MASTER'S THESIS

New Product Forecasting with Analogous Products

Applying Random Forest and Quantile Regression Forest to forecasting and inventory management

Author
R.M. van Steenbergen

October 2019

Supervisors University of Twente
dr.ir. M.R.K. Mes
dr. C.G.M. Groothuis - Oudshoorn

Supervisors Slimstock
J.M. Veldhuizen, MSc
B. van Gessel, MSc

UNIVERSITY OF TWENTE.

Drienerlolaan 5
7522 NB Enschede
The Netherlands

slimstock

Zutphenseweg 29
7418 AH Deventer
The Netherlands

Management summary

New product forecasting is challenging compared to forecasting demand of existing products since historical sales data is not available as an indicator of future sales. Additionally, there is limited analysis time and there is a general uncertainty among new products. Despite its complexity, these forecasts are crucial for a company since they guide critical operational decisions. Poor forecasts can result in stock-outs or overstock situations. Both have a direct impact on the company's profitability and may also decrease customer satisfaction and market share.

Due to the importance of new product forecasting, Slimstock wants to support stock-keeping companies with their product introductions. Slimstock wonders how they can utilise product characteristics for new product forecasting. The number and type of these characteristics differ per company. Since the goal is to design a method that can be employed at multiple companies, it should be adaptable to specific situations. The inherent uncertainty among new products should be conveyed by prediction intervals, such that risks in decision-making can be anticipated. This leads to the following research objective:

> *"Develop and validate an analytical method that provides pre-launch forecasts for the first four months of demand of new products. The method should utilise product characteristics of new and existing products to generate forecasts with prediction intervals, based on the historical demand of existing comparable products."*

Method

After analysing new product data and investigating relevant literature, we developed demandForest: a new product forecasting method based on Random Forest algorithms. DemandForest divides the demand of an introduction period into a demand profile and the total amount of demand. Demand patterns of existing products are clustered in distinctive profiles with K-Means. Afterwards, a Random Forest is trained to predict the profile for a new product based on the product characteristics. Besides, a Quantile Regression Forest is trained to predict the total demand and its conditional distribution. The conditional distribution estimated the uncertainty of the demand of a new product and can be used to generate prediction intervals and can be used to set certain target service levels. Two extensions are proposed that fit a Log-Normal and Gamma distribution to the conditional distribution. The aim of the extensions is to improve the conditional distributions, since they are often based on a limited set of comparable existing products. Combining the profile with the demand results in a forecast, whereas the profile and the conditional distribution can be used as order level for inventory management.

To asses demandForest on both quality and robustness, we evaluate the performance using six data sets, one synthetic and five from stock-keeping companies, which are clients of Slimstock. Additionally, we suggest two benchmark methods, based on the average and percentiles of existing products (called ZeroR) and the most similar product (called Proximity). By using the most similar product, the Proximity method is defined such that it imitates the current forecasts and decisions of supply chain planners.

Results

The forecast quality of demandForest is evaluated on the individual predictions of the profile and the total demand, and on the combined forecast. The performance for predicting the profiles is compared to the average profile of existing products. The predictions resulted in a better performance than the average profile when the kappa score was above 0.40. For predicting the total amount of demand, the demandForest methods, its extensions, and the benchmark methods are compared. We evaluated the Root Mean Square Error (RMSE) for the predictive performance, and the Prediction Interval Coverage Probability (PICP) and Prediction Interval Normalised Average Width (PINAW) for the prediction intervals. The demandForest methods provided better results for both the forecast and the intervals. Only for company C, the forecast of the Proximity method was more accurate. The Log-Normal and Gamma extensions slightly improved the regular demandForest performance. The Proximity method showed the most unreliable prediction intervals, which were often too wide or too narrow. Also for the combined forecast, the demandForest methods obtain a better forecasting quality, see Table 1. Only company C and D showed different results, for these companies the ZeroR method was better or comparable to the demandForest methods. These are also the companies for which the separate predictions of the profile and the total demand of demandForest were comparable or worse than the benchmark methods.

	Syn	A	B	C	D	E
demandForest	10.8	0.681	32.5	2.22	3.59	3.61
dF + Log-Normal	**10.7**	0.683	32.6	1.83	**3.03**	**3.31**
dF + Gamma	10.8	**0.676**	**32.0**	2.15	3.48	3.56
Proximity	13.2	0.705	40.2	2.57	4.11	5.36
ZeroR	15.2	0.784	36.4	**1.78**	3.15	4.24

Table 1: RMSE of combined forecasts for each method and data set

For the inventory performance of demandForest and the benchmark methods, we evaluated the methods with four inventory management cases. In these cases, we varied the lead time of the products. Three cases included a replenishment policy and had lead times of respectively zero, two, and six weeks. The last case included only a one-time order for all new products.

In these cases, we evaluated the consistency between the quantiles and the Cycle Service Levels. The CSL is the probability of not having a stock out during a inventory cycle, whereas a quantile can be used as a target service level. Hence, it is expected that the quantile (i.e., the target service level) is similar to the CSL (i.e., the actual service level). Considering all data sets, the demandForest methods showed the most consistency between the quantiles and the CSLs. Nevertheless, there were no significant differences between the demandForest method and its extensions. For all methods, the performance decreased when the lead time of the products decreased. Hence, the application of the profiles is not able to maintain the service levels which are calculated for the total demand. Comparing the demandForest methods with the Proximity method, we observe large improvements. While the Proximity method imitates the current behaviour of supply chain planners, it was the least reliable approach. The Proximity method resulted in both too high and too low service levels. The absolute deviation of the CSLs from the target service levels of 75%, 90%, and 95% decreased with respectively 3.2, 8.1, and 9.3 percentage points, when comparing the demandForest methods with the Proximity method. Hence, demandForest can greatly improve the current reliability of the target service levels. The other benchmark method, ZeroR, provided accurate results for the synthetic data set, and for company A and B. However, for company C, D, and E ZeroR provided the least accurate CSLs. Hence, ZeroR is less robust and less suitable to apply at a wide range of companies.

Regarding the inventory costs, the demandForest methods were again the most robust meth-

ods. Between the demandForest methods, no large differences were observed. These methods generally obtained the lowest inventory costs, with a few exceptions. The Proximity method often achieved comparable costs and sometimes the lowest costs. Comparing the demandForest methods with the Proximity method for the service levels of respectively 75%, 90%, and 95%., the costs for the demandForest methods were on average 18%, 11%, and 17% lower than the Proximity method. ZeroR resulted often in much higher costs, except for company C and E. For company C, it resulted in the best performance for the replenishment cases. For company E, it did not result in the lowest costs, but it achieved better results for the highest range of service levels.

To conclude, with a few exceptions, the demandForest methods obtained best forecasting quality, most reliable service levels and lowest inventory costs. The extensions with the fitted theoretical distributions did not significantly improve the results. Considering the industry data sets, the best performances were obtained at the companies with a large number of products in the data set, at least 5 product characteristics and a varied mix of demand types. The main weakness of demandForest are the profiles. These cannot always be clearly distinguished and predicted, and also depreciated the consistency of the service levels in inventory management.

Recommendations

We advise Slimstock to keep track of introduction dates, save historical forecasts and if possible, more product characteristics. In this way, the quality and amount of data can improve, which benefits the forecasting. Furthermore, we recommend to implement demandForest in SQL Server with Machine Learning Services as an integrated pilot version, such that the data gathering and calculations can be performed in-database. Besides demandForest, Slimstock can also provide its clients with additional insights, such as a top 5 comparable products and the influence of each product characteristic on the new product demand. Both of these additional insights can be extracted from the Random Forest algorithms. Future work can focus on improving the profiles, focusing of specific groups of products, including more predictive features besides product characteristics, updating the demandForest forecasts in the introduction period when new data becomes available, and investigating other machine learning algorithms than Random Forest and Quantile Regression Forest.

Preface

It is my pleasure to present to you my master's thesis, which is the result of more than half a year of research. As one of the last steps of my master Industrial Engineering and Management, it marks the end of a joyful period as a student. I had the privilege of writing this thesis during my internship at Slimstock. They provided me with the challenge of predicting the unknown. By taking on this challenge, I learned a lot and that would not have been possible without the support and confidence of many people.

First of all, I would like to thank Thijs for his daily supervision, creative ideas, proofreading, useful insights, and boundless discussions. I also would like to thank Bart for providing me with the opportunity to work on this project and his strategic and practical advice. Furthermore, I would like to thank all the colleagues of Slimstock. You made me enjoy my time at the office and we had fruitful discussions during our walks.

Furthermore, I would like to thank Martijn for being my first supervisor at the university, for his confidence throughout this process, and for his proofreading, useful comments and advice. I also would like to thank Karin for her second opinion and additional advice and knowledge.

Lastly, I am grateful for the support of my friends and family. A special thanks to my girlfriend, for her unconditional encouragement, support, and love during my graduation period. For now, this thesis might be the end of my internship and student life, more is yet to come and I am ready for the next step!

Robert
October 2019

List of Abbreviations

ADI Average inter-Demand Interval. 8, 9, 75

ANN Artificial Neural Network. 24–29, 76

CH Caliński-Harabasz. 11

CSL Cycle Service Level. ii, vii, 32, 35, 41, 42, 45, 46, 56–63, 67, 68, 71, 73

CV² Squared Coefficient of Variation. 8, 9, 75

DT Decision Tree. 23, 24

EOQs Economic Order Quantities. 42

FR Fill Rate. 32, 41, 56

GB Gradient Boosting. 25, 76

KPI Key Performance Indicator. 30, 32, 33, 39–41

MAE Mean Absolute Error. 30–32

MAPE Mean Average Percentage Error. 30–32, 41

MLR Multiple Linear Regression. 21

MOQs Minimum Order Quantities. 42

MSE Mean Square Error. 30–32

OOB Out-Of-Bag. 24, 25, 38, 39, 45, 47, 51

PICP Prediction Interval Coverage Probability. ii, 31, 32, 40, 45, 52, 53, 70

PINAW Prediction Interval Normalised Average Width. ii, 32, 40, 45, 52, 53, 70

QRF Quantile Regression Forest. iii, 25, 28, 29, 34–36, 38–41, 45, 47, 48, 52, 69, 70, 72, 73

RF Random Forest. iii, 23–25, 28, 29, 34, 35, 37–41, 45, 47, 48, 50–52, 70, 72–74

RMSE Root Mean Square Error. ii, 30–32, 39–41, 45, 47–52, 56, 61, 70

sMAPE Symmetric Mean Average Percentage Error. 30–32, 41

SVM Support Vector Machine. 27, 28, 76

Contents

Management summary **i**

Preface **iv**

List of Abbreviations **v**

1 Introduction **1**

1.1 Scientific context . . . 1
1.2 Problem statement . . . 2
1.3 Research objective . . . 3
1.4 Research questions . . . 4

2 Preliminary analysis of new product data **6**

2.1 Data from industry partners . . . 6
2.2 Demand characteristics of new products . . . 8
2.3 Identifying possible demand clusters . . . 10
2.4 Relations between product characteristics and demand . . . 12
2.5 Conclusion on the analysis of new product data . . . 15

3 Literature review **17**

3.1 New Product Forecasting . . . 17
3.1.1 Diffusion models . . . 18
3.1.2 Direct demand forecasting by analogous products . . . 19
3.1.3 Conclusion on new product forecasting methods . . . 20
3.2 Machine learning techniques . . . 21
3.2.1 Multiple linear regression . . . 21
3.2.2 Logistic regression . . . 22
3.2.3 Decision Trees . . . 23
3.2.4 Random Forests . . . 23
3.2.5 Gradient Boosting . . . 25
3.2.6 Artificial Neural Networks . . . 25
3.2.7 Support Vector Machine . . . 27
3.2.8 Summarising table . . . 28

3.2.9 Conclusion on machine learning techniques . . . 28
3.3 Performance indicators for machine learning and forecasting . . . 29
3.3.1 Classification performance . . . 30
3.3.2 Regression and forecasting performance . . . 30
3.3.3 Interval performance . . . 31
3.3.4 Inventory management performance . . . 32
3.3.5 Conclusion on performance metrics . . . 32

4 Proposed method and experimental design 34
4.1 demandForest . . . 34
4.2 Extended demandForest . . . 35
4.3 Defining benchmark methods . . . 36
4.4 Experimental design . . . 38
4.4.1 Training the proposed method . . . 38
4.4.2 Testing demandForest . . . 39
4.4.3 Application to inventory management . . . 41
4.4.4 Synthetic data set . . . 43
4.4.5 Overview of experiments . . . 44

5 Experimental results 47
5.1 Training the algorithms . . . 47
5.1.1 Finding the best value for *mtry* . . . 47
5.1.2 Feature importance . . . 48
5.1.3 Conclusion on tuning and feature importance . . . 51
5.2 Testing the methods with unseen data . . . 51
5.2.1 Predicting the profile . . . 51
5.2.2 Predicting the total demand during the introduction period . . . 52
5.3 Forecast accuracy of the proposed methods . . . 56
5.4 Inventory performance of the methods . . . 56
5.4.1 Consistency between quantiles and CSLs . . . 57
5.4.2 Inventory costs for different service levels . . . 60
5.5 Conclusion . . . 70

6 Conclusion and recommendations 72
6.1 Conclusion . . . 72
6.1.1 Scientific contribution . . . 73
6.1.2 Limitations . . . 73
6.2 Recommendations . . . 74
6.3 Directions for future research . . . 75

Bibliography 76

A Appendix Data Sets **82**

B Appendix Clustering **83**

C Appendix Synthetic Data Set **85**

D Appendix Prediction Intervals **89**

1 Introduction

This thesis describes the research performed at the development department of Slimstock. Slimstock is founded in the Netherlands in 1993 and is a knowledge partner in inventory optimisation. With around 350 professionals, Slimstock is European market leader and serves more than 1000 clients all over the world. Its main software solution contains forecasting, demand planning, supply chain optimisation, and inventory management. Besides software solutions, Slimstock offers project-based support and professional services, including coaching & training sessions, analytics, and interim professional support. Slimstock provides its clients with the tools and knowledge to generate insights and to reduce inventories while at the same time increasing the service levels. Slimstock is organised in several departments; the responsibility of the development department is to further develop and improve the software solution. This research, which forms the graduation assignment of the master's programme Industrial Engineering and Management at the University of Twente, aims to pave the way for an improved scalable new product forecasting method for Slimstock and its clients.

This chapter further introduces this research. In Section 1.1, we analyse the context in which our research resides. This leads to the problem statement in Section 1.2. Afterwards, we define the research objective in Section 1.3. Finally, in Section 1.4 we introduce the research questions.

1.1 Scientific context

For years, stock-keeping and manufacturing companies are struggling with new product forecasting, which is one of the most critical and difficult management tasks (Assmus, 1984). New product forecasting is challenging compared to forecasting demand of existing products since historical data is not available as an indicator of future demand. Additionally, there is limited analysis time and there exists a general uncertainty related to consumer acceptance and competitive reactions (Assmus, 1984; Voulgaridou, Kirytopoulos & Leopoulos, 2009; Lee, Kim, Park & Kang, 2014; Goodwin, Meeran & Dyussekeneva, 2014; Baardman, Levin, Perakis & Singhvi, 2018). Despite its complexity, these initial forecasts are essential for the operations of a company since they guide important decisions like capacity planning, procurement and inventory control (Voulgaridou et al., 2009; Wright & Stern, 2015; Baardman et al., 2018). Because these decisions are guided by forecasting, a proper sales forecasting approach is key to prevent complications during or right after the product launch. Poor forecasts can result in stock-outs or overstock situations, which both have a direct impact on the company's profitability and may also decrease customer satisfaction and market share (Lee et al., 2014; Basallo-Triana, Rodrıguez-Sarasty & Benitez-Restrepo, 2017; Loureiro, Miguéis & da Silva, 2018).

Global competition, increasing customer expectations, and technological innovations are decreasing the lifetimes of products in many industries (Basallo-Triana et al., 2017; Baardman et al., 2018). Shorter life cycles do not change the fundamental problem of forecasting sales of new products, but they force companies to produce forecasts more frequently. This increases the scale of the problem (Lee et al., 2014; Baardman et al., 2018). Therefore, the need for analytic

approaches to new product forecasting rises. Goodwin et al. (2014) argue that quantitative models should be at the core of the forecasting process of new products. Nevertheless, Kahn (2014) states the use of analytical methods for new product forecasting is still limited among companies. Due to the lack of data, qualitative methods involving judgment, experience, and intuition are usually applied for new products.

The use of qualitative methods is emphasised by Kahn (2014). Kahn pointed out that forecasting new products should focus on creating meaningful estimations to anticipate risks, whereas regular forecasting focuses on accuracy and reliability. Due to the inherent uncertainty and lack of historical data, it is difficult to quantitatively generate reliable and accurate forecasts for new products. Human judgment, experience, and intuition may be more useful for anticipating the risks. When the uncertainty increases, quantitative point forecasts may not be sufficiently reliable to use for operational decisions. In this case, the level of uncertainty can be conveyed by means of prediction intervals. By expressing the uncertainty of new product forecasts, companies can anticipate these risks in their decision-making (Kahn, 2014). Due to the uncertainties, prediction intervals can be a lot more informative than point forecasts. Despite these advantages, little attention has been paid to analysing the uncertainty among new products (Goodwin et al., 2014).

All in all, new product forecasting remains a difficult, but important task. The scale of new product forecasting is increasing, but there is still a limited use of quantitative methods and methods that consider the uncertainty of new product demand. This highlights the relevance and importance for research. In the next section, we zoom in on how this problem resides within the companies that work with the inventory management software of Slimstock.

