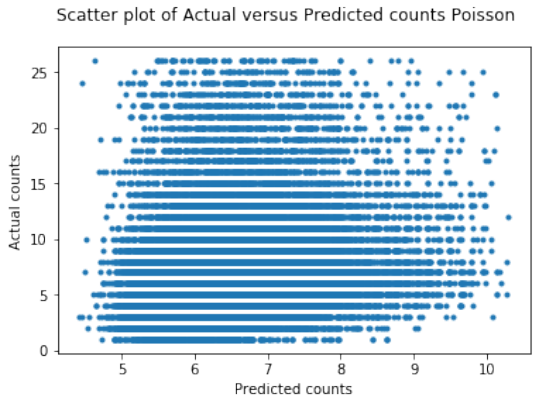


Figure 5.11(a): Scatterplot Poisson.

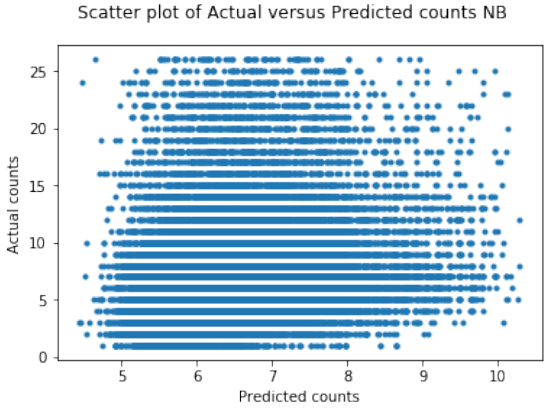


Figure 5.11(b) Scatterplot Negative Binomial.

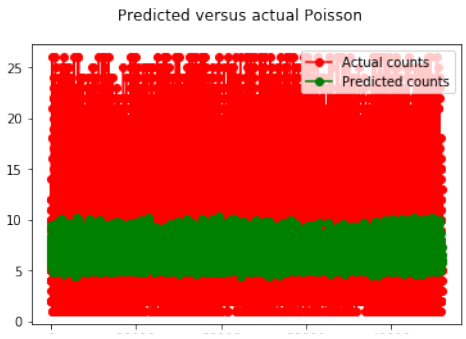


Figure 5.12(a): Plot between predicted versus actual counts.

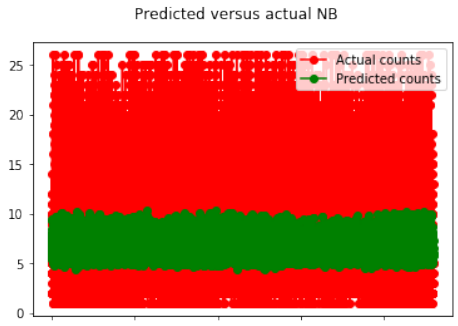


Figure 5.12(b): Plot between predicted versus actual counts.

Figure 5.13 shows the histogram of the duration for the Poisson and Negative Binomial for the month July. The figure shows differences within the decimals of the durations per day. Figure 5.14 shows only small differences between the residual histogram of the models. The residuals are calculated by subtracting the actual durations from the predicted durations. Therefore, a positive number indicates an overestimation of the duration. From the figure, we can conclude that the duration is more often overestimated than underestimated.

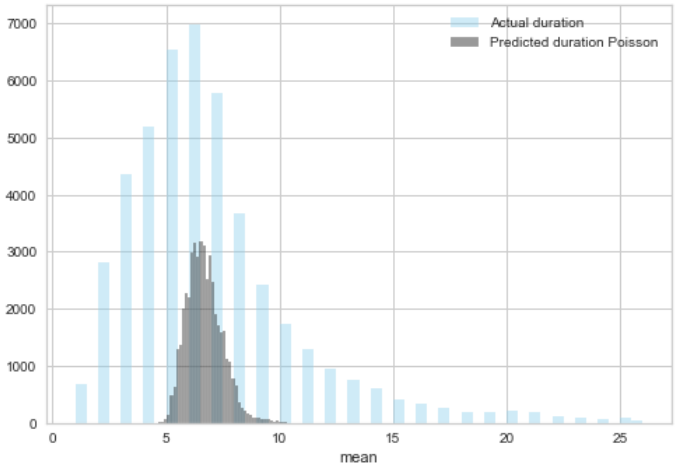


Figure 5.13(a): Histogram of duration Poisson.

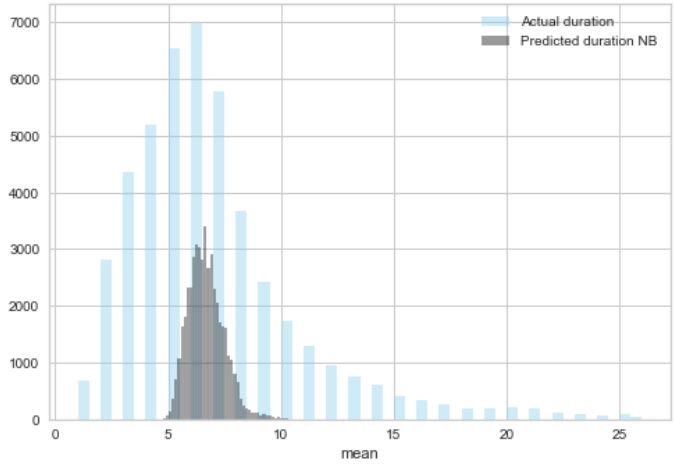


Figure 5.13(b): Histogram of duration Negative Binomial.

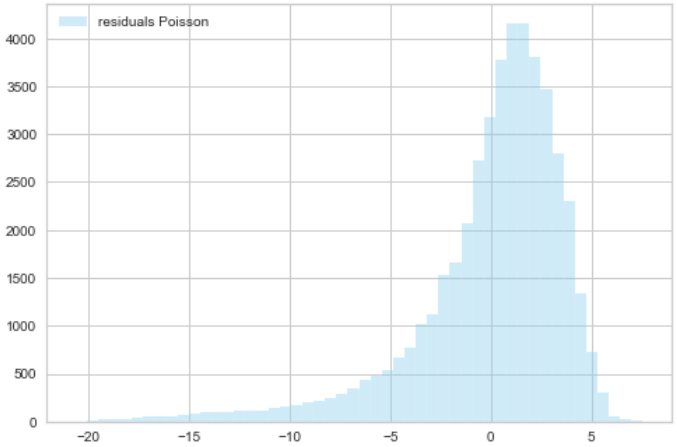


Figure 5.14(a): Residual histogram Poisson.

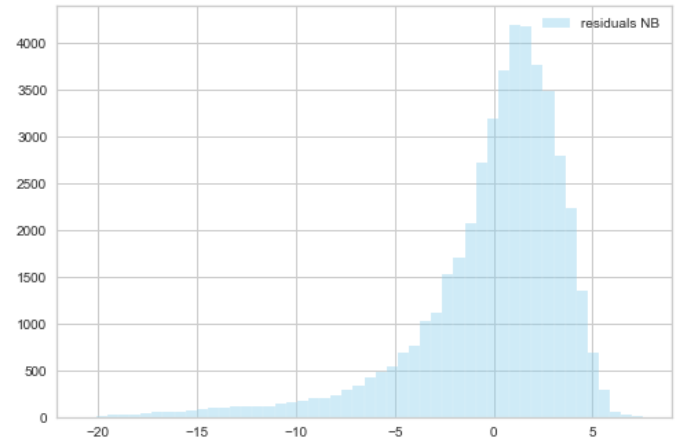


Figure 5.14(b): Residual histogram Negative Binomial.

From Table 5.6 we can conclude that the performance of the Poisson Regression model and Negative Binomial Regression is higher compared to LASSO in terms of R-Squared. Namely a value of around 0.25 compared to 0.06 of Lasso Regression. Although, the R-Squared value remains quite low. Which we believe is due to the fact that no additional data is available regarding the return process of the customer. For example, the delivery of the customer to the PUP, or information regarding the behavior of the customer. If this information would be included in the model, the accuracy would increase. Furthermore, a low R-Squared value is expected, since human behavior in returning a product is harder to predict. From the tables we can conclude that the AIC value of the Negative Binomial Regression is better compared to Poisson Regression. The differences between the AIC values are larger compared to the R-Squared values, because the AIC value is a relative value that uses a maximum likelihood estimation. Which can be explained by how likely one is to see their observed data.

Because Poisson Regression requires less parameter estimation and updating, Bol.com prefers this method over Negative Binomial Regression. Which is due to higher interpretability and lower computation time. However, since the AIC score of Negative Binomial Regression is higher, we will also investigate the impact of the Negative Binomial Regression on the total number of returns.

	POISSON		**NB**	
	AIC	R-Squared	AIC	R-Squared
JANUARY	855847	0.2484	757185	0.2482
FEBRUARY	830476	0.2366	724408	0.2365
MARCH	831656	0.2408	726356	0.2406
APRIL	823549	0.2506	715452	0.2505
MAY	885417	0.2486	802019	0.2485
JUNE	865976	0.2557	807337	0.2556
JULY	963840	0.2478	963840	0.2478
AUGUST	942120	0.2494	910935	0.2494
SEPTEMBER	967706	0.2589	915650	0.2588
OCTOBER	944912	0.2444	858888	0.2443
NOVEMBER	1125846	0.2377	997770	0.2376
DECEMBER	997770	0.2542	1295163	0.2541
AVERAGE	**919593**	**0.2478**	**872917**	**0.2477**
MAX DEVIATION	*206253*	*0.0111*	*422246*	*0.0112*

Table 5.6: Performance results response variable timing using Poisson and NB Regression on average.

As stated with the LASSO Regression, the natural logarithm changes the dataset in a more normalized dataset. Therefore, we test the performance of the Poisson and Negative Binomial Regression also using this transformation. The prediction interval increases as shown in Figure 5.15 for both models.

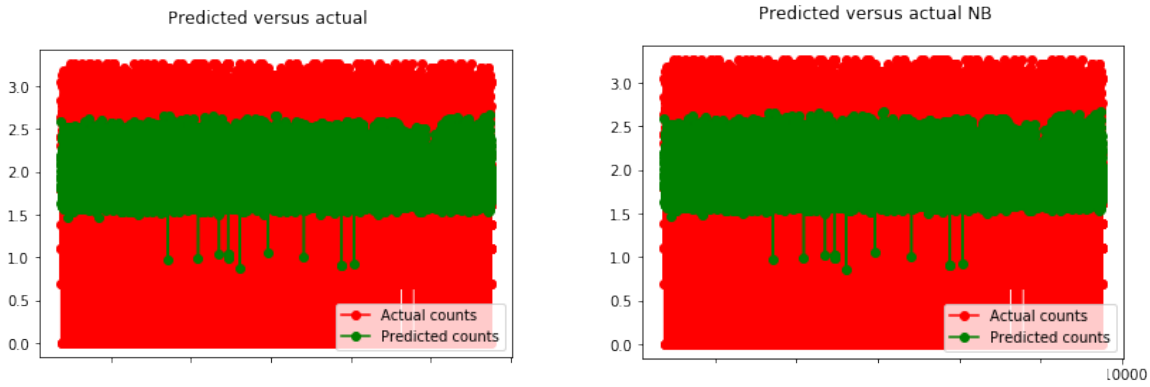


Figure 5.15: Predicted versus actual plot after transformation for Poisson and NB respectively.