1.2 Problem statement

Also the companies that manage their inventories with the software of Slimstock face an increasing number of product introductions. Since there is no historical data available for these new products, companies need to estimate the initial demand themselves.

The supply chain planners of the companies usually determine these forecasts and discuss them with managers during Sales & Operations Planning sessions. The initial forecasts rely on human judgment of planners and experts or on the historical sales of a predecessor or related product. This corresponds with Kahn (2002), who suggested that expert opinions, surveys and the average sales of similar products are the most widespread techniques for predicting demand of new products. Additionally, it can happen that a manager decides to purchase a large number of new products, for example due to a volume discount of the supplier. In that case, a forecast for the introduction period is not necessary anymore. Nevertheless, it can also result in excess stocks, when the products are not as popular as expected.

Considering the historical sales of similar products, many planners manually determine which products are comparable to a new product. This is not only time consuming, but these judgmental selections may also suffer from bias. Planners may select only one reference product, or only products that are easily recalled by the planners (Lee, Goodwin, Fildes, Nikolopoulos & Lawrence, 2007). Hence, there exists a possibility that not all relevant information is utilised. If insufficient information is used by planners, the potential risks can be underestimated (Kahneman & Tversky, 1977). These problems may increase when the number of introductions increases and the time available for creating a forecast decreases.

After a forecast is determined by a planner, the software of Slimstock calculates safety stocks and order levels for the new product assuming Normally distributed demand. These calculations do not only require a forecast, but also a standard deviation of the forecast. In that case, safety stocks can be determined according to a predefined service level. However, for new products is

the standard deviation unknown. To overcome this problem, Slimstock analysed the demand data of several companies to find a common factor for the standard deviation. The empirical analysis showed that a coefficient of variation of 0.45 provides a satisfactory estimation of the monthly demand variability. This factor is generally used for all new fast-moving products. Nevertheless, companies are able to adjust this factor to a value that is suitable for them.

Regarding the statistical methods used for general forecasting in the software of Slimstock, these require usually four weeks or months of historical data to generate forecasts. After four time periods, there is some data available to determine the future demand and the standard deviation for safety stock calculations. Since these methods require four weeks or months, we will focus on creating forecasts for the first four months of a new product. Since there is little to no data available in this period, it is a challenging period. Nevertheless, it is also an important period determining the success and growth of a new product.

Due to the importance of new product forecasting, Slimstock wants to support companies with their product introductions. Slimstock wonders if utilising the product characteristics may be valuable for new product forecasting. Within the software of Slimstock, product characteristics are usually available, such as the price and product category. The number and type of these characteristics differ per company. Currently, this information is not regularly used for new product forecasting. Nevertheless, comparable products can be identified based on these characteristics. Consequently, the historical demand of these existing products can be used for providing insights into the demand during the introduction period.

To use product characteristics for the forecast of an increasing number of product introductions at several companies, there is a need for an analytical method. This method should provide companies with an initial forecast for the introduction period. This research project intends to find a satisfactory solution, which brings us to the research objective.

1.3 Research objective

The objective of this research is to develop an analytical new product forecasting method to support supply chain planners. The method should create forecasts for the first four months after the introduction for new products before their introduction. For this method, we will utilise the characteristics of products that are available in the inventory management system of a company. The method should be able to compare the product characteristics of a new product with the product characteristics of all products that are previously introduced by the company. It should utilise the historical demand data of comparable products to obtain a forecast for the new product. The goal is to develop an method that can be implemented at multiple companies. Therefore, the method should be adaptable to the specific situation of a company. With this, we might make a serious impact by preventing significant under- and overstocking issues at a wide range of companies. Since there exists an inherent uncertainty for the demand of new products, the method should determine prediction intervals for the forecasts. These intervals quantify the uncertainty for the demand of a new product. Prediction intervals provide meaningful information for planners and managers to support decision-making. Additionally, the method should also be validated sufficiently, to prove that it can improve the current processes within several companies. It should be able to provide robust forecasts and also lead to improvements within inventory management.

This leads to the following research objective:

> *"Develop and validate an analytical method that provides pre-launch forecasts for the first four months of demand of new products. The method should utilise product characteristics of new and existing products to generate forecasts with prediction intervals, based on the historical demand of existing comparable products."*

To achieve the objective of this research, we divide the research into several parts. For each part, we define one or more research questions. These research questions are stated in the next section.

1.4 Research questions

The following research questions are defined to obtain a satisfactory result towards to objective. First, we will analyse the data of new products that is made available by clients of Slimstock. We want to explore the available data of the companies for patterns and relations that might be useful for forecasting.

1. Are there certain patterns and relations within the data of new products?
   (a) Which data is available for analysis?
   (b) Are there patterns in the demand during the introduction period?
   (c) Are there certain relations between the product characteristics and demand?

Second, we study the literature that is related to our research. In the literature, we analyse what methods are previously applied to new product forecasting problems. We also investigate several statistical methods and machine learning algorithms to discover which methods are valuable to our research. Additionally, we discuss performance metrics used in literature to evaluate machine learning algorithms and forecasting methods.

2. Which methods are applied to new product forecasting according to literature?
3. Which statistical method or machine learning algorithm in literature is most suitable for this research?
4. What are metrics to evaluate the performance of machine learning algorithms and forecasting methods?

Afterwards, we want to combine the data exploration with the literature review to design a forecasting method that is most promising towards our research objective. We should also define how this method can be validated to prove its usability in practice.

5. What forecasting method do we propose to improve new product forecasting?
   (a) What steps are taken within the method?
   (b) How can the method be compared to the current situation?
   (c) How should the performance of this method be evaluated?

At last, we want to analyse and validate the proposed forecasting method. This analysis is two-fold. First, the forecasting method will be evaluated according to the forecasting performance. Second, the method will be evaluated according to the performance within inventory management.

6. What is the expected quality of the forecasting method?
    (a) What is the quality of the forecast when applied at different companies?
    (b) What is the expected impact of the method on inventory management?

Answering these research questions should collectively lead to achieving the research objective. The remainder of this thesis is structured as follows. In Chapter 2, we explore the data that is made available for this research by several clients of Slimstock. In Chapter 3, we review literature regarding new product forecasting, statistical methods, machine learning algorithms and performance metrics. Thereafter, we propose our methods in Chapter 4 and describe the experimental design of the evaluation procedure. In Chapter 5, we analyse the results and performance of the proposed methods. At last, we provide in Chapter 6 our conclusions and recommendations.

2 Preliminary analysis of new product data

In this research, we want to investigate the possibilities to leverage product characteristics to generate pre-launch forecasts for new products. Although there is no evident relation between product characteristics and demand, intuitively there should exist certain patterns between these to use them for forecasting. Therefore, in this chapter, we analyse the data sets made available by industry partners. First, we describe the data gathering and give a summary of the data sets in Section 2.1. In Section 2.2, we explore the demand characteristics. In Section 4.4.2, we investigate if we can identify similar demand patterns among the products. Afterwards, we analyse the relations of the product characteristics with the demand of new products in Section 2.4. Lastly, in Section 2.5 we conclude the exploratory data analysis and suggest further steps.

2.1 Data from industry partners

In this section, we give a brief summary of the data that is obtained from companies that use the inventory management software of Slimstock. In total, we received the data of 5 different companies. In Table 2.1, a brief description of each company is given.

Company	Industry type	Market	Type of products
A	Retail	B2C	Household items
B	E-commerce	B2B & B2C	Lighting
C	E-commerce	B2C	Sanitary ware
D	Wholesale	B2B	Agricultural machinery (spare) parts
E	Wholesale	B2B	Garden tools and forestry machines

Table 2.1: Brief description of each industry partner

For all five companies, we retrieved data from the databases of the inventory management system. The specific data that was available differed per company. Below we will describe how we gathered the data.

2.1.1 Data gathering process

Five companies made their data available for this research. In this section, we describe the process of data gathering. We retrieved the product characteristics and historical sales data from the databases of the inventory management systems of the companies. Gathering the product characteristics was straightforward, but the historical sales data required more attention. For this research, the obtained historical sales data is recorded weekly.

Unfortunately, company A is the only company that keeps track of the date at which products

are introduced. For the other companies, we needed to make an assumption to determine when a product was introduced. We assume that the week of introduction is the week in which the first sale occurs. Unfortunately, this results in a bias in the data, since products may be introduced earlier and not sold directly. Spare parts are an example of products that are not likely to be sold directly after the introduction. Therefore, the inability to retrieve the actual introduction date is a limitation of the data. We need to take this bias into consideration for the remainder of this research.

Once we determined the date of introduction, we selected the sales data of the first 18 weeks (approximately 4 months, which is the research objective). Naturally, we only selected products of which the first 18 weeks of historical sales were available. From these products, only products that are kept in inventory were selected. Products that were in total more than 14 days out of stock during these 18 weeks were excluded from the selection, since the historical sales data of these products does not reflect the actual demand properly. The amount and date of returned products is also recorded in the software. When these returns were put back in inventory, we deducted the amount from the sales in that week. Since it may be possible that a product is returned in a week in which nothing was sold, we assume a minimum demand of zero.

The data of company A required some further processing due to its supply chain. Company A sells their products through more than hundred shops, while they also keep inventory centrally at a distribution centre. Shops are replenished from the distribution centre. When they introduce a new product, they centrally purchase the item and sell these through the shops. When gathering the sales data, we retrieved the sales data from the individual shops. We aggregated the shop data to obtain an estimate of the overall sales data at the distribution centre. The products of company A are not necessarily introduced in all shops. Some products are only sold locally at a few shops. Since we only want to consider products that are stocked and distributed from the central distribution centre, we only select products that are introduced in at least 10 shops. Furthermore, products are not always introduced at the same moment in each shop. Therefore, we assume that the timing of the introduction is independent of the sales in each shop. We aggregate the demand of the first sales week of each individual shop (which may not be the same calendar week for each shop), and repeat this for the other 17 weeks. Afterwards, we divide the aggregated demand by the number of shops to obtain the average demand per shop for a new product. And overview of the retrieved data is given in Table 2.2. In the table, the time frame in which the new products are sold is given. Additionally, the product characteristics are mentioned and the total number of introduced products for which the data is gathered at each company.

Company	Time frame	Characteristics	Number of products introduced
A	11-2017 to 02-2019	Supplier, category, sales price, margin, collection type, product group, space facing, circle type	16.229
B	06-2014 to 01-2019	Supplier, category, purchase price, sales channels, article type	3197
C	10-2016 to 12-2018	Supplier, category, subcategory, subsubcategory, sales price, margin, product type, brand, brand collection	592
D	09-2017 to 03-2019	Supplier, category, purchase price, brand	1172
E	06-2016 to 03-2019	Supplier, category, purchase price, brand	660

Table 2.2: Brief description of the data sets

The characteristics of the products are characteristics defined by the companies themselves. Company A has put effort in their categories and product groups. These product characteristics are numerical instead of categorical. Therefore, comparable categories and groups have values close to each other. For example, the categories with electrical devices all have a value between 300 and 399, whereas categories with tableware have a value between 800 and 899. Therefore, we can handle these categories and groups as ordinal numbered data. Additionally, the collection type, space facing, and circle type are also numerical. Therefore, the only categorical characteristic at company A is the supplier. For the other companies, all product characteristics, except the prices and margins, are categorical. For example, the data set of company E contains 48 suppliers, 8 categories and 44 different brands. For descriptions and more information on the characteristics of the different companies, we refer to Appendix A. In the next sections, we perform a preliminary analysis for better understanding of the data. Before analysing the different product attributes and their relations to the demand, we first investigate the characteristics of the demand itself.

2.2 Demand characteristics of new products

In this section, we analyse the demand characteristics during the introduction period of new products. For a company, the ideal situation is that the demand of a new product increases gradually after the introduction. However, in the data sets we do not observe this kind of pattern clearly. On the contrary, we observe a lot of weeks with zero demand and also a lot of variety in demand size. To create an overview of all data sets, we will investigate this behaviour. In the next subsections, we classify the data into several demand types according to the four types of demand defined by Syntetos, Boylan and Croston (2005).

2.2.1 Types of demand patterns

To classify the demand based on characteristics, Syntetos et al. (2005) developed a matrix to distinguish four types of demand: (1) smooth demand, (2) intermittent demand, (3) erratic demand, and (4) lumpy demand. The division into these demand types is based on values of two parameters: the Squared Coefficient of Variation (CV^2) and the Average inter-Demand Interval (ADI). These parameters measure the variation in demand quantities and the regularity of demand in time. The cv2 is the standard deviation of the demand, and the ADI is the average interval between two consecutive demands. The CV^2 and ADI are given by (Costantino, Di Gravio, Patriarca & Petrella, 2018):

$$CV_i^2 = \left(\frac{\sqrt{\frac{\sum_{n=1}^{N_i} \left(d_i^n - d_i\right)^2}{N_i}}}{d_i} \right)^2 = \left(\frac{\text{stdev}\left(d_i^n\right)}{\text{mean}\left(d_i^n\right)} \right)^2 \tag{2.1}$$

$$ADI_i = \frac{\sum_{n=1}^{N_i} t_i^n}{N_{p_i}} = \text{mean}\left(t_i^n\right) \tag{2.2}$$

where
i is the product
N_i represents the number of periods with non-zero demand of product i
d_i is the average demand when the demand of product i is non-zero: $d_i = \frac{\sum_{n=1}^{N_i} d_i^n}{N_i}$

With these two parameters, the demand can be classified as follows (Syntetos et al., 2005):

- Smooth demand (regular demand over time, with limited variation in quantity)
  - $CV^2 \leq 0.49$ and $ADI \leq 1.32$
- Intermittent demand (sporadic demand with limited variation in quantity)
  - $CV^2 \leq 0.49$ and $ADI > 1.32$
- Erratic demand (regular demand over time, but large variation in quantity)
  - $CV^2 > 0.49$ and $ADI \leq 1.32$
- Lumpy demand (sporadic demand with large variation in quantity)
  - $CV^2 > 0.49$ and $ADI > 1.32$

The classification based on the variability of the demand size and the length of inter-demand intervals can be linked to the ability to forecast a product accurately. The higher the variability of the demand size or the longer the inter-demand intervals, the harder it is to forecast the demand pattern. Hence, forecasting 'smooth' demand is the easiest. On the other hand, accurately forecasting 'lumpy' demand is the most difficult. With 'lumpy' demand, both the variability in size and inter-demand intervals is high.

2.2.2 Classification of demand data

For the data sets of all 5 companies, we will classify the items into the four types of demand. Since we gathered the weekly demands of the products at each company, the length of one period is one week. By this classification, we obtain a better overview of the type of new products at each company. The distributions of the demand over the different demand types are shown in Table 2.3.

Company	A		B		C		D		E	
CV^2	≤ 0.49	> 0.49	≤ 0.49	> 0.49	≤ 0.49	> 0.49	≤ 0.49	> 0.49	≤ 0.49	> 0.49
$ADI > 1.32$	8.6%	8.9%	23.0%	39.0%	86.0%	8.8%	93.2%	6.7%	81.5%	14.7%
$ADI \leq 1.32$	45.1%	37.4%	14.6%	23.4%	3.4%	1.9%	0.2%	0.0%	0.6%	3.2%

Table 2.3: Distribution of the demand types per company

Based on the classification of demand types according to the matrix of Syntetos et al. (2005), a lot of introduced products show sporadic demand patterns. Furthermore, company B introduces products with higher demand variability. Only the products introduced by company A show smooth demand patterns. This can be explained by the aggregation of the demand of each shop. Predicting the demand becomes more difficult when the sporadicity and variability increases. Considering the demand patterns of the companies, predicting the demand of the new products accurately will be rather difficult, especially for company B.

When we further look into the data, we observe that a lot of products only have sales in the first week (Company A: 0.2%, B: 2.2%, C: 29.9%, D: 68.2%, E: 50.3%). This implies an intermittent demand pattern with a CV^2 of 0 and ADI of at least 18 weeks. That the product is only sold in the first week can have multiple reasons. The product can be a slow mover, it might be unsuccessful or since company D and E are wholesalers, the products are sold in

large quantities to fill the shops of their customers and these customers order a new batch of products a few months later. That these products are slow movers is also very likely. The specific parts or larger sanitary goods sold by company C, such as specific baths, are also sold infrequent. The wholesale companies D and E sell specialist products and spare parts, which are often intermittent slow movers (Bucher & Meissner, 2011). Company A and B introduce a less products that are only sold in the first week compared to the other companies. This can be explained by the different markets of these companies, the aggregation of company A and the higher sale volumes of company B.

Besides classifying demand patterns, previous studies identified distinctive groups with similar demand patterns. When there exist specific demand patterns of new products, it may be possible to predict these patterns beforehand. In the next section we investigate whether such clusters exist.

2.3 Identifying possible demand clusters

Multiple studies have clustered demand patterns of products and computed distinctive groups of products. Some studies have tried to predict these clusters for new products with machine learning techniques, such as decision trees and neural networks, based on the product characteristics (e.g., Thomassey and Fiordaliso, 2006; Thomassey and Happiette, 2007; Hibon, Kourentzes and Crone, 2013). Other researchers computed specific algorithms for predicting the demand of each cluster of products (e.g., Lu and Wang, 2010; Lu and Kao, 2016; Huber, Gossmann and Stuckenschmidt, 2017). In this section, we also investigate if we could identify certain profiles in the demand data of the new introduced products of each company. If it is possible to identify distinctive profiles, it is interesting to predict these profiles. Such profiles can provide valuable insights for a planner about the demand over time.