Figure 5.16 shows the transformation of the distribution of the duration of Poisson. Based on the figure, the transformation results in a better fit. The results for the Negative Binomial are equal.

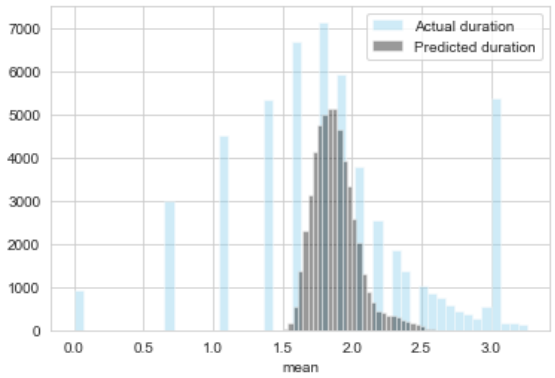


Figure 5.16(a): Histogram after transformation for Poisson.

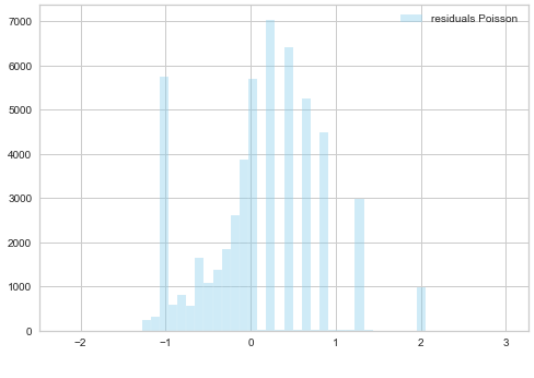


Figure 5.16(b): Histogram of residuals after transformation.

If we look at the performance measurements, we see an improvement. The R-Squared value increased from around 0.25 to 0.27. The AIC value is higher for Negative Binomial, but lower for Poisson Regression. Because the difference is not large, we will not investigate the impact of the Poisson Regression model on the total number of returns using a natural logarithm transformation.

Comparison to literature

We compare the main findings of the features from the Poisson and Negative Binomial Regression with the researches about regression models mentioned in our literature review in Chapter 3. Based on all the folds for each month, the following features have proven to have a significant impact with a p-value of 0.05 in our model:

- Quantity (-);
- Registration hour (+);
- Day of the week (-);
- Both selling parties (own (+) and LvB (+));
- The sources of registration (only BLUE (+) and Webshop (+));
- The reason codes (+);
- Some of the product groups.

Based on the same researches mentioned in the previous Section 5.1.1, we provide an overview of the features that influence the timing of the return in Table 5.7. However, most of them were about the quantity of the return. Although, Hess and Mayhew (1997) could also not find a significant impact of the price on the timing of a return. Furthermore, similar to the research of Potdar (2009) and Potdar & Rogers (2012), reason codes have significant impact on the return timing. The main differences of the features compared to the response variable quantity is the addition of price.

	TIMING
HESS AND MAYHEW (1997)	- Price has no significant impact
POTDAR (2009) AND POTDAR & ROGERS (2012)	- Reason codes influence timing
ASDECKER & KARL (2018)	X
AND CUI, RAJAGOPALAN & WARD (2020)	X

Table 5.7: Results response variable timing compared to the literature review.

From the results we can conclude the following:

- The higher the number of returns per customer, the *shorter* the timing of a return.
- The later the request was registered in hours on a day, the *longer* the timing of a return.
- The later the request registration in the week, the *shorter* the timing of a return. Therefore, we can assume that requests registered in the weekend are likely to be returned sooner.
- The selling parties, sources of registration, reason codes and product groups influence the timing of a return.

To illustrate, Figure 5.17(a) and Figure 5.17(b) show the boxplots of the registration hour and the quantity. The first boxplot provides insights in the registration hour. From the figure we can see that the mean duration increases from 16:00 o'clock. Therefore, on average we can state the earlier the return, the shorter the timing of a return. For the second boxplot we can see the negative impact of quantity on the duration. The higher the quantity of the returns, the shorter the duration.

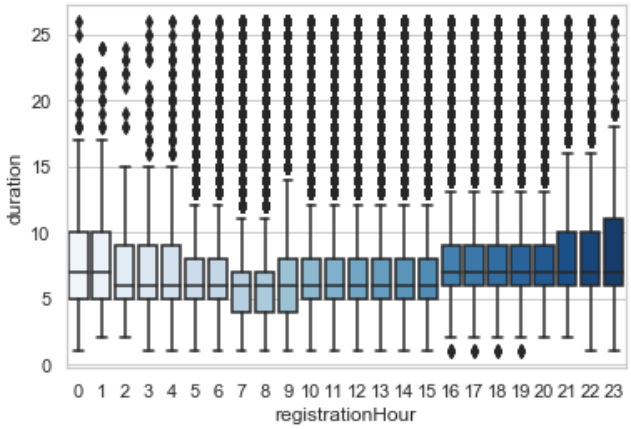


Figure 5.17(a): Boxplot response timing: registration hour.

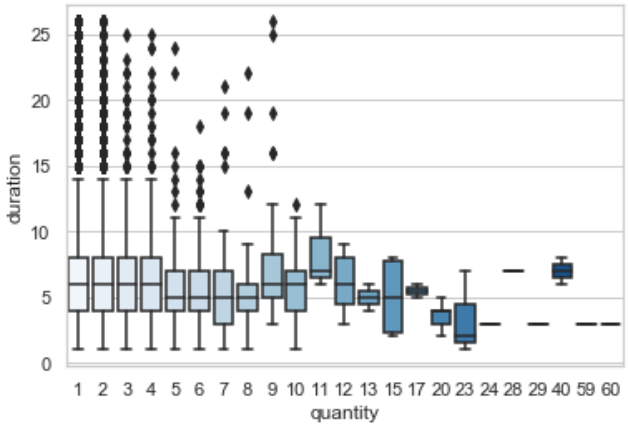


Figure 5.17(b): Boxplot response timing: quantity.

For the reason codes, we looked at the incidence rate ratios to see which reason code has the most influence on the timing of the return in Table 5.8. The rate ratios imply that if the dependent variable is increased by one point, the response variable would be in- or decreased with the incidence rate ratio. Based on the results we can conclude that all reason codes positively influence the timing of a return. The reason codes delivery too late, wrong article received and no reason provided increase the timing of a return the most.

REASON CODES	INCIDENCE RATE RATIO
CODE_ARTICLE_BROKEN	1.08
CODE_ARTICLE_DAMAGED	1.10
CODE_ARTICLE_DELIVERY_TOO_LATE	1.16
CODE_ARTICLE_INCOMPLETE	1.07
CODE_INCORRECT_PRODUCT_INFORMATION	1.08
CODE_NO_REASON_PROVIDED	1.11
CODE_OTHER	1.05
CODE_WRONG_ARTICLE_ORDERED	1.04

CODE_WRONG_ARTICLE_RECEIVED	1.14
CODE_WRONG_SIZE_ORDERED	1.06

Table 5.8: Incidence rate ratios reason codes for response variable timing.

5.1.3 Total number of returns

In order to make the forecast and to see how our forecasting model performs, we need to look at the total number of returns. Without this step, the forecast is incomplete. As shown in Figure 4.1, we need the two complementary models for the response variable quantity and timing as an input for the overall determination of the number of returns per day. In addition, the number of returns per day are increased with the direct returns, using a percentage based on the weekday and month as explained in Section 4.1. In this section, we show the process of the final step in the forecast.

The process of determining the total number of returns based on our dataset is provided in Figure 5.18. Based on the coefficients of the Logistic Regression described and determined in Section 4.3, we can predict for each return request whether the request is true and will be returned to the warehouse. All truly classified returns are an input for the prediction of the timing. Which means, the requests that are predicted not to be returned, are left out of the remainder of the prediction for the total number of returns per day. So, for each truly classified request, the timing is predicted using the Poisson and Negative Binomial Regression coefficients described in Section 4.4. As a result, we have a prediction of the duration for each truly classified request. Based on the duration, we determine the arrival day at the warehouse. We sum all items per day, to receive a total number per day. However, this total number per day should be adjusted with the *direct returns*. Therefore, the total number returns per day should be multiplied with the direct return percentage as described in Section 4.5. As a result, we forecasted the total number of returns per day.

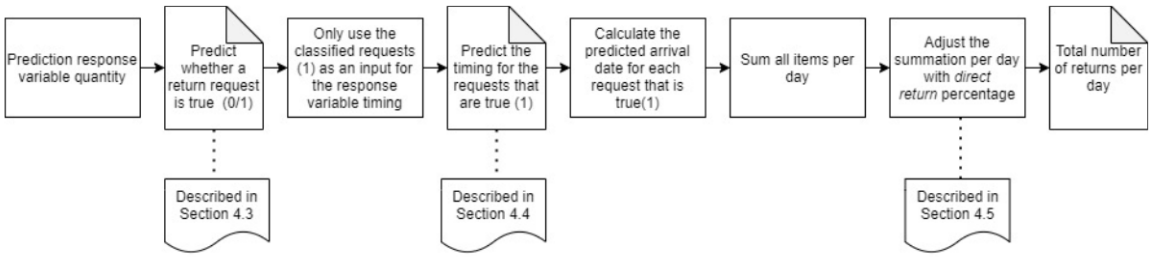


Figure 5.18: Process of determining the total number of returns per day.

If we look at the total number of return items per day, based on Logistic and Poisson Regression, we see the following results in Figure 5.19. The overall MAPE of both the new forecasts is lower compared to the current forecast. The current forecast was based on the hold data of the sales together with the return percentages per cluster. The average MAPE of our forecast is 13.3% for Poisson and 13.5% for Negative Binomial Regression, compared to 15.1% of the current forecast, which shows a significant improvement. Yet, from figure 5.20, we can conclude that the number of returns is overestimated, which is mainly due to the Logistic Regression results. The classification of returned requests is in almost all cases predicted as true. Which is an explanation of why the number of returns is overestimated.