For clustering the demand patterns, we use the normalised cumulative demand. By normalising the demand, we can compare the demand patterns despite the total demand. By using the cumulative demand, we aim to find a better overall result. Since this research coincides in an inventory management environment, it is less important if the demand is in week 8 or 9. By cumulating the demand, we are more close to the resulting effects on the stock levels. Whether there is for example demand forecasted in week 8, but occurs in week 9, the stock levels only differ in one week, the time between the forecasted and actual sale. By cumulating the demand, we observe the same difference. However, when using the non-cumulative demand, we observe a difference in both week 8 and week 9.

To determine the normalised cumulative demand, we first normalise the demand. The demand a per week i is scaled by dividing the demand in each week by the total demand of 18 weeks to obtain the normalised cumulative demand n in week i:

$$n_i = \frac{\sum_{x=1}^{i} a_x}{\sum_{x=1}^{18} a_x} \qquad \forall\, i \in 1 - 18 \tag{2.3}$$

For clustering the demand patterns, we use the k-means algorithm. The k-means algorithm is the most widely used clustering method in practice and is effective and efficient in most cases (Wu et al., 2008; Jain, 2010). It is also applied to cluster demand patterns (Espinoza, Joye, Belmans & De Moor, 2005; Thomassey & Fiordaliso, 2006; Lu & Kao, 2016; Huber et al., 2017). Since k-means is a greedy algorithm, it converges to local minima based on the initial partition. To overcome this limitation and increase the chance of finding the global minima, the algorithm can be run with multiple initial partitions. The partition with the smallest sum of the squared error can be chosen as final partition (Jain, 2010).

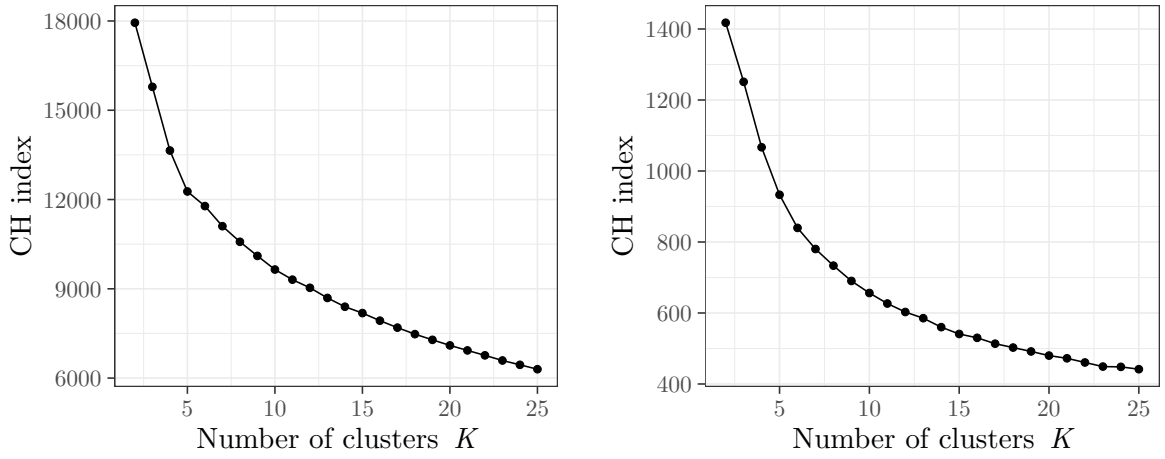
The most important parameter choice of the k-means algorithm is the number of clusters k (Jain, 2010). To determine the number of clusters, the Caliński-Harabasz (CH) index is used. This index was proposed by Caliński and Harabasz (1974). The CH-index is a competitive index for finding the number of clusters (Milligan & Cooper, 1985; Arbelaitz, Gurrutxaga, Muguerza, PéRez & Perona, 2013). It is calculated for the number of clusters k with:

$$CH(k) = \frac{N-k}{k-1} \frac{BCSS(k)}{WCSS(k)} \tag{2.4}$$

Where N is the number of observations, k the number of clusters, $BCSS$ the between cluster sum of squares, and $WCSS$ the within cluster sum of squares. The number of clusters with the highest CH-index indicate the optimal number of clusters. We implement the k-means algorithm in the R environment (R Core Team, 2014). To increase the chance of finding the global minima, according to Jain (2010), we run the algorithm 25 times. To determine the optimal number of clusters, we determine the CH-index for 2 to 25 clusters.

2.3.1 Clustering results

For all companies, the optimal number of clusters is 2. In Figure 2.1, the values of the CH-index of the k-means clustering of the demand data of company A (Fig. 2.1a) and E (Fig. 2.1b) are portrayed in a graph. The other companies all had similar decreasing values of the CH-index, which can be found in Appendix B.



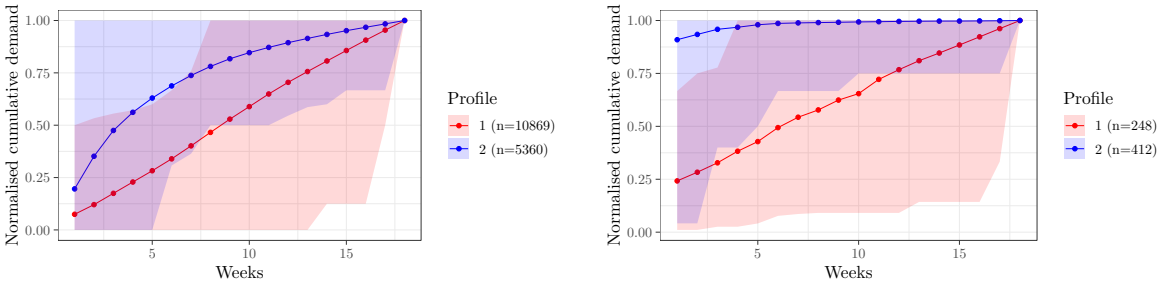
(a) Company A

(b) Company E

Figure 2.1: Optimal number of clusters

When calculating the cluster centres with two clusters, we obtain quite similar shapes among all companies. The cluster centres, or profiles, and ranges of company A and E are displayed in Figure 2.2. Each company has one concave increasing profile and one rather linear profile. This means that some products are mainly sold in the first weeks after the introduction, while other products are more stable or somewhat increasing. The second profile of company E, Figure 2.2b, already starts at a normalised cumulative demand of 0.90. This means that the demand of these products are mainly in the first week, and only some demand in the weeks following. This large differences in the first week is this the main distinction between the two profiles. In the other weeks, has the profile with a high demand in the first week lower demands later. These patterns within the first week and the other weeks can also be observed for the data sets of company C and D, similar to the results of company E. The results of company B are more similar to

company A. Interestingly, company A and B are the companies with the largest number of new products and a larger variety of demand types.



(a) Company A

(b) Company E

Figure 2.2: Cluster centres with $k = 2$

The cluster with the peak in the first week can be explained by the bias in the data sets. In Section 2.2, we noticed that there exist a lot of products with only one sale. Due to the bias in the data sets, these single sales are always in the first week of the demand data. Therefore, the profile with the peak in the first week implicitly makes a distinction between products that are sold more regularly (Profile 1) and products that are sold only once (Profile 2). Nevertheless, the profiles show some of overlap with each other. This can be explained by the variable and sporadic demand patterns. Therefore, the profiles should be handled with care.

2.4 Relations between product characteristics and demand

In the previous sections, we identified the characteristics in the demand data to find which parts of the demand should be predicted. In this section, we explore the possible relations between the product characteristics and the demand.

In Figure 2.3, we plot the total demand of each product in the first 18 weeks of company B relative to the price of a product. It is clear that if the price relatively high, products are sold less. With low prices, the demand of a product can be either high or low. The majority of the products seems to have a low price and low demand. At the other companies, we observe similar patterns between the price and the demand.

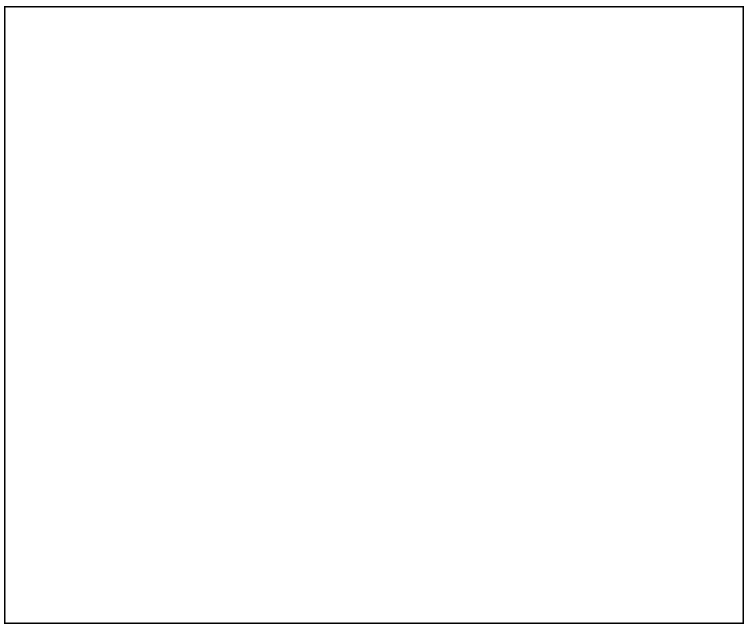


Figure 2.3: The total demand relative to the price at company B

A less clear pattern is visible when we compare the demand to the product margin. For company C (Figure 2.4), the relation between the margin and the demand seems irregular. The outliers of the demand coincide within the more frequent margin ranges.

Figure 2.4: The total demand relative to the margin at company C

Besides the price and margin, most product characteristics are categorical. Many of these categorical characteristics have many different levels. For example, the assortment of company C consists of 98 different brands. Some categories also consist of a lower number of levels, which is easier to visualise. Figure 2.5 shows the boxplots of the demand per product among the different brands of company D. Brand C is the brand with the largest number of products, while brands F and G show the smallest boxplots with only a few products. Figure 2.6 displays the demand per product in each category of company E. Half of the assortment belongs to category

B, while the boxplot of this category is rather small. Both category F and H also show similar low demand volumes. On the contrary, the demand of products in category A and G are rather high and show less overlap with the other categories.

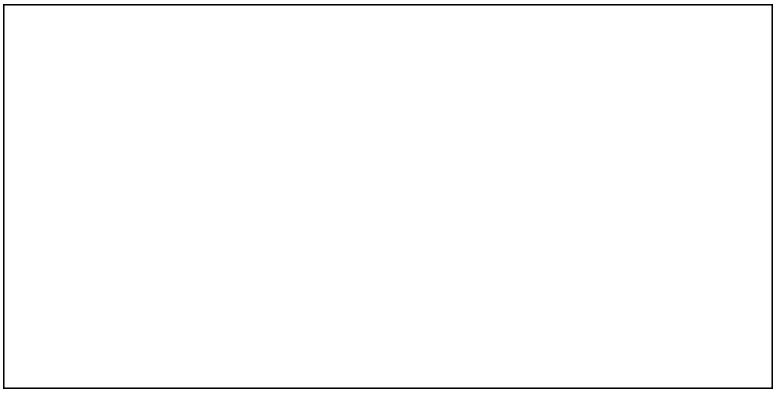


Figure 2.5: The total demand per brand at company D

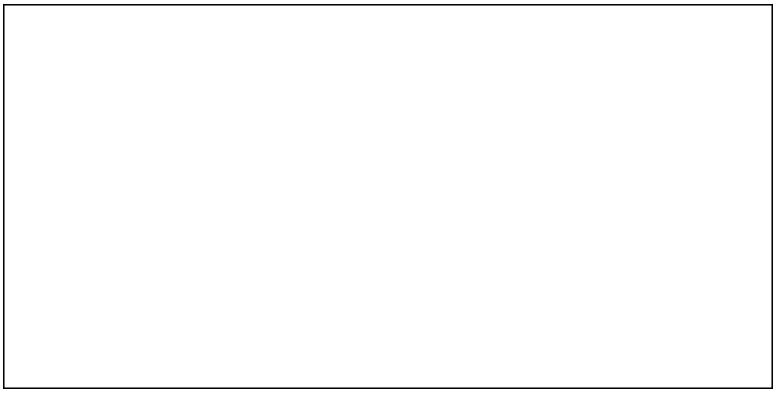


Figure 2.6: The total demand among product categories at company E

2.4.1 An example of analogous products

The figures in the previous sections provided overall relations between the characteristics and the demand. In this section, we picked some individual products with matching characteristics. Table 2.4 provides an overview of the characteristics and the total demand during the introduction period of three products from Company B.

Article	Supplier	Category	Price (€)	Sales channel	Article type	Demand
X	**sup1**	**cat1**	**8.00**	**chan1**	**stat1**	**181**
Y	sup2	**cat1**	**9.00**	chan2	stat2	**186**
Z	**sup1**	cat2	17.12	**chan1**	**stat1**	300

Table 2.4: Three products with some matching characteristics

In the table, we see that product X and Y have the same category, a similar price and also a similar demand. Product X and Z are also comparable, regarding the supplier, sales channel and article type, but the demand is almost twice as high for article Z. Now we look at the normalised demand pattern throughout the weeks. These are visualised in Figure 2.7. In this figure, we observe that product X and Y have different patterns throughout the weeks, while product X is very comparable to Z. Hence, if we want to make a forecast of product X, we should ideally use

the total demand of product Y and the pattern of product Z. However, this will probably not be the case with other products, although it is likely that some combinations of characteristics can point out certain patterns and ranges of the demand.

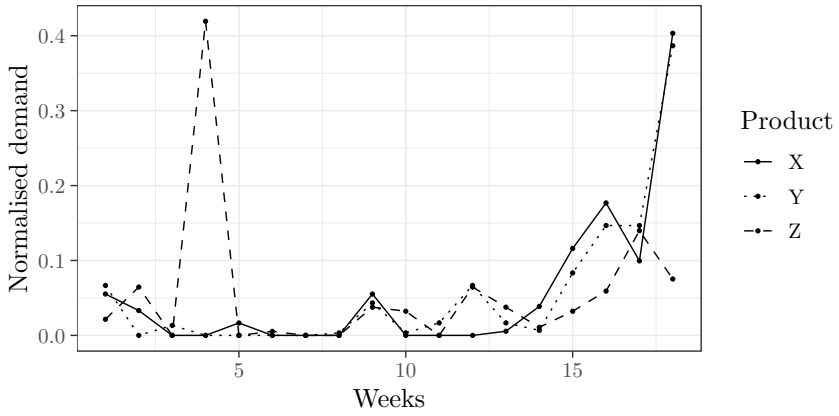


Figure 2.7: Demand patterns of three products of company B

2.5 Conclusion on the analysis of new product data

In the analysis of this chapter, we saw different results for the products of each company. This makes sense, since each company sells different types of products and serves different markets. Nevertheless, also some similarities exist within the demand and product characteristics.

By classifying the demand on the squared coefficient of variation and the average inter-demand interval, it became clear that the companies have different type of demand patterns. Company C, D and E have mostly intermittent demand patterns, of which a large percentage of the products only sell once in the first 18 weeks. Only a few products at each company are sold very frequent.

Besides the classification of the demand, we clustered the demand to identify profiles. Clustering the demand did not lead to several distinctive groups of products. Nevertheless, it distinguished two profiles: one with the majority of the demand in the first week, and another profile with demand more spread throughout the weeks. These two profiles distinguish the two main demand patterns that are also found in the demand classification. It must be noted that the two profiles showed some of overlap due to the variable and sporadic demand patterns. Hence, they can be used to predict different progressions through the introduction period, but should be handled with care.

Some relations between the product characteristics and the demand were found. The most clear pattern was found between the price of a product and the demand. It seems that more expensive products are not sold frequently in either data set. When the price is lower, the demand of a product shows a lot more variability. A new product with a low price may be sold once, but can also sold very frequently. The total demand of products show different patterns between several categories or brands, but no clear distinctive patterns were pointed out in the analysis. Additionally, combinations and interaction effects between products were not investigated. Considering all characteristics and data sets, such kinds of analysis become rather complex.

Due to the different product characteristics and demand patterns of each company, it is impossible to define general rules to generate a forecast. To uncover potential significant and clear relations, advanced analytical approaches will be required. Hence, in the remainder of this

study, we need to develop a model that can adapt to the specific data sets to generate insights into the demand of new products. In this case, the model should consider the different product characteristics. The field of machine learning provides such models, since this field studies algorithms that have the ability to learn without being explicitly programmed. Therefore, we will analyse scientific literature in the next chapter about machine learning algorithms. We also describe what research is already performed on similar cases of new product forecasting.

3 Literature review

In this chapter, we investigate what is currently known in literature about new product forecasting and machine learning techniques. First, in Section 3.1, we elaborate on new product forecasting and the current methods applied in literature. Second, we discuss a variety of machine learning techniques in Section 3.2. Lastly, we elaborate about several performance metrics used for machine learning and forecasting in Section 3.3.

3.1 New Product Forecasting

New product forecasting has a unique nature compared to forecasting existing products (Kahn, 2014). Kahn (2014) highlights that the most distinguishing factor is the lack of historical data with new product forecasting. Therefore, common forecasting methods such as simple exponential smoothing, Holt-Winters or Autoregressive Integrated Moving Average (ARIMA) are not possible to apply to new products. Hence, assumptions have to be made. Due to the lack of data, qualitative methods involving judgment, experience, and intuition are usually applied for new products. However, Goodwin et al. (2014) argue that quantitative models should be at the core of the forecasting process of new products.