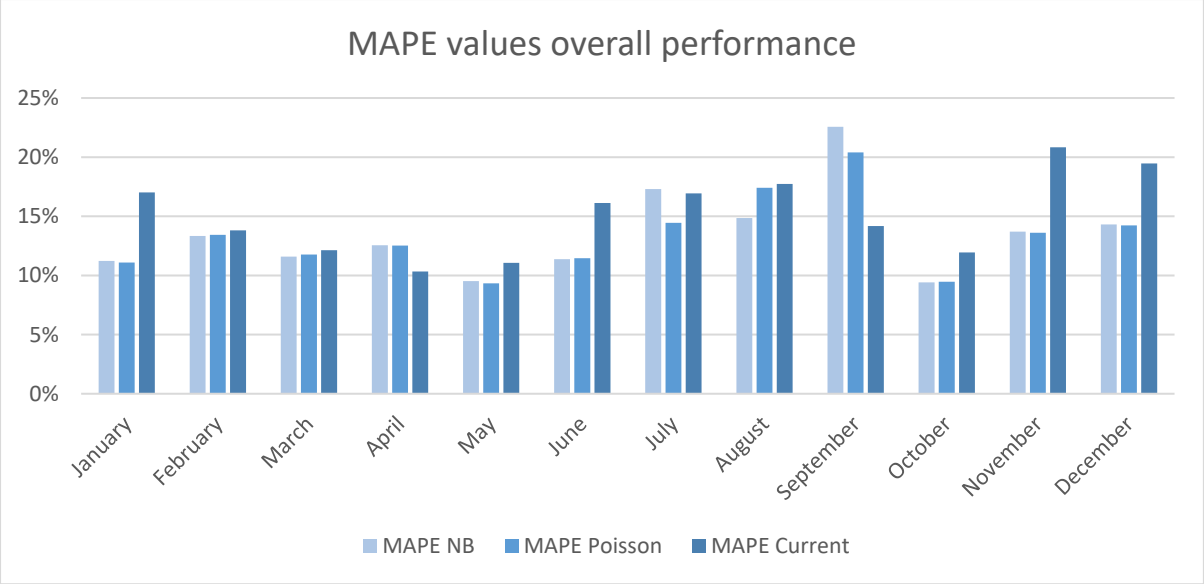


Figure 5.19: MAPE performance per month based on processing date.

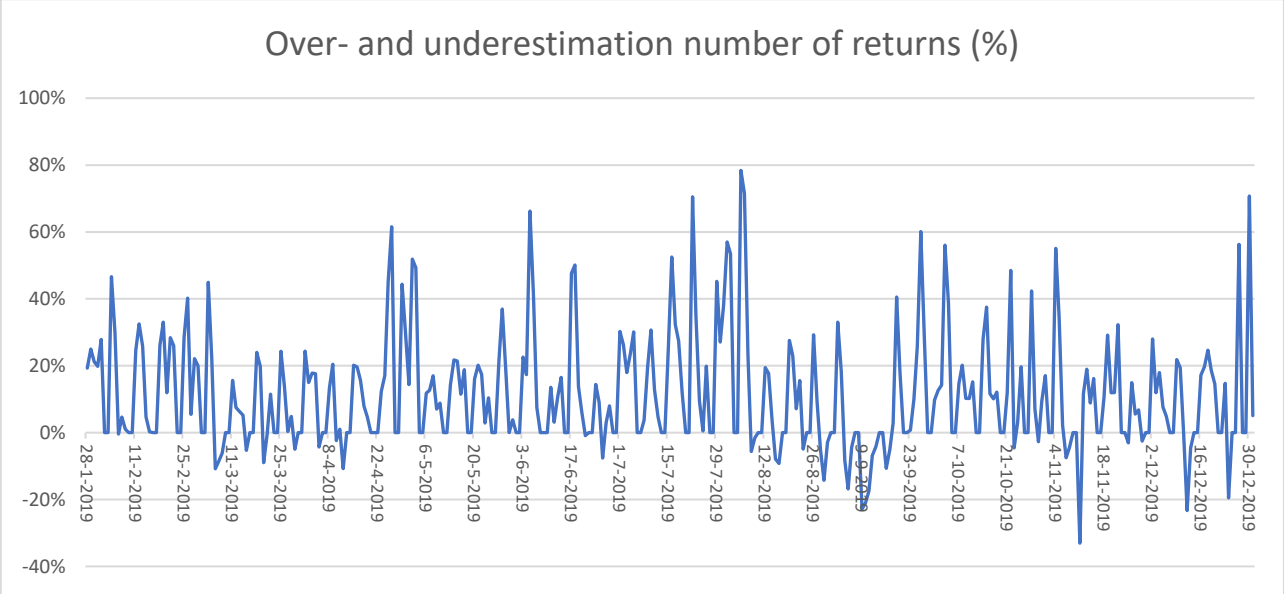


Figure 5.20: Over- and underestimation from the Poisson Regression of the number of returns for 2019.

5.1.4 Conclusion

Based on the results of the best performing forecasting methods for the response variables quantity and timing, we determined the total number of return items per day.

For the classification of the return requests, the Random Forest tree has a higher accuracy compared to the Logistic Regression. However, the differences were only small. As we mentioned in Section 3.2, we prefer higher interpretability, which is the case for the Logistic Regression. Hence, we used the Logistic Regression for the classification of return requests. However, the method is not good at predicting the true negatives, which leads to an overestimation of the number of requests that will be returned.

For the response variable timing, the results from the Poisson and Negative Binomial outperformed LASSO Regression. The differences between Poisson and Negative Binomial are small and almost nothing for the R-squared, but the AIC value was better for Negative Binomial. Since Negative Binomial requires one more parameter estimation, Poisson Regression has a higher interpretability. Therefore, we will test the performance for both Poisson and Negative Binomial Regression for the timing of returns. The transformation using the natural logarithm did not have much impact and only results in a slightly better performance.

For the overall predicted number of return items per day, we combined the models of the response variable quantity and timing. The overall performance was the highest, when we used the Logistic Regression with the Poisson Regression. Compared to the current forecasting method, the overall MAPE decreased with 12.12%.

5.2 Validation and verification

After all these results, we also check if the results can be validated and verified. We validated the used models based on a 5-fold Cross-Validation. If the performance measurements remain stable for each fold, we assume the model to be validated. The codes are verified individually and run multiple times and show no errors. In addition, we test the predictive power of the used models with a different planning window, using aggregated return requests per day instead of each request individually, to see if the performance increases.

5.2.1 Response variable quantity returns

The models are validated through a 5-fold Cross-Validation. As mentioned in the previous section, the classification's performance is measured using precision, recall and the F1-score for each fold, which is shown in Tables 5.9, 5.10 and 5.11. From the tables we can see that the results are stable and do not deviate much. The maximum deviation is 0.034 for the recall value in January. The highest values for all 5 folds per measurement are light-blue colored. The highest average scores are visualized with dark blue. Due to the low deviation per fold, we assume the models for the response variable quantity to be validated.

January												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.810	0.820	0.810	0.810	0.810	0.810	0.830	0.840	0.830	0.820	0.820	0.820
Fold 2	0.800	0.810	0.800	0.790	0.800	0.790	0.820	0.830	0.820	0.800	0.810	0.800
Fold 3	0.820	0.830	0.820	0.820	0.830	0.820	0.840	0.850	0.840	0.820	0.840	0.830
Fold 4	0.830	0.840	0.830	0.830	0.830	0.830	0.840	0.850	0.840	0.840	0.850	0.840
Fold 5	0.850	0.860	0.850	0.850	0.860	0.850	0.860	0.870	0.860	0.860	0.860	0.860
Average	**0.822**	**0.832**	**0.822**	**0.820**	**0.826**	**0.820**	**0.838**	**0.848**	**0.838**	**0.828**	**0.836**	**0.830**
Max deviation	**0.028**	**0.028**	**0.028**	**0.030**	**0.034**	**0.030**	**0.022**	**0.022**	**0.022**	**0.032**	**0.026**	**0.030**
February												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.820	0.830	0.820	0.800	0.810	0.810	0.840	0.850	0.840	0.830	0.840	0.830
Fold 2	0.810	0.830	0.810	0.800	0.800	0.800	0.850	0.860	0.840	0.820	0.840	0.830
Fold 3	0.830	0.850	0.830	0.820	0.830	0.820	0.860	0.870	0.860	0.840	0.850	0.840
Fold 4	0.840	0.850	0.840	0.830	0.830	0.830	0.860	0.870	0.860	0.850	0.860	0.850
Fold 5	0.850	0.870	0.850	0.850	0.860	0.850	0.880	0.890	0.880	0.870	0.880	0.870
Average	**0.830**	**0.846**	**0.830**	**0.820**	**0.826**	**0.822**	**0.858**	**0.868**	**0.856**	**0.842**	**0.854**	**0.844**
Max deviation	**0.020**	**0.024**	**0.020**	**0.030**	**0.034**	**0.028**	**0.022**	**0.022**	**0.024**	**0.028**	**0.026**	**0.026**
March												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.820	0.830	0.820	0.800	0.810	0.810	0.860	0.860	0.850	0.830	0.840	0.830
Fold 2	0.810	0.840	0.810	0.800	0.800	0.800	0.850	0.860	0.850	0.830	0.840	0.830
Fold 3	0.840	0.850	0.840	0.820	0.830	0.830	0.860	0.870	0.860	0.840	0.860	0.840
Fold 4	0.840	0.860	0.840	0.830	0.840	0.830	0.860	0.870	0.860	0.850	0.860	0.860
Fold 5	0.850	0.870	0.850	0.850	0.860	0.850	0.880	0.880	0.870	0.870	0.880	0.870
Average	**0.832**	**0.850**	**0.832**	**0.820**	**0.828**	**0.824**	**0.862**	**0.868**	**0.858**	**0.844**	**0.856**	**0.846**
Max deviation	**0.022**	**0.020**	**0.022**	**0.030**	**0.032**	**0.026**	**0.018**	**0.012**	**0.012**	**0.026**	**0.024**	**0.024**
April												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.830	0.840	0.820	0.810	0.820	0.820	0.840	0.850	0.840	0.820	0.840	0.820
Fold 2	0.810	0.830	0.810	0.800	0.800	0.800	0.840	0.860	0.840	0.820	0.840	0.820
Fold 3	0.830	0.850	0.830	0.830	0.840	0.830	0.860	0.870	0.860	0.840	0.850	0.840
Fold 4	0.830	0.850	0.830	0.830	0.830	0.830	0.860	0.870	0.860	0.840	0.860	0.850
Fold 5	0.850	0.860	0.840	0.840	0.850	0.850	0.860	0.870	0.860	0.860	0.870	0.860
Average	**0.830**	**0.846**	**0.826**	**0.822**	**0.828**	**0.826**	**0.852**	**0.864**	**0.852**	**0.836**	**0.852**	**0.838**
Max deviation	**0.020**	**0.016**	**0.016**	**0.022**	**0.028**	**0.026**	**0.012**	**0.014**	**0.012**	**0.024**	**0.018**	**0.022**

Table 5.9: Outcomes classification report response variable quantity: January-April.