As mentioned in Section 1.1, forecasts for new products are vital for operational decisions such as capacity planning, procurement and inventory control. Contrastingly, the use of analytical methods for new product forecasting are still limited (Kahn, 2014). Especially methods that evaluate the level of uncertainty has been deserving little attention in research (Goodwin et al., 2014). Forecasting models that analytically forecast the demand of new products and evaluate uncertainties have growing importance, since companies introduce products more frequently (Lee et al., 2014; Baardman et al., 2018). The risks of these new products should be anticipated in decision making using meaningful forecasts (Kahn, 2014).

New product forecasting methods can be categorised as judgmental, customer/market research or quantitative (Kahn, 2006). Judgmental models are based on the opinion of experts and stakeholders. The goal is to generate forecasts from experience, intuition, and judgment. Consumer/market research focuses on the potential response of consumers and markets by means of pre-tests. This is usually conducted during the product development process (Mas Machuca, Sainz Comas & Martınez Costa, 2014). Quantitative methods are mainly based on analogous forecasting to overcome the lack of demand history (Green & Armstrong, 2007). This means that these models use data of similar products to generate forecasts for new products.

In this literature review, we focus on the quantitative analogous forecasting methods. The advantage of these methods compared to judgmental methods and customer market research is that they can be computed and are therefore relatively time efficient when the number of product introductions increases. The critical assumption with analogous forecasting is that similarity between products translate into similar demand patterns. Nevertheless, there are no assurances that the historical demand of analogous products corresponds to the future demand of new products (Kahn, 2014) and this should be handled with care. The main distinction

can be made between diffusion models and approaches that directly estimate the characteristics of demand. Diffusion models estimate parameters of diffusion to generate a forecast. Other approaches employ statistical methods and machine learning techniques to estimate the demand based on analogous products. In the next subsections, we discuss both.

3.1.1 Diffusion models

Diffusion models estimate the growth rate of product demand by considering factors that influence consumers adopting the product. It assumes that products follow a certain pattern, which can be explained by parameters of the market, such as the potential market size. These parameters for a new product can be estimated by market research, but also with analogous products of which the parameters are known. The most dominant diffusion model for forecasting product sales of the last decades is the Bass model (Albers, 2004). Diffusion models are based on the diffusion of innovations theory discussed at length by Rogers (1962).

Bass model

The Bass model deserved quite some attention in the literature due to its simple structure and relatively high explanatory power (Lee et al., 2014), and different extensions of the model have been proposed (Bass, 2004). The model makes a pre-launch forecast of the demand during the product life cycle by estimating the diffusion.

Bass (1969) defines two groups of consumers: innovators and imitators. The innovators decide to adopt an innovation independently of decisions of other individuals in a social system. The imitators are influenced by the adoption of other individuals. The behaviour of these groups are specified by the coefficient of innovation (p), and the coefficient of imitation (q). Together with the potential market size (m), the model predicts the product sales over time. The most common way to estimate these parameters is to assume that its adoption will follow the pattern of comparable products (Thomas, 1985).

Examples with product characteristics

Comparable products can be determined by judgment, but also with statistical methods or machine learning techniques. For example, Lee, Boatwright and Kamakura (2003) developed a hierarchical Bayesian model, that used album characteristics to estimate the parameters of a diffusion model and generate a prelaunch sales forecast. With limited background characteristics, they could provide information about the potential demand of a music album. Goodwin, Dyussekeneva and Meeran (2013) used analogous products to estimate the parameters of the Bass model for new electronic products. Among others, they apply nearest neighbour and regression analysis to select comparable products. They concluded that the use of multiple products instead of one analogous product improved the forecast. Lee et al. (2014) applied several regression algorithms such as multiple linear regression, artificial neural networks and decision trees to estimate the parameters based on product features. In an illustrative example, they estimated the parameters for the annual demand of 3D TVs in the North American market. However, they only estimated the two coefficients, p and q, while they estimated the potential market size separately from market research. They did not estimate the potential market size by analogous products, since this size is likely to be affected by various environmental factors and marketing efforts rather than by product characteristics (Mahajan & Peterson, 1978).

Complications of diffusion models

Besides the various factors that influence the market size, there are some other complications with the Bass model and other diffusion models. To prevent biases in the estimates of the parameters, the demand data of analogies should include observations of the introduction of the product and observations when the demand reaches its peak. Due to this limitation, Goodwin et al. (2013) rejected products with less than 5 years of sales observations, because forecasts were likely to be unreliable under these conditions in a Bass model.

Additionally, the intention of Bass (1969) was focused on the life cycle prediction of infrequently purchased products and predicted the demand per year. Diffusion models are mainly used to model emerging technologies or new-to-the-world products at market level (Kahn, 2002), such as the adoption of electric cars in Europe. Using these models for predicting the demand during the introduction period of new SKUs might lead to significant errors (Baardman et al., 2018). Therefore, other new product forecasting methods are proposed in literature. These other methods do not use certain parameters, but directly forecast the demand based on comparable products. We discuss these methods in the next subsection.

3.1.2 Direct demand forecasting by analogous products

Since diffusion models are not applicable in all situations, other forecasting methods are developed. Instead of estimating parameters of a model, these methods derive demand characteristics directly from analogous products. Methods predict, among others, the total demand, the demand pattern, or predict a group of comparable products. Statistical methods, machine learning, and pattern recognition techniques are used to analyse comparable products.

The general approach is to match a new product with a number of existing products and use the historical data of these existing products as prediction for the new product. Since these methods consider some distinctive products, instead of all existing products, these models can produce more accurate forecasts (Lu & Kao, 2016). A simple approach to find similar products is to use a statistical method such as the nearest neighbour based on the product characteristics. The disadvantage of this method is that it finds analogous products solely on product characteristics and there is no guarantee that the demand is also similar. Additionally, all characteristics are weighted equally, while there may exist only a few significant predictive characteristics. Hence, human judgment is still required to identify characteristics and relevant analogous products (Goodwin et al., 2013). Therefore, several authors also applied other types of data-driven models for new product forecasting.

Proposed forecasting methods in literature

In 1999, Neelamegham and Chintagunta proposed a forecasting model that considered, among others, movie attributes such as genre, and the presence of movie stars as factors to predict total viewership of movies in the first week. The model showed proper predictions on movie-country level. A Bayesian approach was applied to provide a measure of uncertainty of the forecasts.

Thomassey and Fiordaliso (2006) proposed a forecasting system for mid-term forecasting in the apparel industry. First, demand profiles were distinguished from existing products by k-means clustering. Afterwards, they trained a decision tree with the C4.5 algorithm. This decision tree classified products to the profiles based on the price, the starting time of sales, and the life span of items. With this classification tree, Thomassey and Fiordaliso could predict the sales pattern of future items. The predicted profile, which is the mean pattern of the underlying existing products, was used as forecast for future items. Thomassey and Fiordaliso showed that the decision tree classification performed better than models such as the Naive Bayesian classifier

and the nearest neighbour classifier.

One year later, Thomassey and Happiette (2007) applied the same procedure with neural clustering and classification techniques. Demand profiles were derived by using a Self-Organising Map and k-means clustering. Predicting the demand profile was performed by a Probabilistic Neural Network. However, this model did not perform better than Naive Bayes classification.

An extreme learning machine was applied by Sun, Choi, Au and Yu (2008) to forecast the sales of fashion items, which outperformed other neural network-based methods. The relationship between the sales and product characteristics such as the color, size, and price were investigated.

Szozda (2010) proposed a forecasting method that compared the initial sales of a relatively new product to the initial sales of existing products. Hence, this method did not generate a pre-launch forecast, but used the initial sales as input. The time series were adjusted in order to find similar demand patterns with a different scale. The demand shape of the existing product with the highest similarity was used to adjust the sales forecast of the new product.

Another approach was proposed by Hibon et al. (2013). They considered both sales patterns as product features during clustering. New products are assigned to clusters based on the minimum Euclidean distance of the product features and initial orders. A forecast was generated by determining the average percentage change in sales for each time period of the group of existing products.

Tehrani and Ahrens (2016) forecasted sales combining classification and regression models. A probabilistic approach identified the class of products in terms of sale. Thereafter, kernel machines empowered with a probabilistic approach was used to predict the number of sales. The combined approach showed robust and promising results.

In 2017, Basallo-Triana et al. applied a fuzzy Gustafson-Kessel algorithm for clustering the time series of analogous products. Instead of generating a pre-launch forecast, Basallo-Triana et al. assigned new products to a cluster based on the initial sales. For each time period t, new products are assigned to a cluster. The forecasts were generated using multiple linear regression, support vector machines and neural networks.

Baardman et al. (2018) proposed a scalable algorithm that iteratively determined distinctive groups of similar existing products. New products were probabilistically assigned to these groups based on their product features with multinomial logistic regression. The forecast of the new product was the weighted average of the group forecast weighted by the group probabilities.

Loureiro et al. (2018) explored the use of multiple algorithms for the prediction of the total sales of new fashion items, based on product characteristics and the expectation level of the sales. With k-fold cross validation, multiple algorithms such as a deep neural network, Random Forest and support vector regression were analysed. The best MSE and RSME were achieved by deep neural networks, whereas the Random Forest obtained the best R^2, MAPE, and MAE. Although deep neural networks showed good potential, Loureiro et al. suggest that Random Forest is more suitable in practice, since it provides satisfactory predictive performance and the training process is less complex.

3.1.3 Conclusion on new product forecasting methods

Both diffusion models and other methods are applied to forecasting new product demand. Diffusion models are widely used and mainly focus on market predictions and emerging technologies. The other methods are more suitable for predicting demand at SKU level. These methods are more flexible for the application to individual companies. Several statistical methods and machine learning techniques were applied. To select a method that is suitable for our research, we will discuss the literature about the most common machine learning methods in the next

section.

3.2 Machine learning techniques

Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed. This field focuses on creating algorithms that can learn en make predictions based on data and feedback. By training algorithms with data, they can derive implicit rules and knowledge. This field seems to fit particularly well for finding implicit relations between product characteristics and the initial demand of new products.

We will investigate classification and regression algorithms. Classification algorithms can predict a class, such as a demand profile or a type, whereas regression algorithms can predict continuous values such as the demand of a product. Both algorithms are supervised methods, which means that the algorithms are trained with both input and output data. The algorithms learn the mapping from input to output (Alpaydin, 2010). In our case, the training set would consist of existing products, with the product characteristics as input and for example the demand as output. After a supervised algorithm is trained, it can be confronted with new data and predict the outcome.

In this section, we review the most common methods, which often can be used for both classification and regression. The methods are multiple linear regression, logistic regression, decision trees, Random Forests, gradient boosting, artificial neural networks, and support vector machines. These models will only be discussed on a high level. The exact mathematical derivations and solving approaches are left out of the scope of this literature review.

3.2.1 Multiple linear regression

Linear regression is a basic regression technique to predict a value for output Y with input parameters $X = (X_1, X_2, \ldots, X_N)$. Linear regression with more than one input parameter is called Multiple Linear Regression (MLR). The relationship between input parameters X and the output parameter Y can be written as follows:

$$Y = \beta_0 + \sum_{i=1}^{N} \beta_i X_i \tag{3.1}$$

where
Y is the target (dependent variable)
$X_1 \ldots X_N$ are the predictors (independent variables)
$\beta_0 \ldots \beta_N$ are the partial regression coefficients

The model either assumes that the regression function is linear, or that it is a reasonable approximation (Hastie, Tibshirani & Friedman, 2001). Input data for linear regression can be both numerical and categorical. Numerical features have one predictor, while with categorical data each individual level is considered as an input parameter. The most popular method to estimate the parameters β is least squares. Least squares determines the coefficients β such that the residual sum of squares is minimised (Hastie et al., 2001).

The advantage of this method is its simplicity. It is easy to interpret, since a partial regression coefficient is generated for each predictor and categorical level, which shows the direct impact on Y. However, linear regression can only approximate linear relationships between input and output parameters. Therefore, it is not suitable for modelling complex relations.

Interval prediction

A widely used method to generate prediction intervals for linear regression is proposed by Koenker and Bassett Jr (1978), which is called regression quantiles. Regression quantiles estimate percentiles of the underlying distribution and minimises the weighted absolute distance to the q^{th} regression line. Hence, for each quantile a new line is calculated.

3.2.2 Logistic regression

Although the name would indicate that logistic regression is a regression algorithm, it is used for classifying binary or multi-class variables (Delen, 2011). Logistic regression employs the techniques of linear regression to calculate the probability of belonging to a certain class. The linear predictor function is similar to the function of linear regression:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \beta_i X_i \tag{3.2}$$

Logistic regression transforms the output of this linear function with the logistic function, illustrated in Figure 3.1. The logistic function transforms any value from the linear function into a value between zero and one, which is the modelled probability to belong to a certain class:

$$p_i = \frac{1}{1 + e^{-f(x)}} \tag{3.3}$$

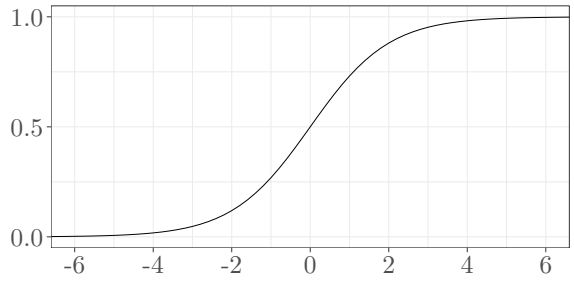


Figure 3.1: The standard logistic function

Classification with more than two classes is called multinomial logistic regression. Both binary logistic regression and multinomial logistic regression use maximum likelihood estimation to determine the coefficients (Starkweather & Moske, 2011). It can occur that the maximum likelihood estimation finds complex models that overfit the data. This means that the model is very accurate for predicting the trained data, but not very accurate on new data. To counter overfitting, usually a regularisation term is used to penalise complex models. Logistic regression can only handle numerical data and categorical data should be encoded to numerical values. The predictions of the logistic regression model can be tuned by adjusting the threshold level of the probability. The threshold is usually 0.50 with binary classification, where the prediction is class 0 if the probability is less than 0.50 and class 1 if the probability is greater than or equal to 0.50. The main advantages of logistic regression are the simplicity and interpretability of the algorithm. An disadvantage of the model is the linearity within the linear regression function (Delen, 2011).

3.2.3 Decision Trees

A Decision Tree (DT) is a widely used algorithm for both classification and regression, which is popular due to its simplicity. A DT tries to split a data set into homogeneous subsets by following a sequence of binary decisions. They are easily interpretable since they represent a set of simple rules.

To construct a DT, each split needs to be determined sequentially. At each split, we consider all available features and determine the best split among these features. The feature with the highest information gain is chosen. The information gain can be determined with an impurity measure, which is the residual sum of squares for regression trees (Breiman, Friedman, Olshen & Stone, 1984). The Gini index and the entropy are commonly used as impurity measures for classification. Both are measures for 'chaos' in the data. For each split of a DT, the data is split such that the impurity is minimised. The prediction is the mean value of observations in the leaf node (the node that is not further splitted). The size of a decision tree and the risk of overfitting can bounded by the maximum depth and the minimum number of samples in a leaf node.

The major advantage of decision trees is the human interpretability. The binary splits of a DT can be followed easily. Decision trees are also computationally simple and they can handle both numerical and categorical data, although categorical data may also be encoded to numerical values. Decision trees can also work with missing data or categories which do not exist in the training set by treating them as an extra category. However, the predictions of a DT are limited by the range of outputs in the training data and cannot extrapolate beyond this range. Additionally, decision trees are vulnerable to overfitting to details of the training data. A small change in the data can result to a large change in the final tree. To prevent overfitting, decision trees can be pruned. Pruning reduces the complexity of a tree by removing some terminal nodes. In practice, other algorithms generally achieve better performance (Hastie et al., 2001). To improve the performance of decision trees, alternative tree-based methods are proposed. We discuss gradient boosting and random forests in the next paragraphs after discussing interval predictions with decision trees.

Interval prediction

An implementation for prediction intervals for decision trees is proposed by Chaudhuri, Loh et al. (2002). With this method, p-values from linear quantile regression were used with the splitting criterion. A polynomial was fitted to the samples in each terminal node, such that quantiles could be determined from a piecewise polynomial model.

3.2.4 Random Forests

Random Forest (RF) is an ensemble of decision trees for both classification and regression, introduced by Breiman (2001). Random Forest grows a set of parallel decision trees. The final prediction of a RF is the majority vote of each tree (with classification) or the mean prediction of the individual trees (for regression). The idea of Random Forest is similar to asking a lot of individuals to make a prediction independently and then using the majority vote or average of the group of people as final prediction.

To grow a set of diverse trees from one data set, Breiman (2001) applied two sampling methods. The first method is bootstrapping, which selects random samples (bootstraps) from the training data to grow each tree (Breiman, 1996). The second method is the selection of random features at each split while growing the trees (Ho, 1995). Instead of considering all features at each split, a random selection of features is considered. From this random set of features,

the best split is chosen. The combination of these methods results in a highly independent set of decision trees. This prevents overfitting, since the predictions of these diverse trees are averaged to obtain a final prediction. Additionally, the predictions of RF generally result in high accuracies compared to single decision tree predictions (Fawagreh, Gaber & Elyan, 2014).

Random Forest has two main parameters, the number of trees and the number of features that are randomly selected at each split (denoted by *mtry*). The number of trees should be sufficiently high and usually 500 trees are enough to stabilise the performance (Oshiro, Perez & Baranauskas, 2012). When the number of trees is sufficiently high, the *mtry* has the most influence on the performance of RF (Probst, Wright & Boulesteix, 2019). The lower the value of *mtry*, the lower the correlation between the individual trees, which results in better stability of the algorithm. A lower *mtry* can also exploit features with moderate effects, which are otherwise masked by features with stronger effects. However, too low values of *mtry* can decrease the performance of RF, since trees may only be built on insignificant features, selected out of a small number of randomly chosen candidate features. The typically chosen value for *mtry* is the square root of the number of features for classification problems and the number of features divided by 3 for regression problems. Nevertheless, the results are generally close to the best result over a wide range of values for *mtry* (Meinshausen, 2006).

RF is widely used due to its efficiency, and robustness to outliers and noise (Breiman, 2001; Fawagreh et al., 2014). It overcomes overfitting (a disadvantage of Decision Tree) and obtains substantially better results due to the bootstrapping and the random selection of features. Another major advantage of RF is that it is fairly simple to apply. There are only two main parameters, which are relatively simple to tune and provide proper results over a wide range of values. Additionally, Random Forests can be applied to various types of data sets, since it can handle continuous and categorical data, as well as handling missing values (Ali, Khan, Ahmad & Maqsood, 2012). Random Forest can obtain high accuracies with a relatively small number of samples, but more complex methods such as Artificial Neural Network and Support Vector Machines may obtain higher accuracies when tuned properly, especially when the number of samples increases. Since each tree uses a selection of samples, there are also samples which are not considered in a tree, the Out-Of-Bag (OOB) samples. These samples can be used to evaluate the performance of Random Forest without the need of a separate validation set or cross validation. The availability of OOB samples is a significant computational advantage of Random Forest.

Compared to Decision Trees, Random Forests are not easily understood with hundreds of trees. Therefore it is often treated as a "black-box" model. Nevertheless, RF comes with two interesting components that provide insight into the model: feature importance and proximity. The importance is a measure of how much individual features contribute to the overall performance. The feature importance can be determined by sequentially permuting each feature and calculating the decrease of the performance. With a large decrease, the feature contributes significantly to the overall performance and is regarded as important. With a low decrease in the performance, a feature is considered less important. The feature importance can be calculated by permuting OOB samples and re-run these samples to an already trained RF algorithm. Additional to the feature importance, RF comes with the proximity. The proximity is a measure of comparability between samples. The proximity between two samples is the percentage of trees in which they end up in the same leaf node. When two samples end up in the same leaf node for a significant number of the trees, they are presumably very similar. However, when two samples never end up in the same leaf nodes, they are probably unrelated to each other. Furthermore, the proximity can be used for outlier detection, clustering, and missing data imputation (Breiman, 2003).

Interval prediction

To predict intervals, Meinshausen (2006) introduced Quantile Regression Forest (QRF), which gives a non-linear and non-parametric way of estimating conditional quantiles. QRF grows a set of trees in the same way as Random Forest. However, instead of keeping only the mean of the observations, QRF keeps all observations for each leaf node in each tree. Hence, QRF approximates the full conditional distribution and can provide a prediction interval for new observations. Since Quantile Regression Forest is only a small modification of Random Forest (RF), it comes with comparable computational costs.

Quantile Regression Forest has recently been successfully applied in quantifying uncertainties in other fields, such as photovoltaic electricity production (Zamo, Mestre, Arbogast & Pannekoucke, 2014), weather forecasting (Taillardat, Mestre, Zamo & Naveau, 2016), soil mapping (Vaysse & Lagacherie, 2017), and wind power forecasting (Lahouar & Slama, 2017).

3.2.5 Gradient Boosting

Gradient Boosting (GB) is an ensemble of decision trees introduced by Friedman (2002). Instead of parallel trees like Random Forest, Gradient Boosting builds trees sequentially. Every next tree is trained on the residuals of the previous trees. The residuals are the differences between the actual value and the value predicted by the trees. Each time a tree is added, it should learn from the previous tree and decrease the residuals, which should lead to a better prediction. This tree-based method can be used for both classification and regression.

Gradient Boosting generally produces accurate predictions, mainly due to the reduction of residual values by additional trees. Additionally, since it consists of decision trees, it can handle both numerical and categorical data as well as missing values. Compared to Random Forest it may obtain a higher performance when properly tuned, but it is less robust to noise and it is prone to overfit the training data. Overfitting can be prevented by adding a regularisation term and tuning the control parameters of the algorithm, such as the learning rate, the number of trees or the maximum depth of each tree (Pedro, Coimbra, David & Lauret, 2018). The choice for these parameters depends on the problem instance. To obtain proper results, the algorithms should be tuned on these parameters for each specific problem. Like Random Forest, Gradient Boosting is also not understood easily with multiple sequential trees. Therefore GB is also often treated as a "black-box" model. Similar to RF, it is possible to determine the feature importance. However, GB has no OOB samples. Therefore, a separate validation set is required to evaluate the feature importance.

Interval prediction

It is possible to calculate prediction intervals or quantiles with gradient boosting. However, a Gradient Boosting algorithm can only be trained for a single quantile (Nagy, Barta, Kazi, Borbély & Simon, 2016). Therefore, two models need to be trained to determine a prediction interval (e.g., the 5^{th} and 95^{th} quantile for a 90% prediction interval). When multiple intervals are required, the computational costs of Gradient Boosting will be considerably high (Nagy et al., 2016).

3.2.6 Artificial Neural Networks

An Artificial Neural Network (ANN) is inspired by biological neural systems, such as the brain, and can be applied to classification and regression problems. An ANN consists of several connected neurons (also called “nodes”). Each neuron receives input signals, processes these signals

and produces an output signal (Zhang, 2009). These neurons are organised in three types of layers (see Figure 3.2). The input layer is the first layer and receives the input data. The second type are the interior layers, called hidden layers, which receive input signals from the previous layer and sends the output signal to the next layer. The third type is the output layer, which receives signals from the previous layer and produces the output of the neural network.

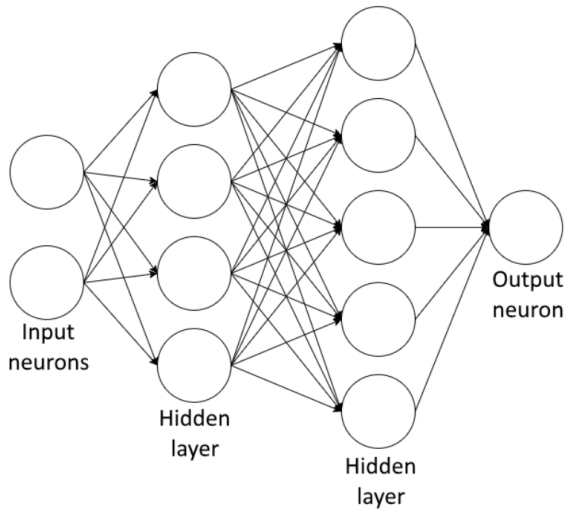


Figure 3.2: Schematic overview of a Artificial Neural Network

Each individual neuron receives input signals. Processing these signals consists of three steps. First, the input signals are multiplied by their corresponding weights. Then these values are all summed. In the last step, the summed value is transformed by an activation function. This can be any function, for which usually a logistic, hyperbolic tangent or rectified linear unit is applied (Schmidhuber, 2015). Since the data is transformed by the activation functions, ANNs cannot deal with categorical data. To use categorical data, it should first be encoded to numerical data.

Initially, the weights in an ANN are chosen at random. Thereafter, training samples are presented to the network, which result in certain predictions. The error between the predictions and the actual values are determined by a loss function, which is usually the residual sum of squares with regression. Then, the first order derivative of the error with respect to the weights is determined. With this derivative, the weights are updated to minimise the error. This process is repeated until the reduction of the residual sum of squares drops below a certain threshold.

The main advantage of Artificial Neural Networks is that they are extremely flexible. Due to multiple layers of non-linear combinations, it is possible to map complex relations of the data. To prevent overfitting of the flexible ANNs, a regularisation term is added to the objective function. When presented with a sufficient amount of data, ANN can produce very accurate predictions. This bring us also to the limitations of an ANN. They need a large amount of data in order to obtain their full potential. The complexity of an ANN can also bring long computational times and it needs to be tuned sufficient (Shrivastava, Khosravi & Panigrahi, 2015). Additionally, the results of a ANN are quite difficult to explain, since the network of hidden layers makes it a "black-box" model. To provide some insight into an ANN, multiple methods are developed to show which features are important for the prediction of an ANN (Olden, Joy & Death, 2004). These methods analyse the weights in the network or analyse the change in the output by sequentially removing input features from the network.

Interval prediction

In literature, several techniques have been proposed for the construction of prediction intervals for neural networks, such as the Delta, Bayesian, mean-variance estimation, bootstrap, and Lower Upper Bound Estimation technique (Khosravi, Nahavandi, Creighton & Atiya, 2010, 2011). However, the Delta, Bayesian and mean-variance estimation models have limited applicability. The Delta technique is based on the assumption that a non-linear ANN can be linearised using Taylor's series expansion (Hwang & Ding, 1997). The Bayesian method and mean-variance estimation require appropriate distributions, such as Normally distributed data. Additionally, the Bayesian and bootstrap method may be limited by the large computational burden (Shrivastava et al., 2015). The Lower Upper Bound Estimation is a more recent technique, which constructs a ANN with two outputs for estimating the prediction interval bounds. By training, the ANN minimises a specific objective function, which includes the interval width and coverage probability. The Lower Upper Bound Estimation technique is less computational expensive and does not have limiting assumptions about the data. However, it will only output one prediction interval. Multiple models are required for multiple prediction intervals and point-estimates.

3.2.7 Support Vector Machine

A Support Vector Machine (SVM) is an algorithm developed at the AT&T Bell Laboratories by Vapnik and his co-workers for classification problems (Cortes & Vapnik, 1995). Later on it was extended to regression problems as well, which is called Support Vector Regression.

With a Support Vector Machine, a hyperplane is constructed in a high dimensional space. In case of a classification problem, it is constructed such that the hyperplane linearly separates the classes. A regularisation term is often used to allow small and prevent large deviations from the hyperplane. In case of a regression problem, it is constructed such that the hyperplane fits the training samples (Vapnik, 1999). A hyperplane is a plane whose dimension is one less than its ambient space. A hyperplane is linear, but the training samples are often not linear. To fit samples that are non-linear, the so called 'kernel trick' was developed. With the kernel trick, samples are transformed to a higher dimensional space to achieve a better linear separation or regression of the hyperplane. Each transformation is called a kernel. Examples of kernels are polynomial or hyperbolic transformations. An example of a transformation from a 2 dimensional space to a linearly separable 3 dimensional space is illustrated in Figure 3.3.

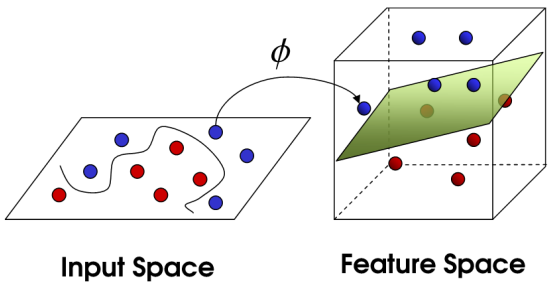


Figure 3.3: A hyperplane can separate the classes due to the kernel trick

The main advantages of a Support Vector Machine is that it can approximate non-linear functions accurately and avoids overfitting due to regularisation and the kernel functions (Zhao, Dong, Xu & Wong, 2008). SVM can be easily explained graphically with lower dimensions (see Figure 3.4). However, with more than 3 dimensions SVM become difficult to explain. SVM may be computationally more efficient than ANNs, but they can still suffer from extensive memory

requirements due to the high dimensional mapping of large data sets (Suykens, 2003). The SVM can be tuned by choosing the right kernel and the right kernel parameters. Tuning is important for its performance and can be an extensive task. Since SVM apply mathematical transformations, they cannot handle categorical data. To use categorical data, it first needs to be encoded into numerical data. This brings another complexity to the application of SVMs.

Interval prediction

Several methods are proposed to provide prediction intervals with SVM (Hwang, Hong & Seok, 2006; Zhao et al., 2008; De Brabanter, De Brabanter, Suykens & De Moor, 2010; Shrivastava et al., 2015). These methods are very efficient and accurate for certain problems and can produce better results than QRFs or ANNs. These methods also suffer from similar disadvantages as Artificial Neural Networks, such as assumptions about data distributions and the requirement of a full model for each quantile.

3.2.8 Summarising table

In the previous sections, we discussed multiple statistical and machine learning techniques. To provide an overview of these models, we summarised the characteristics of the models in Table 3.1.

3.2.9 Conclusion on machine learning techniques

In this section, we discussed multiple statistical and machine learning techniques. All techniques have their own specific aspects. A model should fit the requirements of a specific problem. Nevertheless, sometimes categorical encoding is necessary to fit the data to a model. Besides the fit of a model, there is generally a trade-off between the complexity and the accuracy of a model. To obtain a higher accuracy, usually a more complex model is required. More complex models have more computational effort and generally require more specific tuning. Additionally, complex models are "black box" models and results are harder to explain. To explain the results of a model is important in our case, since planners or managers may want insights into the model before they base decisions upon the predictions.

Considering the problem characteristics and the trade-off between complexity and accuracy, the Random Forest algorithm seems the most suitable. RF can handle both classification and regression problems, can model non-linear relations, averages bootstrapped predictions for a

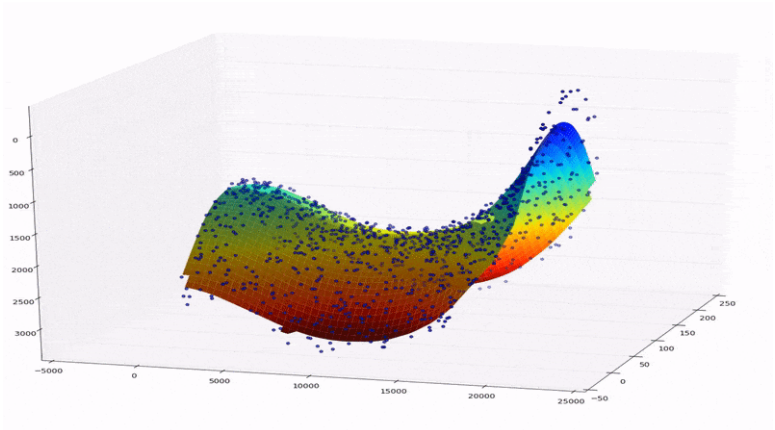


Figure 3.4: Graphical visualisation of a SVM

Algorithm[1]	MLR	LR	DT	RF	GB	ANN	SVM
Type	Regression	Classification	Regression & classification	Regression & classification	Regression & classification	Regression & classification	Regression & classification
Tuning effort[2]	++++	+++	+++	+++	+	+	+
Performance[2]	++	++	++	+++	++++	++++	++++
Categorical data	Individual coefficients / Categorical encoding	Categorical encoding	Tree-based / Categorical encoding	Tree-based / Categorical encoding	Tree-based / Categorical encoding	Categorical encoding	Categorical encoding
Relations	Linear	Classification	Non-linear	Non-linear	Non-linear	Non-linear	Non-linear
Data required[2]	++++	++++	++++	+++	++	+	+++
Overfitting	Models only linear behaviour	Regularisation	Pruning	Included by bootstrapping and random feature selection	Tuning & regularisation	Regularisation	Regularisation & kernel functions
Interpretability[2]	++++	++++	++++	+++	++	+	++
Insightful functions	Coefficients as feature importance	Class probabilities	Visual representation of the tree	OOB feature importance and proximity	Feature importance with a validation set	Feature importance with a validation set	Feature importance with a validation set
Computational complexity[2]	++++	++++	++++	+++	+++	++++	+++
Interval prediction	One model per quantile	-	Piecewise polynomial model	Full conditional distribution	One model per quantile	One model per quantile	One model per quantile

Table 3.1: Overview of statistical techniques and machine learning algorithms
[1] MLR = Multiple Linear Regression, LR = Logistic Regression, DT = Decision Trees, RF = Random Forests, GB = Gradient Boosting, ANN = Artificial Neural Networks, SVM = Support Vector Machine
[2] Least favourable: +, most favourable: ++++

better generalisation and can therefore be more accurate and versatile than multiple linear regression, logistic regression and decision trees. Nevertheless, it is still rather simple to tune with the *mtry* at major parameter. This is favourable, since it makes the implementation at multiple companies easier, compared to Gradient Boosting, Artificial Neural Network, and Support Vector Regression. Like these algorithms, RF is usually treated as a "black-box" model. Nevertheless, the possibilities for variable importance and proximity can provide insights into the model. Besides, with RF there is no further encoding of the categorical input necessary due to its tree-based nature. Another reason why RF fits the research problem, it that it does not need a large number of samples to obtain reasonable results. This favourable, since not all companies introduce thousands of products. Lastly, prediction intervals can be generated using Meinshausens Quantile Regression Forest.

3.3 Performance indicators for machine learning and forecasting

The objective of this research is to develop certain methods to create more insights into the demand of new products. To analyse and validate the performance of these methods, we require performance indicators. In this section, we elaborate about several performance indicators to

assess the performance of machine learning algorithms and forecasts and we conclude which indicators we use as Key Performance Indicators (KPIs).

3.3.1 Classification performance

The performance of a classification algorithm is based on the number of correct predictions. The predictions of a classification algorithm can be summarised in a so-called confusion matrix. The general form of a confusion matrix with a 2-class classification problem is shown in Table 3.2.