May												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.820	0.830	0.820	0.790	0.810	0.790	0.850	0.860	0.850	0.830	0.840	0.830
Fold 2	0.810	0.830	0.810	0.790	0.800	0.800	0.840	0.860	0.840	0.820	0.840	0.830
Fold 3	0.820	0.840	0.820	0.800	0.820	0.810	0.860	0.870	0.850	0.830	0.850	0.830
Fold 4	0.830	0.850	0.830	0.810	0.830	0.820	0.860	0.870	0.850	0.840	0.850	0.840
Fold 5	0.840	0.860	0.840	0.820	0.840	0.830	0.860	0.870	0.860	0.860	0.870	0.860
Average	**0.824**	**0.842**	**0.824**	**0.802**	**0.820**	**0.810**	**0.854**	**0.866**	**0.850**	**0.836**	**0.850**	**0.838**
Max deviation	**0.016**	**0.018**	**0.016**	**0.018**	**0.020**	**0.020**	**0.014**	**0.006**	**0.010**	**0.024**	**0.020**	**0.022**
June												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.780	0.820	0.750	0.770	0.820	0.740	0.820	0.840	0.810	0.800	0.830	0.800
Fold 2	0.800	0.840	0.780	0.820	0.840	0.760	0.820	0.850	0.820	0.800	0.840	0.810
Fold 3	0.800	0.840	0.780	0.830	0.840	0.770	0.820	0.850	0.820	0.820	0.850	0.820
Fold 4	0.810	0.850	0.780	0.780	0.840	0.770	0.830	0.850	0.830	0.830	0.860	0.830
Fold 5	0.800	0.840	0.780	0.710	0.840	0.770	0.840	0.860	0.830	0.840	0.860	0.830
Average	**0.798**	**0.838**	**0.774**	**0.782**	**0.836**	**0.762**	**0.826**	**0.850**	**0.822**	**0.818**	**0.848**	**0.818**
Max deviation	**0.018**	**0.018**	**0.024**	**0.072**	**0.016**	**0.022**	**0.014**	**0.010**	**0.012**	**0.022**	**0.018**	**0.018**
July												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.770	0.820	0.750	0.680	0.820	0.750	0.790	0.830	0.790	0.780	0.830	0.780
Fold 2	0.790	0.830	0.750	0.680	0.830	0.750	0.790	0.820	0.790	0.790	0.830	0.790
Fold 3	0.760	0.830	0.760	0.740	0.830	0.760	0.790	0.830	0.800	0.800	0.840	0.790
Fold 4	0.780	0.840	0.770	0.770	0.840	0.770	0.810	0.840	0.810	0.810	0.850	0.810
Fold 5	0.800	0.840	0.770	0.710	0.840	0.770	0.810	0.850	0.810	0.820	0.850	0.810
Average	**0.780**	**0.832**	**0.760**	**0.716**	**0.832**	**0.760**	**0.798**	**0.834**	**0.800**	**0.800**	**0.840**	**0.796**
Max deviation	**0.020**	**0.012**	**0.010**	**0.054**	**0.012**	**0.010**	**0.012**	**0.016**	**0.010**	**0.020**	**0.010**	**0.016**
August												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.770	0.820	0.750	0.750	0.820	0.740	0.800	0.830	0.800	0.790	0.830	0.790
Fold 2	0.770	0.830	0.760	0.770	0.830	0.750	0.810	0.840	0.810	0.800	0.840	0.800
Fold 3	0.780	0.830	0.770	0.860	0.830	0.760	0.810	0.840	0.810	0.810	0.840	0.800
Fold 4	0.770	0.830	0.770	0.690	0.830	0.750	0.810	0.840	0.810	0.810	0.840	0.800
Fold 5	0.800	0.840	0.780	0.710	0.840	0.770	0.830	0.850	0.820	0.840	0.860	0.820
Average	**0.778**	**0.830**	**0.766**	**0.756**	**0.830**	**0.754**	**0.812**	**0.840**	**0.810**	**0.810**	**0.842**	**0.802**
Max deviation	**0.022**	**0.010**	**0.016**	**0.104**	**0.010**	**0.016**	**0.018**	**0.010**	**0.010**	**0.030**	**0.018**	**0.018**

Table 5.10: Outcomes classification report response variable quantity May-August

	Logistic Regression			Logistic Regression (10)			Random Forest			Random Forest (10)		
September												
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.770	0.800	0.740	0.750	0.790	0.740	0.780	0.810	0.780	0.770	0.800	0.770
Fold 2	0.770	0.800	0.740	0.750	0.790	0.750	0.790	0.810	0.780	0.770	0.800	0.770
Fold 3	0.780	0.810	0.760	0.760	0.810	0.750	0.800	0.820	0.790	0.790	0.820	0.790
Fold 4	0.770	0.800	0.750	0.760	0.800	0.750	0.780	0.810	0.780	0.780	0.810	0.780
Fold 5	0.800	0.830	0.770	0.780	0.820	0.770	0.810	0.830	0.810	0.810	0.830	0.800
Average	**0.778**	**0.808**	**0.752**	**0.760**	**0.802**	**0.752**	**0.792**	**0.816**	**0.788**	**0.784**	**0.812**	**0.782**
Max deviation	**0.022**	**0.022**	**0.018**	**0.020**	**0.018**	**0.018**	**0.018**	**0.014**	**0.022**	**0.026**	**0.018**	**0.018**
October												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.770	0.810	0.760	0.800	0.810	0.730	0.810	0.830	0.800	0.800	0.820	0.790
Fold 2	0.770	0.810	0.750	0.820	0.810	0.730	0.810	0.830	0.800	0.800	0.830	0.800
Fold 3	0.780	0.820	0.770	0.830	0.820	0.750	0.830	0.850	0.820	0.820	0.840	0.810
Fold 4	0.770	0.820	0.770	0.850	0.820	0.750	0.820	0.840	0.820	0.830	0.850	0.820
Fold 5	0.800	0.840	0.790	0.860	0.840	0.770	0.830	0.860	0.830	0.840	0.860	0.830
Average	**0.778**	**0.820**	**0.768**	**0.832**	**0.820**	**0.746**	**0.820**	**0.842**	**0.814**	**0.818**	**0.840**	**0.810**
Max deviation	**0.022**	**0.020**	**0.022**	**0.032**	**0.020**	**0.024**	**0.010**	**0.018**	**0.016**	**0.022**	**0.020**	**0.020**
November												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.770	0.810	0.720	0.740	0.800	0.720	0.820	0.840	0.810	0.820	0.840	0.800
Fold 2	0.770	0.810	0.720	0.650	0.810	0.720	0.820	0.840	0.810	0.830	0.840	0.810
Fold 3	0.790	0.820	0.740	0.670	0.820	0.740	0.830	0.850	0.820	0.840	0.850	0.820
Fold 4	0.760	0.820	0.750	0.800	0.820	0.740	0.830	0.850	0.820	0.850	0.860	0.820
Fold 5	0.770	0.830	0.760	0.780	0.840	0.760	0.850	0.860	0.840	0.860	0.870	0.840
Average	**0.772**	**0.818**	**0.738**	**0.728**	**0.818**	**0.736**	**0.830**	**0.848**	**0.820**	**0.840**	**0.852**	**0.818**
Max deviation	**0.018**	**0.012**	**0.022**	**0.078**	**0.022**	**0.024**	**0.020**	**0.012**	**0.020**	**0.020**	**0.018**	**0.022**
December												
	Logistic Regression			**Logistic Regression (10)**			**Random Forest**			**Random Forest (10)**		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Fold 1	0.700	0.790	0.700	0.700	0.790	0.690	0.800	0.820	0.790	0.800	0.820	0.780
Fold 2	0.710	0.780	0.690	0.730	0.780	0.690	0.800	0.820	0.790	0.810	0.820	0.780
Fold 3	0.730	0.800	0.710	0.760	0.800	0.710	0.800	0.820	0.800	0.820	0.830	0.800
Fold 4	0.730	0.810	0.720	0.720	0.810	0.720	0.810	0.830	0.800	0.840	0.840	0.800
Fold 5	0.760	0.820	0.740	0.750	0.820	0.740	0.830	0.850	0.820	0.860	0.860	0.820
Average	**0.726**	**0.800**	**0.712**	**0.732**	**0.800**	**0.710**	**0.808**	**0.828**	**0.800**	**0.826**	**0.834**	**0.796**
Max deviation	**0.034**	**0.020**	**0.028**	**0.032**	**0.020**	**0.030**	**0.022**	**0.022**	**0.020**	**0.034**	**0.026**	**0.024**

Table 5.11: Outcomes classification report response variable quantity September-December.

5.2.2 Response variable timing

For the response variable timing, the results from the Poisson and Negative Binomial outperformed LASSO Regression. The results for each fold of the Poisson and Negative Binomial Regression models are shown in Table 5.12 to see if the results are stable. Based on the results, we do not see large deviations per fold and assume the results to be validated.