		Predicted	
		Class = 0	Class = 1
Actual	Class = 0	x_{00}	x_{01}
	Class = 1	x_{10}	x_{11}

Table 3.2: Confusion matrix of a two-class problem

In the table, x_{ij} denotes the number of instances of class i that are predicted to be of class j. The instances are correctly classified when $i = j$. If i and j are not equal, the instance is misclassified. The accuracy of an algorithm can easily be determined from this matrix. The accuracy indicates the fraction of the predictions that is correctly classified:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{x_{00} + x_{11}}{x_{00} + x_{01} + x_{10} + x_{11}} \tag{3.4}$$

Besides the accuracy, Cohen (1960) suggested the kappa as performance indicator. The kappa adjusts the observed accuracy for the expected accuracy (random guessing). This metric is useful for imbalanced classes. For example, an accuracy of 0.80 is easier to achieve when the ratio between classes is 75:25 than when the ratio is 50:50. The kappa metric tries to take away this bias. The kappa is defined as:

$$kappa = \frac{Accuracy_o - Accuracy_e}{1 - Accuracy_e} \tag{3.5}$$

where
$Accuracy_o$ is the observed accuracy (Eq. 3.4)
$Accuracy_e$ is the expected accuracy: $Accuracy_e = \frac{(x_{00}+x_{01})\cdot(x_{00}+x_{10})+(x_{10}+x_{11})\cdot(x_{01}+x_{11})}{(x_{00}+x_{01}+x_{10}+x_{11})\cdot(x_{00}+x_{01}+x_{10}+x_{11})}$

With the kappa, a score of 0 equals the performance of random guessing. A kappa score of 1 equals a perfect prediction. Since the predictions of a algorithm can also be worse than random guessing, a negative value for the kappa is also possible.

3.3.2 Regression and forecasting performance

There are several metrics to analyse the accuracy of regression models and forecasts. Let Y_t denote the actual value of instance t and F_t the prediction or forecast of Y_t. Commonly used performance indicators for the error are: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Average Percentage Error (MAPE), and Symmetric Mean Average Percentage Error (sMAPE) (Hyndman & Athanasopoulos, 2018):

$$MAE = \frac{1}{n}\sum_{t}^{n} |Y_t - F_t| \tag{3.6}$$

$$MSE = \frac{\sum_t^n (Y_t - F_t)^2}{n} \tag{3.7}$$

$$RMSE = \sqrt{\frac{\sum_t^n (Y_t - F_t)^2}{n}} \tag{3.8}$$

$$MAPE = 100 \cdot \frac{1}{n} \sum_t^n \frac{|Y_t - F_t|}{Y_t} \tag{3.9}$$

$$sMAPE = 200 \cdot \frac{1}{n} \sum_t^n \frac{|Y_t - F_t|}{Y_t + F_t} \tag{3.10}$$

The MAE, MSE, and the RMSE are scale-dependent metrics, so the scale depends on the scale of the data. These metrics are useful when comparing different methods on the same data set, but are not effective when comparing data sets that have different scales. The MAE is less sensitive to outliers than the MSE and the RMSE, since the MSE and RMSE squares the errors. The advantage of the RMSE compared to the MSE is that the scale of the errors is similar to the scale of the actual values. Therefore, the RMSE is easier to interpret. The MAPE is similar to the MAE, except that it is expressed in percentage terms. Therefore, it is not scale-dependent and can be used to compare data sets that have different scales. However, the disadvantage of the MAPE is that it is undefined if $Y_t = 0$ and extremely high if Y_t is close to zero. Additionally, the MAPE puts a heavier penalty on the positive errors compared to negative errors. The sMAPE adjusts this heavier penalty by including the forecast F_t in the denominator. This also partly solves the problem when the denominator is zero or close to zero. It is only zero when both values are zero and the error is zero. Nevertheless, when one of the values is close to zero, it still results in high percentage errors. Therefore, the MAPE and sMAPE should not be used when the data is close to zero (Hyndman et al., 2006).

3.3.3 Interval performance

In literature, multiple performance indicators have been proposed for the evaluation of prediction intervals (Kabaila, 1999; Gneiting & Raftery, 2007; Khosravi, Nahavandi & Creighton, 2010). For the interval performance, two important aspects should be covered: the coverage probability and the interval width.

The first indicator is the Prediction Interval Coverage Probability (PICP) (Khosravi, Nahavandi & Creighton, 2010):

$$PICP = \frac{1}{n} \sum_{t=1}^{n} c_t \tag{3.11}$$

where
$c_t = 1$ if $y_t \in \left[L_t, U_t\right]$ otherwise $c_t = 0$

L_t and U_t are respectively the lower and upper bounds of the t^{th} prediction interval. Theoretically, the PICP should be equal to the suggested probability. In reality, this may not happen due to the presence of noise and effects of uncertainty (Khosravi, Nahavandi & Creighton, 2010). If the PICP significantly deviates from the suggested probability, the prediction interval is not reliable.

The PICP can be increased by widening the interval width. However, a wide interval width provides less information. Therefore, it is also important to evaluate the width. The width of

the prediction intervals can be measured with the Prediction Interval Normalised Average Width (PINAW) (Khosravi, Nahavandi & Creighton, 2013):

$$PINAW = \frac{1}{nR}\sum_{t=1}^{n}(U_t - L_t) \tag{3.12}$$

where R is the range of the actual values. The PINAW is the average width of the prediction intervals as a percentage of the actual value range. The lower the PINAW, the smaller and more valuable the intervals will be.

3.3.4 Inventory management performance

The objective of this research is not to only evaluate the forecasting performance, but also the impact on inventory management. Common metrics to indicate the performance towards customers are the Cycle Service Level (CSL) and the Fill Rate (FR):

$$CSL = \frac{1}{n}\sum_{t=1}^{n}\mathbb{I}(Y_t \leq F_t) \tag{3.13}$$

$$FR = 1 - \frac{1}{n}\sum_{t=1}^{n}\frac{(Y_t - F_t)^+}{Y_t} \tag{3.14}$$

The Cycle Service Level is the probability of not having a stock out. The Fill Rate is the fraction of the demand that is met without backorders or lost sales (Silver, Pyke & Thomas, 2017). Both the cycle service level en the fill rate can simply be maximised to 100% by increasing the inventory levels. However, this increase in inventory levels results in possible excess stocks and therefore increases the inventory costs. Therefore, it is also valuable to a evaluate the inventory levels and inventory costs.

3.3.5 Conclusion on performance metrics

Several performance indicators are discussed in this section. First, we elaborated about classification indicators. The classification matrix provides an overview of all predictions and actual classes. Two performance indicators that can be derived from this matrix are the accuracy and kappa. The accuracy shows the percentage of correct predictions. The kappa is an accuracy adjusted for imbalanced classes, where 0 is equal to random guessing and 1 is a perfect prediction. To provide a proper overview of classifications, we will use both the accuracy and the kappa as KPIs for classification.

Besides classification indicators, we discussed regression and forecasting indicators. As KPI for regression problems, we will use the RMSE. The RMSE puts a heavier penalty on large errors than the MAE and is easier to interpret than the MSE since it has the same unit as the actual values. The scale-independent indicators MAPE and the sMAPE are not suitable for our research, since these should not be used when data is close to zero.

For the performance of prediction intervals, we described two indicators. These cover the two most important aspects of a prediction interval: the coverage probability and the interval width. Therefore, we will use both the PICP and the PINAW as KPIs.

Lastly, we described two metrics for inventory management: the Cycle Service Level and the Fill Rate. The CSL is the probability of not having a stock out, whereas the FR is the fraction of the demand that is met. Both indicators are ideally as high as possible, but this can also

increase inventory costs. Hence, besides the KPIs, we also need to evaluate the costs. These costs will be defined in the next chapter, where we describe the methodology.

4 Proposed method and experimental design

In this chapter, we propose our method demandForest, that provide a solution to the research objective and we describe with the experimental design how to validate the performance of this method. In Section 4.1, we propose the demandForest method that generates a forecast for the complete introduction period. Thereafter, we introduce an extension to the demandForest in Section 4.2. To compare the proposed methods with the current situation, we describe two benchmark methods in Section 4.3. At last, we elaborate about the experimental design in Section 4.4. In this section, we discuss how we train the applied machine learning algorithms and analyse the performance of the forecasts for the different data sets. We also define experiments to evaluate the inventory performance when the forecasts are employed. Additionally, we propose a synthetic data set for a theoretical evaluation of the methods.

4.1 demandForest

In the previous chapter, we found that the Random Forest and Quantile Regression Forest algorithms seem to be the most suitable for the current research problem. Both algorithms are used in demandForest to generate a forecast for the demand of a new product. Due to these machine learning algorithms, demandForest can learn to generate predictions for products of specific companies. The forecast for the complete introduction period provides insight into the amount and development of the demand of a new product. To provide these insights, demandForest divides the demand during 18 weeks into a demand profile and the total amount of demand. The cumulative demand patterns are clustered in distinctive profiles and predict this profile and the total demand. With the QRF algorithm, not only the total demand, but also the corresponding quantiles can be predicted. The combination of the profile and the total demand is the forecast for 18 weeks. This method is inspired by Thomassey and Fiordaliso (2006) and Loureiro et al. (2018), described in Section 3.1.2. We combine the profile predictions of Thomassey and Fiordaliso (2006) with satisfactory results and suitability of Random Forests as in Loureiro et al. (2018). Furthermore, we enhance the methodology by predicting quantiles with Quantile Regression Forests. The advantage of demandForest is that we only require two predictions (profile and demand) to obtain a forecast with 18 demand points. These two predictions can also be easily interpreted by planners, which may not understand the complete method and algorithms in depth. The method is illustrated schematically in Figure 4.1 and we discuss the utilisation of demandForest in more detail.

As visualised in Figure 4.1, demandForest consists of a preparation phase and an operational phase. In the preparation phase, historical demand is clustered into profiles with a k-means algorithm and the existing products are used to train the Random Forest and Quantile Regression Forest algorithms. In the operational phase, the profiles and algorithms are used to predict the demand of new products.

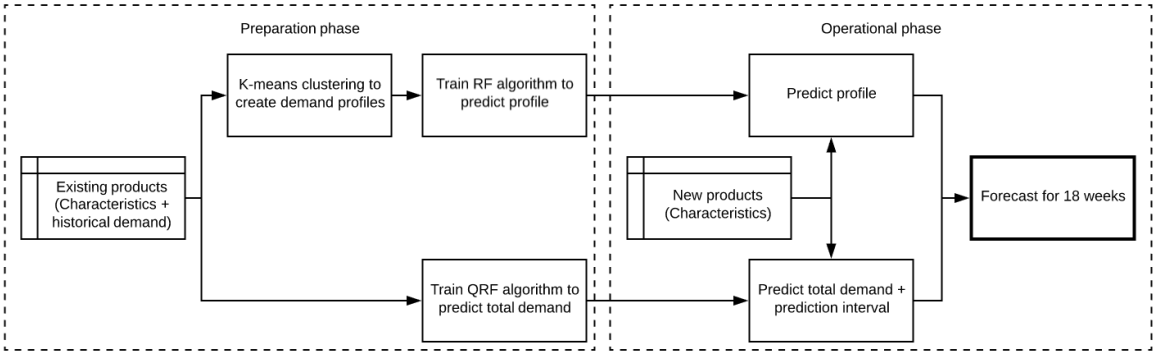


Figure 4.1: Schematic overview of demandForest

In the preparation phase, we first cluster the demand patterns to obtain demand profiles. We cluster the normalised cumulative demand of historical items with the k-means algorithm. The algorithm is repeated 25 times to obtain a stable result, as discussed in Section 4.4.2. As also described in Section 4.4.2, the number of clusters are determined by the CH-index, which resulted in two clusters (e.g., the demand profiles) for all the data sets available for this research. Subsequently, the profiles are assigned to each item. After clustering, a Random Forest algorithm is trained to classify the profile based on the product characteristics.

Besides the prediction of the profile, the total demand during the introduction period is predicted. For this prediction, we use the Quantile Regression Forest algorithm. This algorithm does not only predict the expected demand (the mean), but also a full conditional distribution. This full conditional distribution provides insight into the potential uncertainty of the demand of a new product. The QRF algorithm is trained based on the product characteristics as well.

After the preparation phase, the trained algorithms can be used for predicting the demand and profile for new products. The final forecast for the complete introduction period can be obtained by combining the prediction of the profile and the total demand. Since it is not possible to order or sell half a product, the forecast in each week is rounded towards the closest integer. When new data becomes available from recently introduced products, the algorithms could be trained again. In that case, it will consider the new data, which statistically improves the accuracy of the predictions in the future. Besides predicting the demand, safety stock levels can be determined using the conditional distribution. For example, using quantile 0.9 is equal to a CSL of 90%, which implies a probability of not having a stock out of 90%. This only holds for the prediction of 18 weeks. What the CSL becomes when used in combination with a predicted profile should be investigated.

4.2 Extended demandForest

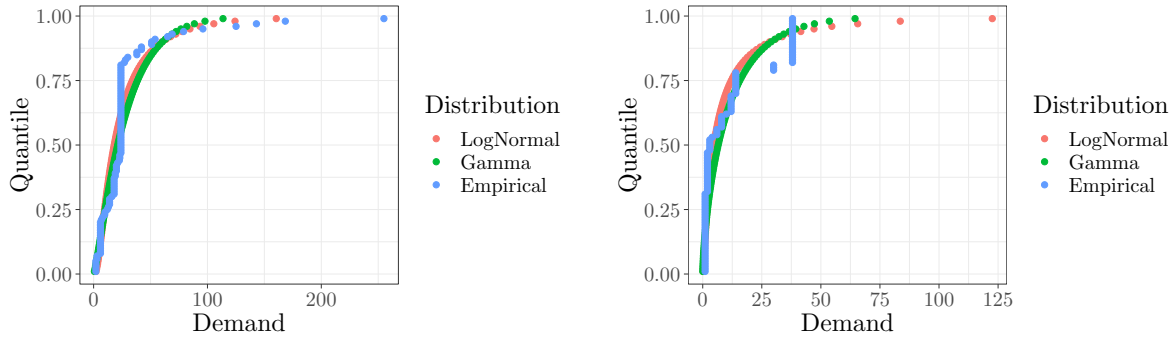
Due to a limited number of comparable products, it might be possible that the predicted empirical distribution of a new product only consists of a limited number of values. In that case, quantile 0.7 might have the same value as quantile 0.8. To overcome this limitation of the empirical distributions of the QRF algorithm, we extend this algorithm and fit a theoretical distribution over the empirical distribution. The empirical distributions might be limited by the number of existing products and might result in significant differences between the empirical and theoretical distribution. This causes that the null hypothesis of goodness-of-fit tests is rejected. However, the theoretical distribution would fit better than the empirical distribution from a limited number of observations.

Since the demand data in all data sets are right-skewed and non-negative, we will fit Log-

Normal distributions and Gamma distributions to the distributions generated by the QRF algorithm. Examples where the Log-Normal distribution is used for the demand in an inventory model are Cobb, Rumı and Salmerón (2013) and Gholami and Mirzazadeh (2018). The Gamma distribution is also often used for the distribution of demand within inventory control literature (Namit & Chen, 1999; Ramaekers, Janssens et al., 2008; Nenes, Panagiotidou & Tagaras, 2010).

The Log-Normal distribution is right-skewed and ranges from 0 to ∞, similar to the Gamma distribution. To calculate the parameters of the Log-Normal distribution, one takes the natural logarithm of the data and determine the mean and standard deviation. For the Gamma distribution, we apply the maximum likelihood estimation to estimate the parameters (Venables & Ripley, 2013). In our case, we use quantiles from 0.01 to 0.99 from the empirical distribution with a step size of 0.01 as data and fit the Log-Normal and Gamma distribution to this data for each prediction.

To provide examples for fitting these distributions, we already trained the QRF algorithm for the data of company B and E. From these results, we randomly picked the empirical distributions of two products. Figure 4.2 shows the cumulative distributions of two example products. The Log-Normal distribution and the Gamma distribution are fitted on the empirical distributions of these products. For the product of company B, Figure 4.2a, we observe the same demand of 24 for the quantiles between 0.47 and 0.81. In Figure 4.2b, we observe a similar effect at the product of company E for the quantiles of 0.82 to 0.99 of the empirical distribution, corresponding to a demand of 38. From the fitted distributions, we can observe that the Log-Normal and Gamma distribution fit the empirical distribution quite well. We assume that these theoretical distributions more accurate in the aforementioned ranges where the empirical distribution is constant at respectively 24 and 38. The underlying assumption is that the actual demand distribution is smooth, and the parametric distributions can represent this. However, we do not have the data to check this. Nevertheless, we will investigate out if these distributions improve the forecasts and inventory performance.



(a) Distribution of a product from company B

(b) Distributions of a product from company E

Figure 4.2: The cumulative distributions of two example products

4.3 Defining benchmark methods

To compare the proposed methods with the current situation, we ideally use the actual historical manual forecasts of planners and their orders. In that case, we can find out whether the proposed methods improve the current way of working. Unfortunately, this data is not available. Therefore, we need to define other types of benchmarks to compare with our proposed methods.

The first benchmark we define is Zero Rule, abbreviated to ZeroR, which simple predicts the average output of the training data for the test data (Amasyali & Ersoy, 2009). In our case, it uses the average demand of each week of the training data as weekly forecast for each new

product, or the average of the complete introduction period as forecast for the total demand of each product. For prediction intervals or safety stock calculations, we extend ZeroR by using the quantiles of each week from the training data, or by using the quantiles the complete introduction period.

ZeroR is a simple benchmark, as it does not make any distinction between the products and also does not relate to the current situation of forecasting. Therefore, we also want to use a method that can imitate the decisions of a planner. As described in Section 1.2, planners usually discuss the forecasts during S&OP meetings with managers and they often base their estimates on one similar existing product, of which they expect a similar demand. A discussion during S&OP meetings is difficult to imitate, but finding similar existing products should be possible.

A simple approach to find a similar product is to determine the 'nearest neighbour' based on the Euclidean distance of the product characteristics. A constraint for using an Euclidean distance is that all characteristics should be numerical. Moreover, all characteristics will be weighted equally. However, a planner may know that there are only a few significant predictive characteristics. Therefore, the nearest neighbour is in our case not suitable to mimic the decisions of a planner.

Another method of similarity among products, is using the proximity measure of the Random Forest algorithm. As mentioned in Section 3.2.4, the proximity between two products is the percentage of trees in which they end up in the same leaf node. With the proximity measure, we can identify an existing product with the highest similarity compared to the new product. A Random Forest algorithm weighs the different features to find the best value for the prediction. In other words, the product characteristics are weighted to find products with similar demand. This may resemble the domain knowledge or experience of a planner. Hence, we should be able to use the demand of the closest existing product as a benchmark for the actual prediction of the Random Forest.

The proximity may overestimate the accuracy of the actual behaviour of a planner. The proximity of a Random Forest considers all products introduced in the past, while a planner may only remember a limited amount. Additionally, a planner does not always use the historical demand of a comparable product as initial forecast. Nevertheless, this method comes quite close to the actual behaviour of a planner and is more suitable than the nearest neighbour. Moreover, when the actual RF algorithm improves the prediction of a most similar product based on proximity, it will probably also improve the actual forecasts of a planner.

As mentioned in Section 1.2, the software of Slimstock currently uses, by default, a coefficient of variation of 0.45 and a Normal distribution for the monthly demand of new products, when a planner manually sets up a forecast. After four months, all forecasting parameters get updated regularly based on the demand data. The forecast and coefficient of variation are used for safety stock calculations and order levels. Therefore, we will also use the factor of 0.45 for our benchmark method. In our case, we do not forecast the demand of one month, but the demand of four months. Hence, the coefficient of variation should be scaled to four months. Scaling the standard deviation or the coefficient of variation can be done by multiplying the value by the square root of the number of periods. Since we want to scale the factor from one to four months, the new coefficient of variation becomes: $0.45 \cdot \sqrt{4} = 0.9$. With this value and the Normal distribution, we can not only determine intimations of forecasts created by supply chain planners, but also prediction intervals, quantiles and safety stocks.

While the proximity is very useful for the prediction of the demand, it may not be very useful for the profile. Currently, planners do not have defined profiles available and may plan a stable demand for all weeks. Nevertheless, we calculate the average profile of all products in the training set. We can use this average profile as benchmark for the predicted profile. When the classification of the profiles does not improve the average profile, it would be easier for a

company to only apply the average profile to their new products instead of predicting profiles.

To compare the demandForest method, the extensions with the Log-Normal and Gamma distribution, and the benchmark methods to see which performs best, we will test the quality of the methods in the next chapter. First, in the next section, we explain the experimental design of this analysis. The experimental design outlines the steps taken to compare the methods in a structured and comprehensive way.

4.4 Experimental design

To analyse and validate the quality of the proposed method, when applied to the different data sets, we will perform several experiments. For these experiments, we go through several steps. The first step is the training phase. This phase resembles the preparation phases discussed in Sections 4.1. In this phase, we will analyse the algorithms to find the best setting for *mtry* to make predictions, and we analyse the feature importance to provide insight into which characteristics influence the demand of new products. Second, we go through the testing phase, which resembles the operational phase. In this phase, we analyse the actual quality of the forecasting methods. Third, we take it one step further and analyse the inventory performance when the forecasts are employed. We discuss these steps in further detail in the following subsections. Additionally, we propose a synthetic data set that can be used as theoretical evaluation of the methods and used for future work.

To train and analyse the methods properly, we need to have separate data sets. Therefore, we partition the data sets into training and testing data sets, which contain respectively 75% and 25% of the complete data sets. The partition is independent on the date at which the products are introduced. This can be done since the RF and QRF methods are not time dependent; they do not consider trends in time and each product or prediction is considered independently. The training sets can be regarded as the existing products, whereas the testing sets can be regarded as new products. The training set is used for training the methods to make predictions and the test set is used for analysing the forecast and inventory performance.

4.4.1 Training the proposed method

We now describe the training phase, which is the phase where we use the training data to train the Random Forest and Quantile Regression Forest algorithms and tune the parameters to obtain appropriate results. We also analyse the feature importance, which shows on which product characteristics the predictions are mainly based. For the tuning the parameters and analysing the feature importance, we use the Out-Of-Bag (OOB) data. We first discuss the concept of OOB data.

Out-Of-Bag data

Out-Of-Bag data is the data which is not used for training the individual trees. Random Forest algorithms apply random sampling with replacement for training each tree. Hence, the probability that a product is randomly picked from n products is $1/n$ and the probability not to be picked is $1 - 1/n$. Consequently, the probability not to be picked after n times (i.e., complete random sampling with replacement) is $(1 - 1/n)^n \approx e^{-1} \approx 0.368$. Therefore, around 63.2% of the data is used for training each tree, while the remaining 36.8% is left out. This left out data is the Out-Of-Bag data. Each tree is trained on a different random subset, the OOB data differs per tree. Hence, all data of the training set is used for the complete Random Forest algorithm, but the OOB data is not used for training the individual trees. Therefore, it can be used for

tuning and analysing the Random Forest algorithms.

Using the Out-Of-Bag data removes the need for a separate validation set or cross-validation to tune the parameters and analyse the feature importance. Moreover, it only requires to train a Random Forest algorithm once to analyse the performance. Nevertheless, analyses with the OOB data use only approximately one third of the trees for analysing the Random Forest algorithm. Therefore, it is necessary to run a sufficient number of trees to obtain an unbiased OOB estimate of the performance (Breiman, 2001). In Section 3.2.4, we mentioned that a Random Forest usually stabilises within 500 trees. To make sure to obtain reliable results for the OOB analysis, we train each Random Forest with 2000 trees. Probst and Boulesteix (2017) also found that their OOB results of the algorithms all stabilised within 2000 trees.

Parameter tuning

The main objective of the training phase is to find the right parameters for the machine learning algorithms. As discussed in Section 3.2.4, the *mtry* parameter is the main parameter. By analysing the performance of the Random Forest algorithms with the Out-Of-Bag data, we will find the best value for *mtry*. For each data set and each RF and QRF algorithm, we will train the algorithms for *mtry* from 1 to the number of product features. For classifying the profiles, we evaluate the prediction accuracy and choose the value for *mtry* such that the accuracy is the highest. We evaluate the regression results of the RF and QRF algorithms with RMSE.

Feature importance

Besides finding the best value for *mtry*, we also use the training phase for analysing the feature importance. The feature importance provides insight into the trained Random Forest by showing the contribution of the different product characteristics to the prediction. We use the permutation feature importance approach (Breiman, 2001), which considers a feature importance if it has a positive effect on the performance of the algorithm. The larger the positive effect, the more important a certain feature is. To evaluate this, the Out-Of-Bag performance is determined first. Thereafter, any relation between a certain feature and the performance is nullified by permuting the values of the feature. The prediction performance is computed again and the difference is the permutation importance. This procedure is repeated for each feature.For the profile, we show the decrease in the accuracy. For the total demand, we present the percentage increase in the RMSE relative to the OOB RMSE without any permutations. The larger the difference, the more important the features are for the predictions. Although requiring some computational effort, the advantage of the permutation importance is that it is unbiased towards the number of categories within a feature (Probst et al., 2019). This is crucial, since the data sets all contain several categorical features with a varying number of categories.

4.4.2 Testing demandForest

To employ the proposed demandForest in practice, we want to guarantee valid results for making the predictions for completely new products. Since the training data is used for choosing the best value for *mtry*, we use the testing data to find the performance for unseen data. In the testing phase, we will evaluate the quality with multiple KPIs for the profile prediction, regression of the total demand, the corresponding prediction intervals, and the combined forecast for 18 weeks.

Predicting profiles

First, we discuss the KPIs for the clustering quality and the predictive performance of the profiles. Since the testing data is not used for clustering the demand profiles, we do not possess the actual profiles of the test data. Nevertheless, we can determine the squared error between the actual pattern and the profiles. In this way, we can assign the pattern to the profile where it should belong, and calculate the accuracy and kappa of the classification of the RF algorithm. Since an average profile is used for the benchmark method, we cannot compare the accuracy or kappa of the benchmark method. Instead, we analyse RMSE between the predicted profile and the actual normalised profile of the new products, and compare this with the RMSE of the average profile and the actual normalised profile. This makes sense, since k-means minimises the squared error within clusters. Since there are multiple profiles, it is expected that the RMSE of the predicted profiles is lower than the RMSE of the average profile. However, when the predictive accuracy is not sufficient, the RMSE of the average profile will be lower. In that case, the average profile is a better estimate. We analyse the RMSE of the normalised profiles and the normalised cumulative profiles. The RMSE of the normalised profiles is an indication for the forecasting quality and the RMSE of the normalised cumulative profiles for the inventory cases, as discussed in Section .

Total demand and prediction intervals

Besides the clustering, we evaluate the predictions of the total demand and the intervals. As discussed in Section 3.3, we use the RMSE as KPI for the prediction of the QRF algorithm. For the intervals, we use quantiles 0.05 and 0.95; these quantiles provide the 90% prediction interval. We determine the PICP and the PINAW for the intervals of each data set. The PICP is the ratio of predictions inside the interval, whereas the PINAW is the normalised average interval width. The schematic overview of the training and testing phase for the algorithms is illustrated in Figure 4.3.

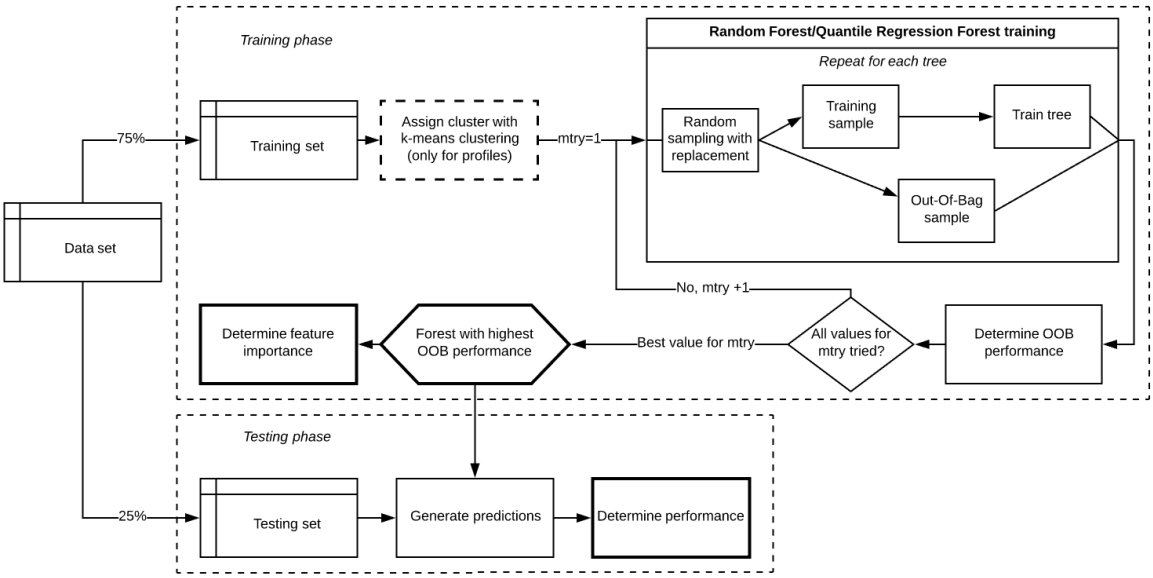


Figure 4.3: Training and testing phase for the individual algorithms

Combined forecasts

Next to the assessment of the performance of the clustering, and prediction of the profile and demand, we also assess the accuracy of the final forecasts by combining the individual predictions. These forecasts are used for predicting the weekly demand during the complete introduction period. We combine the profile and the total demand predicted by the RF and QRF algorithms. As discussed in Section 3.3, we will analyse the RMSE for the forecast accuracy. We do not use the MAPE or sMAPE, since these should not be used for evaluating forecasts with values of zero or close to zero. We compare all predictions and forecasts with a benchmark method, which we discus later in this section.

After evaluating the forecasting performance, we take the analysis one step further. Since the forecasts will be used for inventory management, we also analyse the inventory performance. We elaborate about this in the next part.

4.4.3 Application to inventory management

In this subsection, we describe how we set up the model for inventory management. We will analyse the models for a range of service levels. To analyse the methods, we will describe four different cases with different lead times and define a replenishment policy. Since the forecasts at clients of Slimstock are used to manage their inventories, it is important that the forecasts are not only accurate, but also lead to a good inventory management performance. Clients of Slimstock use demand forecasts to manage their inventories. Therefore, it is important that the forecasts are not only accurate, but also lead to proper safety stocks for inventory management. Since forecasts can never be 100% accurate, companies are confronted with a trade-off. If they want to assure their customers high service levels, they should put more products in stock than the demand forecast. However, this might also lead to large excess stocks. On the other hand, when they decide to have less products in inventory, they risk that they cannot satisfy the demand. In that case, customers may buy their products at competitors and companies miss out on revenue. They may not only lose revenue on these products, but also lose future sales, since customers can switch to competitors. Therefore, it is crucial for a company to find the right balance between service levels, excess stocks and lost sales.

To find a balance, we can use the distributions of the Quantile Regression Forest algorithms in order to evaluate different serivce levels. From the distributions, we can use a various number of quantiles, instead of specific point-estimates. The quantiles are similar to the target Cycle Service Levels. Quantile 0.8 means that there is a probability of 80% that the actual demand is lower than that value, which equals the probability of not having a stock out of 80%, which is a target CSL of 80%. Hence, we use the quantiles as setting for the target service levels, and decide to analyse the resulting CSL as KPI in this research. It is expected that the resulting CSLs are equal to the quantiles, and we will evaluate the consistency. Since the Fill Rate is not related to the quantiles of the QRF algorithm, we do not evaluate the Fill Rate.

Based of the quantiles of the QRF algorithm, we can determine can order sizes. Afterwards, we can analyse what the resulting performance is when ordering according to this values. We determine the Cycle Service Levels, and inventory costs for the quantiles from 0.50 to 0.99 with a step size of 0.01. The inventory costs consist of order costs, holding costs, excess holding costs, and lost sales costs, which we discuss later in this section. This inventory case has multiple objectives. First, by analysing a range of quantiles, we can find out if the given quantiles also result in similar CSLs for each data set. This shows the robustness and reliability of the methods. Second, we can see the costs of the different methods under different service levels. Third, we can determine the quantile that overall provides the lowest costs for a specific data set. Last, we will analyse the performance under different lead times of the products, such that we can

analyse how the model performs under different circumstances.

Each product has is own specific review time and lead time. The review time is the time between each moment a planner checks if he needs to order new products at a supplier, whereas the lead time is the time between ordering products at the supplier and the moment of putting the products in inventory. However, all the different review and lead times complicates comparisons between the products and the data sets. To simplify the analysis, we assume in this research four different inventory cases. The first case considers a one-time purchase of new products at the beginning of the introduction period. During the 18 weeks there is no option to do a second purchase at the supplier. This resembles the situation where products have long lead times (e.g., three months). For the other cases, we consider replenishing with lead times of respectively zero, two and six weeks.

The replenishment policy we use for these cases is a (R,s,S) policy, which stands for a policy with a review time R, reorder level s and order-up-to-level S. The review time R we will apply for all cases is one week. We define the reorder level s and order-up-to-level S based on the demand in the future weeks. To assure a certain service level, the forecast will be calculated by on the combination of a certain quantile of the total demand and the profile. To determine the reorder level s, we use the expected demand during the review time and lead time. In our case is this one, three, or seven weeks. The order-up-to-level will be the expected demand during the review time, lead time, and one week extra. We do not determine Economic Order Quantities (EOQs), since the general EOQ calculations assume a known and stable annual demand, and do not consider excess stocks. We also ignore Minimum Order Quantities, since these might jeopardise or bias the differences between different methods, because the MOQ may dictate an order size much larger that the order-up-to level.

With this policy, the inventory levels are reviewed every week. When the stock levels (including orders in the pipeline) have been dropped below the reorder level s, an amount of products is ordered such that the inventory is increased to S. For all cases, we can determine the CSLs, and costs for each quantile, each method and each data set. With this analysis for each data set, we can determine if demandForest and the extension achieves lower costs than the benchmark methods, and we can determine which quantile achieves the lowest costs under different circumstances. The cases with zero, two, and six weeks lead time, and the one-time order case resemble the different products with short lead times (next-day delivery/zero weeks) up to long lead times (multiple months/one-order).

Inventory costs

For the application to inventory management, we consider four different costs: order costs, holding costs, excess holding costs, and lost sales costs. To compare the results of the different data sets, we assume similar costs. We assume order costs of 25 euros for each individual order. For the holding costs, we assume 0.25 of the purchase price of a product as holding costs of keeping one product in inventory for one year. Besides regular holding costs, we take excess holding costs into account. Excess holding costs are the expected holding costs after the introduction period. When too much products are put in inventory in the introduction period, it may take a while to sell the excess products. To penalise too much products into inventory, especially when the actual demand is low, we define the excess holding costs besides the regular holding costs.