			FOLD 1	***FOLD 2***	***FOLD 3***	***FOLD 4***	***FOLD 5***	**AVERAGE**
JANUARY	AIC	Poisson	856995	861351	854506	847467	858915	855847
	AIC	Negative Binomial	757432	761092	755641	753500	758261	757185
	R-Squared	Poisson	0.2457	0.2336	0.2541	0.2694	0.2394	0.2484
	R-Squared	Negative Binomial	0.2454	0.2334	0.2539	0.2691	0.2391	0.2482
FEBRUARY	AIC	Poisson	833309	834232	829671	823395	831773	830476
	AIC	Negative Binomial	725303	727649	723249	720830	725009	724408
	R-Squared	Poisson	0.228	0.226	0.240	0.256	0.233	0.237
	R-Squared	Negative Binomial	0.228	0.225	0.240	0.256	0.233	0.236
MARCH	AIC	Poisson	837360	836225	833331	825423	825942	831656
	AIC	Negative Binomial	728039	729284	725986	723998	724472	726356
	R-Squared	Poisson	0.226	0.228	0.238	0.257	0.254	0.241
	R-Squared	Negative Binomial	0.226	0.228	0.238	0.257	0.254	0.241
APRIL	AIC	Poisson	826723	826765	827008	815548	821701	823549

	AIC	Negative Binomial	716095	718500	715294	712451	714920	715452
	R-Squared	Poisson	0.245	0.241	0.243	0.270	0.255	0.251
	R-Squared	Negative Binomial	0.244	0.240	0.243	0.270	0.254	0.250
MAY	AIC	Poisson	888383	889280	888640	877558	883226	885417
	AIC	Negative Binomial	802391	805298	801812	799178	801414	802019
	R-Squared	Poisson	0.241	0.237	0.241	0.269	0.255	0.249
	R-Squared	Negative Binomial	0.241	0.237	0.241	0.269	0.255	0.249
JUNE	AIC	Poisson	869189	867456	866893	858472	867869	865976
	AIC	Negative Binomial	807924	809829	806541	804148	808243	807337
	R-Squared	Poisson	0.247	0.252	0.253	0.277	0.250	0.256
	R-Squared	Negative Binomial	0.247	0.251	0.253	0.277	0.250	0.256
JULY	AIC	Poisson	1020831	1020095	1021984	1015170	1022856	1020187
	AIC	Negative Binomial	963506	964365	963450	962715	965165	963840
	R-Squared	Poisson	0.247	0.247	0.244	0.260	0.240	0.248
	R-Squared	Negative Binomial	0.247	0.247	0.244	0.260	0.240	0.248
AUGUST	AIC	Poisson	943286	945993	944389	933877	943053	942120
	AIC	Negative Binomial	911267	913796	911072	907195	911346	910935
	R-Squared	Poisson	0.246	0.239	0.243	0.273	0.247	0.249
	R-Squared	Negative Binomial	0.246	0.239	0.243	0.273	0.247	0.249
SEPTEMBER	AIC	Poisson	968048	970922	968344	960360	970857	967706
	AIC	Negative Binomial	915686	918740	914398	912642	916783	915650
	R-Squared	Poisson	0.258	0.251	0.258	0.277	0.250	0.259
	R-Squared	Negative Binomial	0.258	0.251	0.258	0.277	0.250	0.259
OCTOBER	AIC	Poisson	945915	946780	946904	936887	948075	944912
	AIC	Negative Binomial	858884	861222	857611	857141	859580	858888
	R-Squared	Poisson	0.242	0.240	0.241	0.264	0.235	0.244
	R-Squared	Negative Binomial	0.242	0.240	0.241	0.264	0.235	0.244
NOVEMBER	AIC	Poisson	1124838	1127934	1130343	1114969	1131145	1125846
	AIC	Negative Binomial	996595	999583	998020	993916	1000738	997770
	R-Squared	Poisson	0.240	0.233	0.230	0.259	0.227	0.238
	R-Squared	Negative Binomial	0.240	0.233	0.230	0.259	0.227	0.238
DECEMBER	AIC	Poisson	1440497	1441150	1443758	1435790	1447835	1441806
	AIC	Negative Binomial	1293060	1297496	1294813	1292443	1298003	1295163
	R-Squared	Poisson	0.257	0.254	0.252	0.264	0.244	0.254
	R-Squared	Negative Binomial	0.257	0.254	0.252	0.264	0.244	0.254

Table 5.12: Outcome response variable timing Poisson and NB Regression per fold.

5.2.3 Forecast the aggregate of return requests per day

Another way to validate the predictive power of the models is to use the aggregate of return requests per day. Instead of a predicting whether and when each request will be returned to the warehouse, we predict for all return requests per day when those requests will be returned to the warehouse. Using the aggregate of return requests per day, we test whether the predictive power of the response variable timing could be less due to the long planning window of 26 days. If we predict per day instead of per return request, we can see that although the number of days between the registration of the return and the processing of the return reaches 26 days, Figure 5.21 shows that a possible resource planning based on aggregate measures does not necessitate a planning window of 26 days, but 11 days would be sufficient. The window is diminished because we use the average per day instead of each request individually. Besides the validation, we test also the effect of a reduced planning window on the predictive power. In this way, Bol.com receives an approximation of the impact of including the data of the transporters, which would reduce the planning window even further to 5 days.

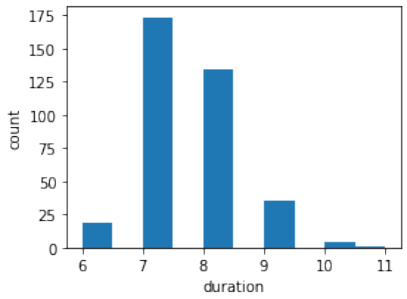


Figure 5.21: Histogram of timing distribution aggregate returns.

We test the LASSO Regression model against the Poisson Regression model. The code for retrieving the data from BigQuery is provided in Appendix M.

The input variables for the aggregated forecast are the following:

- Weeknumber;
- Day of the week;
- Number of registrations;
- Total quantity;
- Registrations via webshop;
- Registrations via customer service;
- Registrations via Docdata;
- Average registration hour;
- Minimum registration hour;
- Maximum registration hour;
- Standard deviation registration hour;
- Average quantity;
- Minimum quantity;
- Maximum quantity;
- Standard deviation quantity;
- Average price;
- Minimum price;
- Maximum price;

- Standard deviation of the price.

Results of the aggregate return requests per day for the timing using LASSO
First, we used a LASSO Regression to forecast the timing of the returns per day using the same method as mentioned in Chapter 4. The difference lies within the aggregate requests per day instead of per request.

The results of the LASSO Regression of the train and test data are quite stable. The residual plot for the training versus the testing data can be found in Figure 5.22 for July. The residuals show some type of pattern, which suggest a nonlinear model would be a better fit. Figure 5.23 visualizes the predicted duration versus the actual duration of the aggregated items for July. Based on the figure, we can see a better fit compared to the individual requests.

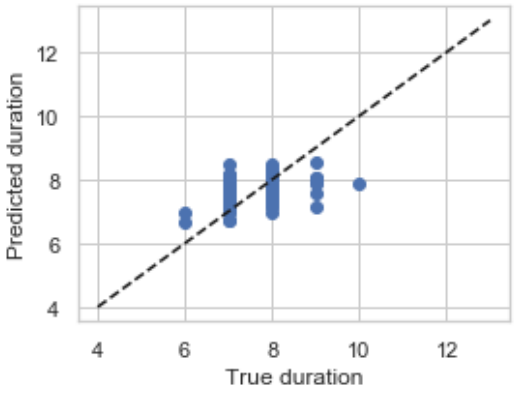


Figure 5.22: Residual plot aggregate requests LASSO. *Figure 5.23: Aggregate actual versus predicted duration LASSO.*

Table 5.13 shows the performance of the LASSO Regression. The results are not stable between the training and test set for the performance measurement R-Squared. The value of the R-Squared is increased compared to the disaggregated forecast but remains low. Therefore, LASSO Regression is in our case not a good forecasting method to determine the number of returns.

	TRAINING		**TEST**	
	R-Squared	*RMSE*	*R-Squared*	*RMSE*
AGGREGATED ITEMS	0.36	0.65	0.14	0.70

Table 5.13: Aggregate results LASSO response variable timing.

Results of the aggregate return requests per day for the timing using Poisson Regression
We also test the predictive power of the Poisson Regression model using the aggregate return requests per day instead of per request. The model remains the same, however the input variables change from request level to day level, with the associated input variables mentioned above. Figure 5.24 shows the actual and predicted duration per day for 2019. The x-axis indicates the day of the 2019 and the y-axis indicates the duration. So, 70 on the x-axis indicates the 70th day in 2019. Deviation between the predicted and actual duration is visible and indicates not a perfect fit. Although, the fit is better compared to the figures in Section 5.1.2.

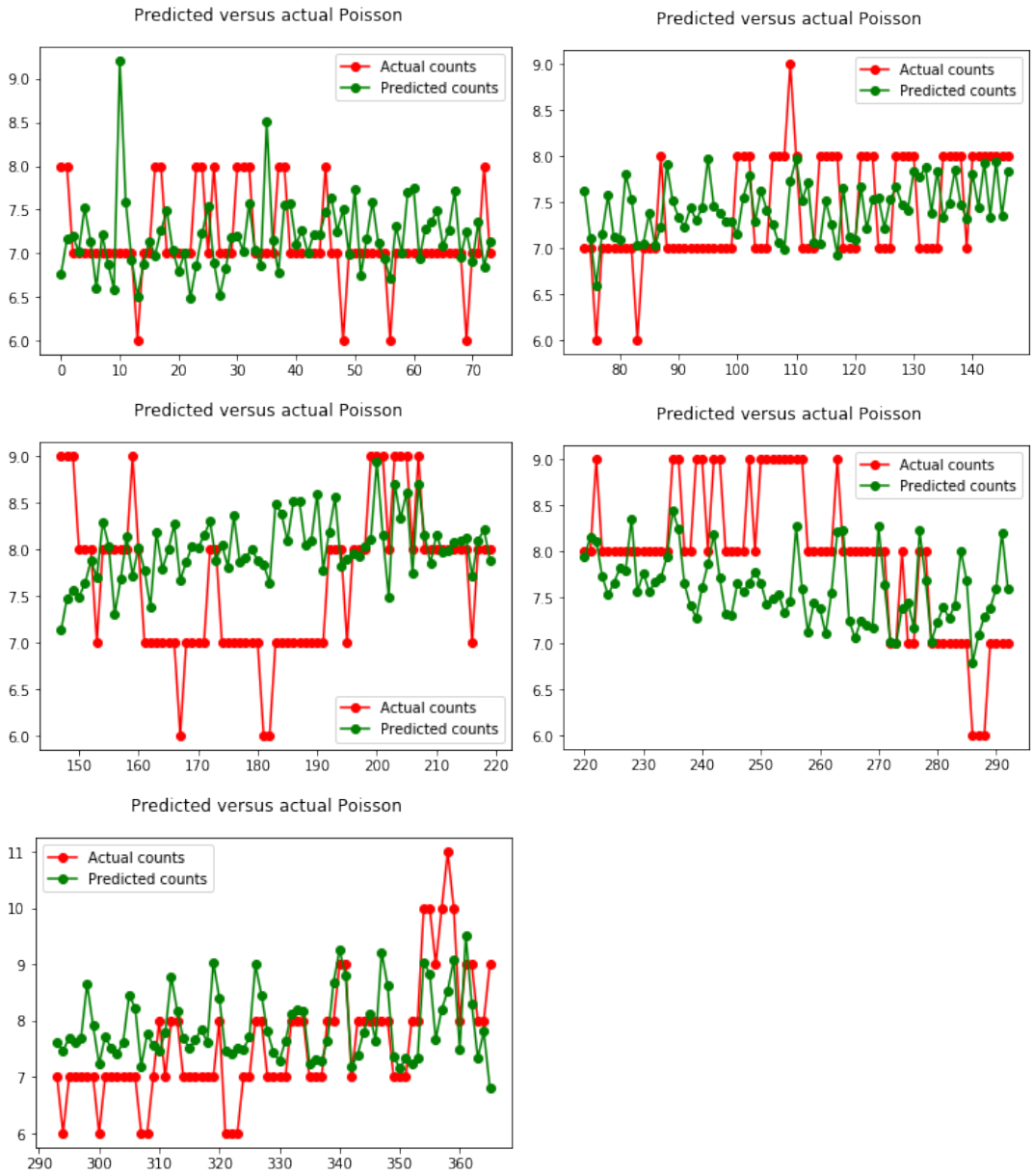


Figure 5.24: Outcomes predicted versus actual duration aggregated Poisson Regression.

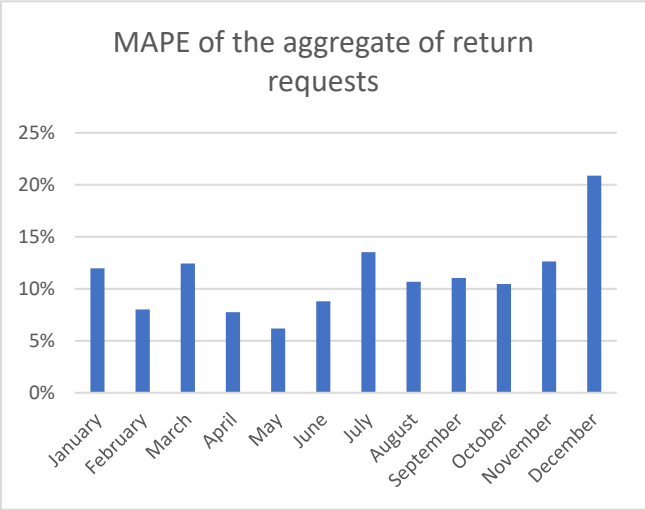
Table 5.14 shows the performance of the Poisson Regression. The performance is better compared to the Poisson Regression on request level. Based on the results, Poisson Regression is better at describing the timing of a return, for both on request level and on aggregated level compared to LASSO. Although, the R-Squared values deviate per fold. The results show that a decrease in the planning window will lead to a major increase in performance and accuracy. The R-Squared value is still not close to 1, but this is common for predicting human behavior. If the transport data would be integrated, the planning window would be only 5 days and the human behavior would be out of scope. We assume the R-squared value to increase even further and be closer to 1.

		FOLD 1	*FOLD 2*	*FOLD 3*	*FOLD 4*	*FOLD 5*
AGGREGATED RESULTS	AIC	1184.86	1186.22	1181.40	1179.85	1181.16
	R-Squared	0.433	0.431	0.515	0.486	0.563

Table 5.14: Aggregate results Poisson Regression response variable timing

Influence using the aggregate of return requests on total number of returns

We have shown an improvement in the performance of the Poisson Regression model if the model is based on the aggregate return requests per day instead of per return request. However, a better model fit does not necessarily imply a better prediction of the total number of returns as shown in Section 5.1.3, where Poisson outperformed Negative Binomial on the overall MAPE. If we use the aggregate of the return request with the Poisson Regression model as stated above, the impact on the overall MAPE is shown in Figure 5.25. The associated average MAPE is 11.20%. Based on the figure, December does not perform well. This is mainly caused by the holidays on which the warehouse is not fully operating. Because our analysis is based on the processed returns instead of actual arrival at the warehouse, this has a major influence on the accuracy of the forecast. Therefore, we would advise to adjust the month December with the workforce to increase the accuracy of the forecast. In Figure 5.19 we compare the overall performance of the models. From the figure, we can conclude that for the current dataset the aggregate of the return requests results in a better prediction compared to the prediction on request level. The current variables are not able to describe the response variable timing with a high accuracy. The predictive variables are based on product characteristics, reason codes and timing aspects. However, they are expected to predict customer behavior instead. Therefore, it is hard to predict on request level and Bol.com should aggregate the requests per day for the prediction of the total number of returns per day to reduce fluctuations, allowing for better data analysis and prediction.



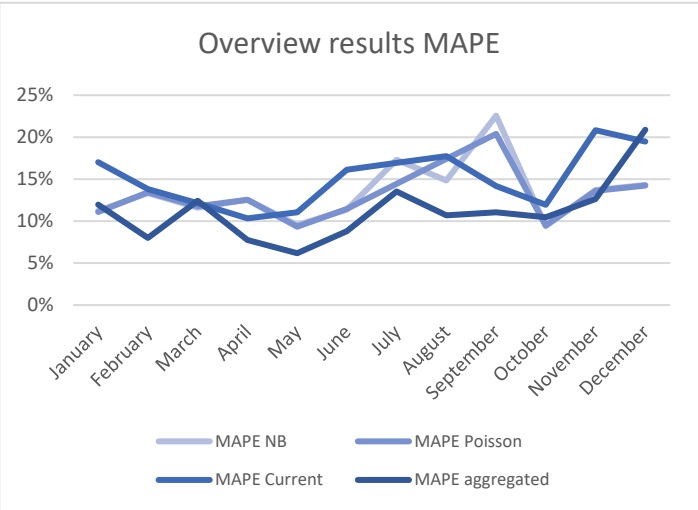


Figure 5.25: Forecast based on the aggregate of return requests. *Figure 5.26: Results comparison all models.*

6. Implementation

In this chapter we provide an answer to the fifth research question: *'How should the model be implemented at Bol.com?'*.

6.1 Implementation of the new return forecast

To implement the new return forecast, some steps need to be taken every day. Bol.com has two different methods to predict the number of returns, namely:

- On return request level: based on a prediction for each request individually, a prediction of the total number of returns is made using Logistic Regression and Poisson Regression.
- On the aggregate of the return request on daily level: based on the aggregate of the return requests per day, a prediction of the total number of returns is made using Poisson Regression.

On return request level:

For the response variable quantity, the following steps should be performed:

1. Export the return requests from Boomerang. Using the BigQuery code provided in Appendix C, each registered request can be exported to a csv-file.
2. Run the Logistic Regression model of the response variable quantity in Python. The code can be found in Appendix E. Only the part of the Logistic Regression needs to be executed, to make predictions.
3. Use the csv-file 'format' created by running the model. Using this format, each categorical variable is converted to a binomial variable. With this format, predictions could be made.
4. Within this 'format' use the concerned coefficients of that month. The coefficients per month are given in Appendix L, which should be used for the prediction.
5. The model will classify for each item whether the return request is true.
6. Use the truly classified return requests as an input for the Poisson Regression model.
7. Insert the coefficients from the Poisson Regression model into the file.
8. The model will predict for the requests that will be returned the timing between the request and return to the warehouse.

To calculate the total number of returns per day, the following modifications should be made:

1. Modify the predictions for the weekend, by randomly changing the duration with 2-6 days for a Saturday and 1-5 days for a Sunday.
2. Modify the data by extracting the number of zero values by looking at the zero value percentage per registration hour, weekday and month.
3. Modify the data by increasing the number of returns with direct returns, by looking at the direct return percentage per weekday and month.
4. The number of items per day should be summed to receive the final total number of returns per day.

Aggregated return request on daily level:

1. Export the aggregated return requests from Boomerang. Using the BigQuery code provided in Appendix M; each registered request can be exported to a csv-file.

2. Run the Python code to create the csv-file ‘format_agg’. Using this format, predictions could be made.
3. Within this ‘format_agg’ use the concerned coefficients of that year stated in the document.
4. The model will determine for the aggregated results per day the predicted arrival date.
5. Sum all predictions of the weekend and divide this over the days using the fixed multiplier index of the previous year, which was explained in Section 2.2.5.

A more detailed implementation plan is provided in Appendix O. Besides the explanation of implementing the model, a guideline for updating the parameters is also included.

6.2 Requirements of the implementation

The model could be implemented using the steps mentioned in the previous section. Bol.com has access to all the data and the forecast will be explained in detail with the associated colleagues. The coefficients of the models should be updated frequently with the new available data to maintain or increase the accuracy.

On return request level

The data from Boomerang should be exported each morning of the previous day to carry out the steps mentioned in the previous section. To run the model, it is necessary that all steps are executed correctly, because it is sensitive for errors. Currently, holidays or other special days are not incorporated in the model. The model has no constraint regarding non-working days. For each week, Bol.com can modify the forecast for those specific days based on their knowledge. Furthermore, the model has no constraint for the weekends, but divides the number of returns of the weekend randomly over the weekdays. Bol.com can change this method by looking at the current analysis of Chapter 2. For example, the multiplier index can be used to divide the number of returns over de week.

Aggregated return request on daily level

The data from Boomerang should also be exported each morning of the previous day to carry out the steps mentioned in the previous section. This method is more straightforward, but provides better results. Therefore, we advise Bol.com to use the aggregate return request model on daily level.

7. Conclusion and recommendations

7.1 Conclusion

This section focuses on answering the main research question: '*Develop a model that forecasts the number of return items for the short-term to improve the accuracy at the warehouse, which leads to a better efficiency and satisfied personnel and clients'.* We created different models to predict the response variables, namely the quantity of request returns and the timing of these return. Using these two complementary response variables, a forecast of the total number of returns could be made.

7.1.1 Response variable quantity returns

The models that were used for the classification of the return requests were the following:

1. Logistic Regression all features;
2. Logistic Regression 10 important features;
3. Random Forest tree with all features;
4. Random Forest tree with 10 important features.

Based on the results, both Random Forest as well as Logistic Regression perform well in predicting whether a return request will be returned. The Random Forest tree with all features performs best, with a F1-score of 0.826. The F1-score indicates the harmonic mean of precision and recall, where a value of 1 indicates perfect precision and recall. However, the differences between Random Forest and Logistic Regression are small and higher interpretability is desirable for Bol.com. Therefore, the Logistic Regression was preferred over the Random Forest tree. The Logistic Regression with all features performed slightly better on average with a F1-score of 0.784 compared to 0.777 of the Logistic Regression with only ten features. The drawback of all four methods with the current data was the difficulty of predicting the true negatives. Almost all return requests were classified as returned, because on average 81% of the requests was returned. Using the Logistic Regression with all features led to a misclassification of 15.6%. Using the Logistic Regression for classification results in a higher prediction of the number of return requests compared to the actual number of returns.

The most important features to classify whether a return request will be returned are: *hour of registration, day of the week, the price, all three sources of registration (BLUE, DOCDATA and webshop), quantity, both selling parties (own and LvB), reason codes and some of the product groups*. The important features are visualized in Table 7.1 and compared with the findings of the literature review of Chapter 3. A positive sign (+) indicates a positive effect on whether the request will be returned and a negative sign (-) a negative effect. The features *hours of registration* and *day of the week* are an extension to the research of Cui et al. (2020), which only showed that the year and month affect the number of returns. All reason codes positively affect the return of a request, except if the *article is delivered too late* or if *no reason is provided*.

The results of our research show similarities to other findings in the literature and makes combinations of the results. Our research showed, in addition to the existing literature, that the *hour of registration* and *day of the week* influence the number of returns. Furthermore, we have shown that the input variable return request was a solid starting point of the prediction instead of sales. The features that have proven to impact the number of returns in the literature, but were not considered in our research, are customer characteristics, delivery time, account age, historical return and sales. We would advise to implement those features as well to see the impact on the number of returns.

FEATURE	CORRESPONDING TO LITERATURE
HOUR OF REGISTRATION (-);	Extension of time component (year/month) Cui, Rajagopalan & Ward (2020)
DAY OF THE WEEK (-);	Extension of time component (year/month) Cui, Rajagopalan & Ward (2020)
PRICE (+);	Hess & Mayhew (1997); Asdecker & Karl (2018)
ALL SOURCES OF REGISTRATION (+)	Extension of retailer component of Cui, Rajagopalan & Ward (2020)
QUANTITY (-);	Asdecker & Karl (2018); Cui, Rajagopalan & Ward (2020)
BOTH SELLING PARTIES (+)	Cui, Rajagopalan & Ward (2020)
REASON CODES (MOST +)	Potdar (2009); Potdar & Rogers (2012)
SOME PRODUCTGROUPS	Similar to production/resources component of Cui, Rajagopalan & Ward (2020)

Table 7.1: Result explanatory features response variable quantity compared to the literature review.

7.1.2 Response variable timing

In this thesis, we used the following three different models to predict the timing of the returns:

1. LASSO Regression;
2. Poisson Regression;
3. Negative Binomial Regression.

Based on the results described in Section 5.1.2, LASSO Regression indicates a poor fit with our dataset, which is mainly due to the assumption of a normal distribution. The results of the Poisson Regression and Negative Binomial Regression were more promising, but still not desirable. We expected the difference between the Negative Binomial Regression and Poisson Regression to be larger, due to the overdispersion. Though, the R-Squared for both regressions were around 0.25. The AIC value of Negative Binomial Regression was lower compared to Poisson Regression, which indicates a better fit. Both Poisson and Negative Binomial Regression are tested for the overall performance. Due to less parameter estimation and updating of the Poisson Regression, this method could also be preferred by Bol.com.

The performance of the R-Squared value is quite low. Which can be explained by the fact that no additional data is available regarding the return process except for the product characteristics and time components. We ask the model to predict the timing between the registration and return to the warehouse, which involves prediction of human behavior. Although, our model is only based on time and product characteristics, such as *day of the week* and *price*. The only interaction with the customer is the *return reason*. There is no extra information available about the customer behavior in the past or if the customer hands in the parcel at the PUP. If this information would be included in the model, the accuracy would increase. The prediction of human behavior is almost impossible if no data about this behavior is integrated. Therefore, a lower R-Squared value is explainable and makes sense. Because the prediction interval was small, we increased this interval using a natural logarithm transformation. Implementing the transformation leads to a broader interval, however the results only improved from 0.25 to 0.27.

Features that have shown significant impact on the timing of a returns with a p-value of 0.05 are shown in Table 7.2. Behind each feature, it is stated whether researches from the literature review showed the predictive power of those features with significance. However, most findings in the literature were only about the prediction of the number of returns. Only Potdar (2009) and Potdar et al. (2012) showed that *reason codes* influence the timing of the returns. However, if we compare the results with the determination of the number of returns, the major difference lies within the price. *Price* has not proven to have significant impact on the timing process, which is also proven by Hess et al. (1997).

Furthermore, another difference is that the *source of registration of DOCDATA* has not proven to significantly affect the timing of the return.

FEATURES	COMPARISON LITERATURE
QUANTITY (-);	-
REGISTRATION HOUR (+);	-
DAY OF THE WEEK (-);	-
THE SELLING PARTIES (+);	-
SOURCES OF REGISTRATION (ONLY BLUE AND WEBSHOP) (+);	-
THE REASON CODES (+);	Potdar (2009), Potdar & Rogers (2012)
SOME OF THE PRODUCT GROUPS (+/-).	-

Table 7.2: Results explanatory features response variable timing compared to the literature review.

7.1.3 Aggregate of return requests prediction

To test the predictive power of the models, we decreased the planning window from 26 to 11 days. The planning window decreased by using the aggregate of return requests per day instead of individual return requests. The performance of the regression models increases when the model shifts from request based to daily based, where return requests are aggregated per day. Although, the LASSO Regression remains still a poor fit. The Poisson Regression provided better results. The R-Squared value of the Poisson Regression increases from 0.25 to 0.56 for predicting the average timing of a return after registration. The decrease of the forecasting window explains the increase in performance and shows the potential of the model if the planning window is diminished. Furthermore, the increase of the forecast accuracy indicates that predicting on item level makes less sense compared to predicting on a daily level. Which is mainly due to the fact that we expect the model to predict customer behavior in the return process to determine the timing, without information about those customers. Aggregating the requests per day reduces fluctuations and thereby noise, which increases the accuracy of the model. Based on the current available data, forecasting on aggregate requests per day would be a better choice.

7.1.4 Overall performance

The total number of returns is predicted using classification of return requests by the Logistic Regression and the timing by Poisson Regression and Negative Binomial Regression. To cope with direct returns, which implies returns without registration, the data is modified using historical percentages based on the day of the week and the month. The MAPE of our proposed model using Poisson Regression (13.3%) shows significant improvement compared to the current forecast MAPE (15.1%). The MAPE of our proposed model using Negative Binomial shows also a significant improvement of the MAPE (13.5%), but is lower compared to Poisson. The MAPE is not reduced as much as we preferred, which is mainly due to the overestimation caused by the Logistic Regression. An increase in the predictive power of the true negatives will lead to a decrease of the MAPE.

If the overall performance is predicted using the aggregated return requests per day, the average MAPE is reduced to 11.2%. Because no additional data regarding the customer behavior or transport is included in the model, other than the *return reasons*, it makes sense that an aggregated forecast results in a better performance.

Based on the results, we would advise Bol.com to implement the forecasting method using Logistic Regression and Poisson Regression to predict the total number of returns on request level. The MAPE can be reduced even further using aggregated return requests per day. This decrease in the MAPE shows

the potential of reducing the forecast window and in addition the advantage of aggregation with the current available data. If Bol.com integrates additional data regarding the transport process, the forecasting model could provide promising results on request level. However, if no additional data is integrated in the model, the aggregated model provides a better performance. Therefore, using the current data, forecasting on daily level using aggregation is preferred. Based on the outcome, we accomplished our main goal to come up with a short-term return forecasting model that increases the accuracy.

7.2 Discussion

In this section we elaborate on the assumptions and interpretations of the models and discuss why the models might not be completely verified and validated.

- A major drawback of the model is the lack of data regarding the actual duration of a return. The model is based on the processing data instead of the arrival time at the warehouse. Due to the current inaccuracy of the forecast, Bol.com ensures that the WIP is high enough to cope with underestimation. Therefore, processed requests could be arrived at the warehouse one or more days upfront.
- Due to an error in the smart returns system of IM, not all returns are recognized. Therefore, around 9% of the returns is not matched with the associated customer. Hence, the timing between the request and processing is zero. Those zero values negatively influence the accuracy of the Boomerang data and therefore the accuracy of the models. Besides those zero values due to an error in the system, there are also zero values due to direct returns without registration send by the customer. This reduces the reliability of the Boomerang data as an input of the model. The zero values are incorporated in the model using historical patterns based on the month, day of the week and registration hour. The used approach is not investigated and verified extensively and only provides an impression of how to cope with those zero values and direct returns.
- The determination of the preferred models should be revisit, if additional data is incorporated, because the performance of the Random Forest tree could have a larger advantage compared to the Logistic Regression model. Furthermore, we believe other models could provide a better fit to our data. For example, non-linear models would describe the dataset in a better way.
- Holidays and weekends are not incorporated in the model. The forecast does not integrate non-working days in the timing of a return. The prediction for the weekends is randomly divided over the week. Therefore, the accuracy of the model is decreased. A more sophisticated method for excluding the weekends and holidays would increase the accuracy of the model.
- The literature review can be more extensive. Adding extra literature would improve the quality of the comparison of our results with the literature.
- The overall performance of the Random Forest tree is not integrated in this research, due to the lower interpretability of this method. However, it could be the case that this method results in a lower MAPE compared to the Logistic Regression.
- The missing values are currently filled with the most often chosen index (MCI). Other methods handling missing values should be considered and the impact on the overall performance should be measured.
- Because the Logistic Regression is poor at predicting the true negatives, almost all requests are classified as being returned. That is why the total number of returns is overestimated. Using the

proposed model results in an overestimation of the total number of returns. We expect that other models or techniques will perform better at predicting those true negatives.

- The forecast is for 2019 specific; however, we assume similarities between the years. Although, for example COVID-19 can have an impact on the return process. That is why the forecasting accuracy should be checked regularly, to adjust the coefficients based on irregular events. This also emphasizes the importance of updating the parameters frequently.
- The total number of returns is forecasted on request level as preferred by Bol.com. Therefore, this method could be integrated in the new Bol.com Return Centre, in which the return process will mainly be automated. The assignment of the automatic production lines per product group can be based on the predicted number of returns per product group. However, based on the current data, forecasting on request level is not preferred. The current variables do not describe customer behavior in the return process, except for the reason codes. Therefore, aggregating the requests per day would be a better choice because fluctuations are reduced in the return process.

7.3 Practical recommendations

The purpose of this section is to give practical recommendations to Bol.com to increase the accuracy of the proposed forecast.

- Our proposed forecasting model increases the average accuracy of the number of returns on daily basis. However, the results could be more promising and accurate if additional data would have been incorporated. As shown in Figure 2.16, product scans are performed during the return process. Currently, this data is not stored correctly and cannot be used. We would strongly advise to investigate the possibilities to extract this information. Including this information would decrease the forecasting window from 26 days to only 5 days. This would have a major impact on the accuracy of the model as shown by the aggregate model per day.
- Based on the results, we would advise to implement the Poisson Regression model based on the aggregate of return requests, if no additional data is integrated in the disaggregated model.
- Furthermore, we would advise to incorporate additional data regarding the past return behavior of the customer if the forecast window is not decreased to 5 days. The prediction of customer behavior is expected by the model, but no data regarding this behavior is included in the model.
- We would strongly advise to implement a scan at the warehouse in Waalwijk to be able to calculate the actual number of returns that are returned to the warehouse per day instead of an estimation.
- Currently, the database Boomerang is a storage place of registrations. However, cancelled registrations will remain in the database. We would advise to remove the cancelled registrations to increase the reliability of the Boomerang data as an input for the forecasting model.
- We advise to solve the problem of the smart return system of Ingram Micro to avoid a mismatch between a registration and the processing code. This would reduce the number of zero values and increase the accuracy. Furthermore, this would decrease the number of customer complaints and requests regarding their unhandled return.
- Besides, we would advise to update the parameters regularly. Every month, the parameters for that particular month can be updated and compared to the previous year. If major changes are visible, the coming month's parameters should also be updated.

- Lastly, we would advise to make it impossible to send a return back to the warehouse without registration. This ensures no direct returns to the warehouse, which increases the accuracy of the model.

7.4 Further research recommendations

There are assumptions in our model that need further research, since they were out of scope, or simplified in this research.

- External variables that are likely to influence the timing of a return should be integrated to increase the predictability of the timing of a return. For example, the weather or holidays could affect the timing of a return from a customer.
- The variables that were not included in our model, but in the literature have proven to impact the number of returns should be tested and integrated in our model. Those variables are customer characteristics, delivery time, account age, historical returns and sales.
- The data cleaning process removed all durations outside the window of 1-26 days. Less than 1% of the data was deleted for durations above 26 days. It is useful to perform more research on those duration above 26 days. For example, to investigate whether those outliers arise through holidays or weekends.
- More effort can be put into the prediction of direct returns. Currently, the prediction of direct returns was not of our interest, but has a major impact on the total number of returns. Therefore, additional research is needed to increase the accuracy of the models.
- Right now, the weekends are included in the forecast. By dividing the total predicted number of returns randomly over the week, the returns are spread out. However, more effort should be put into this division. For example, the fixed multiplier index would be a better approximation instead of the random distribution.
- The testing set contains 20% of the data compared to 80% of training data. We did not investigate the impact of this division. Therefore, we advise for further research to investigate the impact of the size of the testing and training set.
- The transformation using a natural logarithm provided slightly better results. However, this transformation is not used in the overall prediction of the total number of returns, because the differences were only small. Nevertheless, we would advise to investigate other transformations to see the impact on the overall performance of the total number of returns.
- With the current data, predicting the duration on item level is hard. To validate the forecasting models, we investigated the results of the aggregated returned items in addition. Those predictions showed promising results and should be unraveled in further research. For example, the impact of aggregation based on registration hour or product groups should be investigated. Based on the current dataset, a prediction based on aggregated requests provides a higher accuracy.

Glossary

AIC Akaike Information Criterion. 49

AUC Area Under the ROC Curve. 45

Boomerang database for return registrations. 20

IM Ingram Micro. 8

LASSO least Absolute Shrinkage and Selection Operator. 28

LvB Logistics via Bol.com. 7

MAD Mean Absolute Deviation. 15

MAPE Mean Absolute Percentage Error. 15

MCI Most Chosen Index. 44

MSE Mean Squared Error. 33, 49

NB Negative Binomial Regression. 62

OLS Ordinary Least Square. 29

PUP Pick-Up-Point. 8

RFECV Recursive Feature Elimination Cross-Validated. 44

RMSE Root Mean Square Error. 49

R-Squared coefficient of determination. 49

S&OP Sales & Operations Planning. 14

WIP Work in Progress. 10, 21

References

Alonso, A., Torres, A., & Dorronsoro, J. R. (2015). Random Forests and Gradient Boosting for Wind Energy Prediction. *Hybrid Artificial Intelligent Systems*, 26-37.

Asdecker, B., & Karl, D. (2018). Big data analytics in returns management - Are complex techniques necessary to forecast consumer returns properly. *2nd International Conference on Advanced Research Methods and Analytics (CARMA2018)* (pp. 39-46). Valéncia: Universitat Politècnia de València.

Babai, M., Ali, M., Boylan, J., & Syntetos, A. (2013). Forecasting and inventory performance in a two-stage supply chain with ARIMA(0,1,1) demand: Theory and empirical analysis. *International Journal of Production Economics*, 463-471.

Bol.com. (2020, January 16). *Algemene presentatie ENG december.pptx.* Retrieved from Confluence: https://confluence.tools.bol.com/pages/viewpageattachments.action?pageId=13240783&metadataLink=true&preview=/13240783/108254963/algemene%20presentatie%20%20ENG%20december.pptx

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: the X-random case. *International Statistical Review*, 60: 291-319.

Bühlmann, P., & van de Geer, s. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications.* New York: Springer-Verlag.

Clottey, T., Benton, W., Jr, & Srivastava, R. (2012). Forecasting product returns for remanufacturing operations. *Decision Sciences*, 589-614.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction. *International Journal of forecasting*, 635-660.

Cui, H., Rajagopalan, S., & Ward, A. (2020). Predicting product return volume using machine learning methods. *European Journal of Operational Research*, 612-627.

de Brito, M. P. (2004). *Managing reverse logistics or reversing logistics management?* Breda University of Applied Sciences: ResearchGate.

Efron, B., & Tibshirani, R. (1994). *An introduction to the bootstrap.* CRC press.

Flapper, S. (1995). *One-way or reusable distribution items?* Eindhoven: TU Eindhoven. Fac.TBDK,Vakgroep LBS: working paper series; Vol. 9504.

Galli, S. (2020). *Python Feature Engineering Cookbook: Over 70 recipes for creating, engineering, and transforming features to build machine learning models.* Birmingham: Packt Publishing Ltd.

Gardner, W., Mulvey, E., & Shaw, E. (1995). Regression Analyses of Counts and Rates: Poisson, Overdispersed Poisson, and Negative Binomial Models. *Psychological Bulletin*, 392-404.

Goh, T., & Varaprasad, N. (1986). A statistical methodology for the analysis of the life-cycle of reusable containers. *IEE Transactions*, p. 18:4247.

Guide, J. V., & Srivastava, M. (1997). An Evaluation of capacity planning techniques in remanufacturing environment. *International Journal of Production Research*, Volume 35, Issue 1, pp 67-82.

Hamilton, J. D. (1994). *Time series analysis.* Princeton: Princeton University Press.

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

Heerkens, J., & Van Winden, A. (2012). *Geen probleem, een aanpak voor alle bedrijfskundige vragen en mysteries.* Business School Nederland.

Hess, J., & Mayhew, G. (1997). Modeling merchandise returns in direct marketing. *Journal of Direct Marketing*, 20-35.

Horsten, R. (2020, January 22). Wet Arbeidsmarkt in Balans. Waalwijk, Noord-Brabant, The Netherlands.

Huang, J., Cao, L., & Srivastava, J. (2011). *Advances in Knowledge Discovery and Data Mining.* Shenzhen, China: Springer Science & Business Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning - with applications in R.* New York: Springer.

Kelle, P., & Silver, E. (1989). Forecasting the returns of reusable containers. *Journal of Operations Management*, pp. 8(1):17-35.

Khosla, M., Jamison, K., Ngo, G. H., Kuceyeski, A., & Sabuncu, M. R. (2019). Machine learning in resting-state fMRI analysis. *Magnetic Resonance Imaging*.

Kim, S., & Kim, H. (2016, July-September). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, pp. 669-679.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence (IJCAI)*, 1137-1143.

Kolassa, S., & Schütz, W. (2007). Advantages of the MAD/Mean Ratio over the MAPE. *Foresight: The International Journal of Applied Forecasting*, pp. 40-43.

Liang, X., Jin, X., & Ni, J. (2014). Forecasting product returns for remanufacturing systems. *Journal of Remanufacturing*, 4:8.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3).

Mentzer, J., & Cox Jr, J. (1984). Familiarity, application and performance of sales forecasting techniques. *Journal of Forecasting 3(1)*, 27-36.

Micro, I. (2020, 02 07). *About us*. Retrieved from Ingram Micro: https://nl.ingrammicro.eu/over-ons/on-the-company

Mollenkopf, D. A., Rabinovich, E., Laseter, T. M., & Boyer, K. K. (2007, May). Managing Internet Product Returns: A Focus on Effective Service Operations. *Decision Sciences*, pp. 215-250.

Moreno, J., Pol, A., Abad, A., & Blasco, B. (2013). Using the R-MAPE index as a resistant measure of forecast accuracy. *PSICOTHEMA, Volume 25, Issue 4*, 500-506.

Petersen, J. A., & Kumar, V. (2009). Are Product Returns a Necessary Evil? Antecedents and Consequences. *Journal of Marketing*, 73(3), 35-51.

Potdar, A. (2009). *Methodology to forecast product returns for the consumer electronics industry.* Arlington: The University of Texas.

Potdar, A., & Rogers, J. (2012). Reason-code based model to forecast product returns. *Foresight*, 14(2), 105-120.

Savin, I., & Winker, P. (2013). Lasso-type and Heuristic Strategies in Model Selection and Forecasting. In C. Borgelt, M. Gil, J. Sousa, & M. Verleysen, *Towards Advanced Data Analysis by Combining Soft Computing and Statistics* (pp. 165-176). Berlin: Springer.

Silver, E. A., Pyke, D. F., & Thomas, D. J. (2016). *Inventory and Production Management in Supply Chains.* Boca Raton: CRC Press.

Synthethos, A., & Boylan, J. (2001). On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71, 457-466.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society*, 73, Part 3, pp. 273-282.

Toktay, B. L. (2001). *Forecasting Product Returns.* Faculty & research: European Institute of Business Administration: INSEAD.

Toktay, B., van der Laan, E., & de Brito, M. (2003). Managing Product Returns: The Role of Forecasting.

Toktay, B., Wein, L., & Zenios, S. (2000). Inventory management of remanufacturable products. *Management Science*, 46(11), 1412-1426.

Tutz, G., & Ulbricht, J. (2009). Penalized regression with correlation-based penalty. *Stat.Comp.19*, 239-253.

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*, 20: 375-387.

Zhang, Y., Minchin, R. M., & Agdas, D. P. (2017). Forecasting Completed Cost of Highway Construction Projects Using LASSO Regularized Regression. *Journal of Construction Engineering and Management*, v143, n10.

Appendix

Due to confidentiality, the appendices are excluded from this public version.