The excess holding costs can be regarded as a penalty for putting too much products in inventory, which are also not likely to be sold after the introduction period. To determine the excess holding costs, we assume a stable demand after the introduction period. To make a good estimate of this stable demand, we determined the factor of which the demand changes after the introduction. For this factor, we obtained from the data bases the average demand after

the introduction period if available. For example, when the average demand in the introduction period is 50 per week and the average demand after the introduction is 75 per week, the factor is 1.5. When there was no demand data available, because there was only 18 weeks of historical demand data, we multiplied the actual demand with the average factor of the other products in the data set. Table 4.1 shows the average factors of each company data set. With these factors, we can determine how long it takes to sell the products which are in stock after 18 weeks and we can calculate the corresponding holding costs. The low average factor of Company A can have multiple causes. For example, it might be possible that the Company actively promotes new products in their shops, while this extra attention disappears after a few months. On the other hand, Company B has a relative high average factor. This might be caused by the ranking of products in the web shop. Products in the web shop are, by default, sorted from the most sold to the least sold product. Hence, a new product is not likely to be on the front page when it is recently introduced. However, when the sales grow of a product, it also attains a rank in the web shop at which it is more likely that the product will be sold.

Company	Average factor
A	0.436
B	2.670
C	1.286
D	1.167
E	1.090

Table 4.1: Average factors of the future demand compared to the demand during introduction

The last type of costs we consider are the lost sales costs. In our case, we do not consider that customers come back later when the product is on stock again (backordering). Instead, we assume that when a product is out of stock, consumers go to a competitor and the sale is lost. Hence, the costs of a lost sale is at least the profit margin of a product. Since consumers go to a competitor for the product, there exists a risk that a consumer also goes to a competitor for future sales. Especially since it concerns new products, this can be a risk for a company. To take this risk into account, we assume lost sales costs of two times the margin. For the data sets that include the profit margin, the margin is around 50%. Hence, for the data sets which do not have the margin of a product available, we assume a margin of 50%.

4.4.4 Synthetic data set

Besides the real-world data sets, we also evaluate the proposed methods with a synthetic data set. To construct this data set, we define relations between the characteristics and the demand, and we define demand patterns. Hence, the distributions of the data and the relations between product characteristics and demand are known. Besides validating our methods, the data set can also serve as a benchmark instance for future research.

We create the synthetic data set such that there exist relationships between the product characteristics and the demand. In the real-world data sets, this may not be so clear. Therefore, validating the proposed methods with both a synthetic data set and real-world data sets is interesting. When the proposed methods can achieve similar results for the synthetic data set as well as the company data sets, it underlines the applicability of demandForest in practice.

Description of the synthetic data

We define the demand data of fictive articles such that they have 18 demand points and three profiles within this demand. The profiles are, arbitrarily, one with a 10% exponential increase

per demand point, one with a 10% exponential decrease per demand point, and one stable profile. The total demand is generated randomly with the Gamma distribution with $\alpha = 2$ and $\beta = 150$. In this case, the average demand will be 300. Each article is assigned randomly to a profile and demand value. The demand for each demand point is the profile multiplied by the total demand. Normal distributed noise is added to each demand point with a coefficient of variation of 0.25.

The articles come with four product characteristics: colour, category, brand, and price. The names of the colours, brands and categories are chosen arbitrarily. The colour and price will relate to the demand, whereas the category and brand will relate to the profile. Since we want to relate multiple characteristics to the demand or profile, we first determined the demand and profile. Afterwards, we assign the characteristics to these articles. We divide the demand in five equal segments. Specific colours, categories and brands relate to a specific demand segment or profile. To add some noise to the relations, 80% of the demand segment and profiles relate to the correct characteristics, while the other 20% is randomly assigned to other characteristics. The price is a numeric characteristic for which we define a relation with the demand. The price of an article is: $price = 2000/demand$. Hence, the price is inversely proportional to the demand, where a low demand corresponds with a high price, and a high demand with a low price. We apply a noise to this inverse proportional relationship by adding a coefficient of variation of 0.5 to the price.

In this case, there is noise between the product characteristics and the demand or profile and also noise within the demand and profile. In reality, the noise will not be normally distributed and there exist characteristics that to not relate directly to the demand or profile. We leave this out of the synthetic data set such that we can analyse the proposed methods under an controlled environment. An overview of the data set is shown in Table 4.2. Additional information about the synthetic data set can be found in Appendix C.

Considering the excess holding costs discussed in the previous subsection, we also need to assume the growth after the introduction period. In this case, we do not extrapolate the exponential growth, but assume that the growth flattens. Hence, we assume a factor of 3 for the increasing profile, a factor of 1 of the stable profile and a factor of 1/3 for the decreasing profile.

Demand			Gamma(2, 150)
Profiles	Increasing 10%, decreasing 10%, stable (Noise: CV=0.25)		
Colour	Category	Brand	Price
Black	Accessories	Animity	5000/demand
Blue	Computers	Dynotri	Noise: CV=0.5
Brown	Games	Hyperive	
Gray	Kitchen	Kayosis	
Green	Photography	Mudeo	
Orange	Smart home	Octozzy	
Purple	Sound	Outise	
Red	Tablets	Supranu	
White	Telephone	Transible	
Yellow	Television	Verer	

Table 4.2: Characteristics of the synthetic data set

4.4.5 Overview of experiments

In the previous subsections, we described the experimental design with several phases and methods. Therefore, we provide a small overview of these experiments. We execute all these experi-

ments for all data sets. The training experiments are summarised in Table 4.3.

Method/algorithm	Predict	Metric	Tuning mtry	Extra analysis
Predict profile	Profile	OOB accuracy	1 to features	OOB feature importance
Total demand	Mean	OOB RMSE	1 to features	OOB feature importance

Table 4.3: Experiments for training

As already described, we train the algorithms for predicting the profile and the total demand. We analyse the performance metrics of the predictions to tune the value for *mtry*. We want to find the values which obtain the best performance of the RF and QRF algorithms. Additionally, we determine the OOB feature importance for both algorithms.

Once we found the best values for the *mtry*, we can employ the algorithms to analyse the test data (Table 4.4). Again, we predict the profile and total demand. For the latter, we predict the mean and 90% Prediction Interval (PI). We will combine the profile and total demand for the 18 week forecast. We compare the RMSE of all predictions with the benchmark methods, and analyse the prediction intervals with the Prediction Interval Coverage Probability and the Prediction Interval Normalised Average Width.

Target	Method/algorithm	KPI
Profile	RF profile	Accuracy, kappa, RMSE
	Average profile	RMSE
Mean, 90% PI	QRF demand QRF+Log-Normal demand QRF+Gamma demand Proximity demand ZeroR demand	RMSE, PICP, PINAW
18 week forecast	demandForest (dF) dF + Log-Normal dF + Gamma Avg profile + proximity ZeroR profile & demand	RMSE

Table 4.4: Experiments for testing

After the evaluation of the forecast quality, we analyse the methods in an inventory setting. We identified four cases: three cases with lead times of respectively zero, two, and six weeks, a review time of one week, and a (R, s, S) replenishment policy, and one case with a one-time order of new products before the introduction period. We will generate a forecast with the quantiles from 0.50 to 0.99 with a step size of 0.01 for all methods. For each case, we use the forecasts of the methods as input for the two inventory cases and analyse the CSL, RF, and the inventory costs. An overview of the alternatives, which are applied to all data sets, is shown in Table 4.5.

The methods and experiments described in this chapter are implemented in the *R environment* (R Core Team, 2014) using the packages *Ranger* (Wright & Ziegler, 2015) and *fitdistrplus* (Delignette-Muller, Dutang et al., 2015). The *Ranger* package contains Random Forest (Breiman, 2001) and Quantile Regression Forest (Meinshausen, 2006), and *fitdistrplus* is used for the maximum likelihood estimations of the Gamma distribution, as described by Venables and Ripley (2013). An advantage of the implementation in R, is that the scripts eventually can be executed in-database in SQL Server with the Machine Learning Services feature ('SQL Server Machine Learning Services', 2019). SQL Server is the database management system used by Slimstock. In the next chapter, we describe the results of the experiments.

Case	One purchase Replenishment
Forecast	Quantiles 0.50 to 0.99
Method	demandForest dF + Log-Normal dF + Gamma Proximity ZeroR
KPI	CSL, total inventory costs, ordering costs, holding costs, excess holding costs, and lost value

Table 4.5: Different aspects of the inventory analysis

5 | Experimental results

In this chapter, we discuss the results of the proposed method. We apply the method to the synthetic data set and the five data sets from different industries. First, we train and tune the algorithms, and discuss the feature importance (Section 5.1). Thereafter, we discuss the performance of the individual algorithms in Section 5.2. In Section 5.3, we discuss the forecast quality of the proposed method. Lastly, we elaborate about the performance in inventory management (Section 5.4).

5.1 Training the algorithms

In this section, we will train the algorithms of the proposed method for each data set. For each data set, we tune the algorithms by analysing different values for *mtry*. Afterwards, we discuss the feature importances of the predictions of the profiles and total demand at each data set. In the last part of this section, we draw conclusions about the training phase. Note that only the training sets are used for this phase. The testing set will be used in the next section, where we analyse the performance of the algorithms.

5.1.1 Finding the best value for *mtry*

The proposed method for generating a forecast consists of a Random Forest algorithm and a Quantile Regression Forest algorithm. For each algorithm and the six data sets, we vary *mtry* between 1 and the number of features and choose the value for *mtry* such that the performance of the Out-Of-Bag data is the highest. As described in Section 4.4.1, we use the accuracy as measure for the profile classification and the RMSE as measure for the demand regression. The accuracy should be as high as possible, while the RMSE should be as low as possible. Table 5.1 shows the values for *mtry* at which the algorithms obtained the highest Out-Of-Bag performance.

Data set		Synthetic	A	B	C	D	E
Total number of features		4	8	5	9	4	4
Profile	mtry	1	2	1	3	1	2
	Accuracy	0.785	0.816	0.773	0.894	0.769	0.719
Demand	mtry	2	2	2	1	2	2
	RMSE	116.9	9.97	399.8	20.0	19.3	23.1

Table 5.1: Best value for *mtry* and OOB performance at each algorithm and data set

In the table, we see different results per data set. For the synthetic data set, we added 20% noise between the characteristics and the profile. Hence, an accuracy around 80% is expected. The data sets of the companies show a variety of results. Company A and C have a score on the profile accuracy which is higher than the accuracy of the synthetic data set. On the other hand, company B, D, and E score somewhat lower, maybe due to the lower number of product

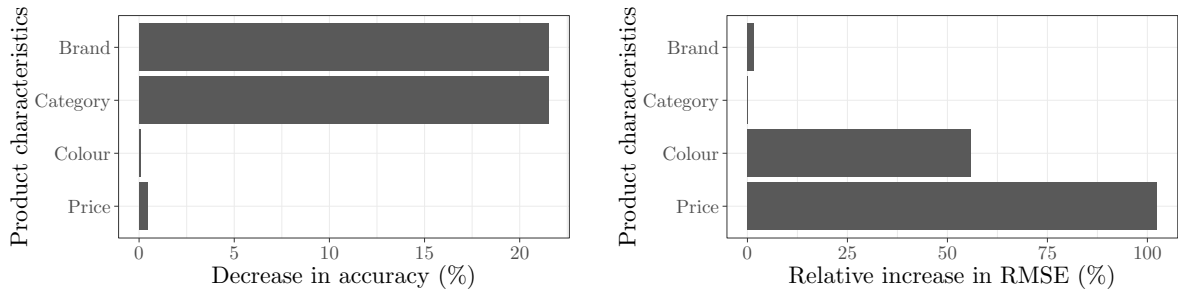
characteristics available compared to the other company data sets. The RMSE of company B is much larger than for the other companies. However, the RMSE is scale dependent and the average demand for company B is also much larger. The differences in performance can arise due to a various reasons, such as the relations (or absence of) between characteristics and demand, the number of characteristics, the similarity between products of a company, the number of products, number of clusters, or the size of the clusters. Nevertheless, to draw conclusions about the actual performance, we will first employ the algorithms with the determined values for *mtry* to the test set. In that case, we also analyse the prediction intervals and combined forecasts. Before we analyse the results of the method on the test sets, we evaluate the feature importance on each training set.

5.1.2 Feature importance

The feature importance shows which features (product characteristics) have a positive effect on the performance of the algorithms. These are interesting insights into the Random Forest and Quantile Regression Forest algorithms, since they explain which amount of influence certain characteristics have on the demand. We discuss the feature importances per data set.

Synthetic data

The feature importance of the synthetic data set is visualised in Figure 5.1. As expected by the design of the synthetic data set (Section 4.4.4), the two main features for profile prediction are the brand and category (Figure 5.1a). When each of these features is permuted, the prediction accuracy decreases about 23%. The price and colour have hardly any influence. When considering the total demand (Figure 5.1b), it is the opposite. The price has the highest impact, followed by the colour. As expected, these are the features that are important for predicting the demand, whereas the brand and category have no significant influence.



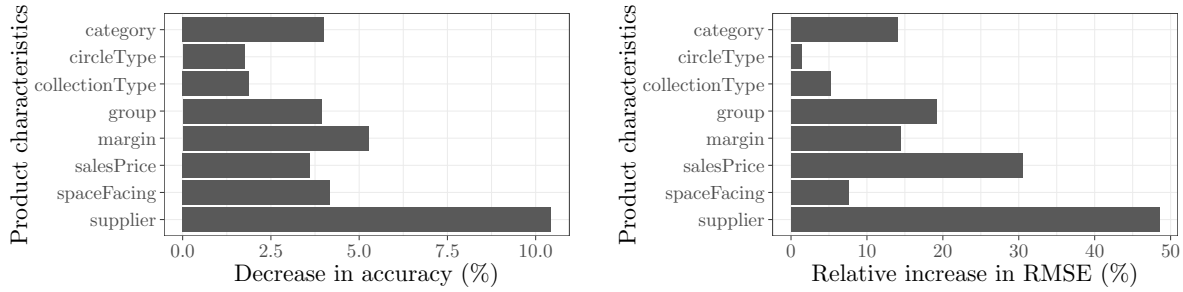
(a) Profile

(b) Demand

Figure 5.1: Permutation feature importances for the synthetic data set

Company A

The feature importances of company A are less clear than for the synthetic data set (Figure 5.2). The supplier is the most important for the profile (Figure 5.2a), with a decrease in accuracy of more than 10%. Other features have moderate effects and the collection type and circle type have the least impact. The supplier also has the largest importance for the total demand, which can be seen in Figure 5.2b. Furthermore, the sales price has a moderate effect of 30%, whereas the collection type and circle type again have the least effects.

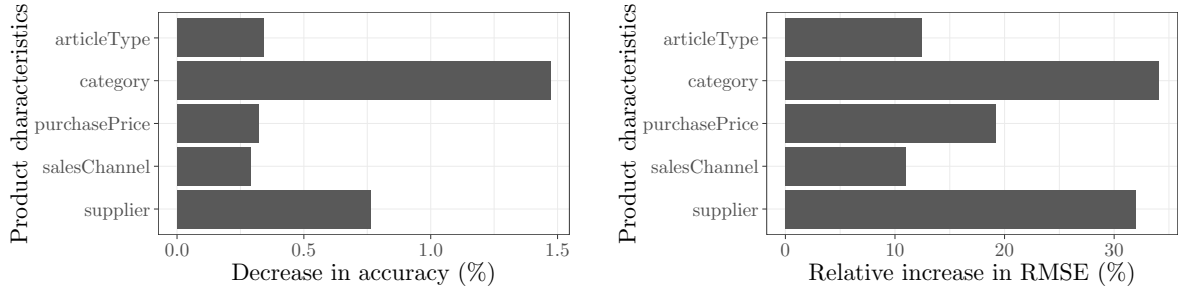


(a) Profile (b) Demand

Figure 5.2: Permutation feature importances for company A

Company B

The feature importance for company B is shown in Figure 5.3. In the profile prediction, the category seems to have the largest effect on the performance (Figure 5.3a). Nevertheless, this is still only 1.5%. The other features hardly have any effect on the predictive performance. For the total demand (Figure 5.3b), both the supplier and the category are rather important. Hence, the supplier and category are the most important for both algorithms. Nevertheless, the influence of the supplier is for company B much smaller than for company A.



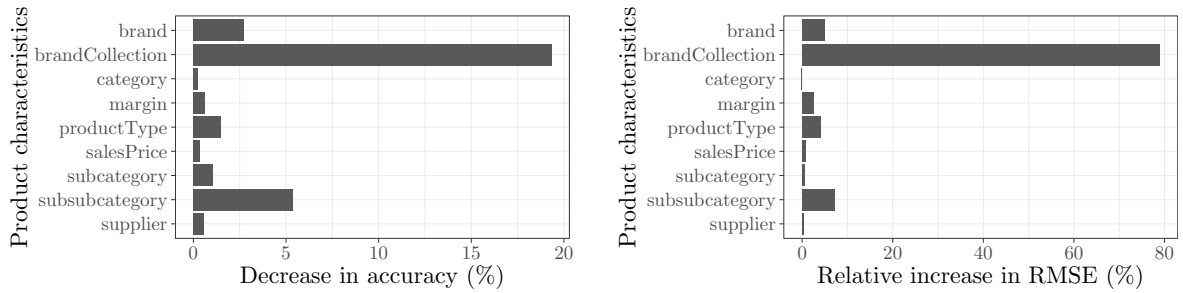
(a) Profile (b) Demand

Figure 5.3: Permutation feature importances for company B

Company C

Regarding the importance for the two predictions for company C in Figures 5.4a and 5.4b, we see comparable results between the two figures. The brand collection is by far the most important feature. This brand collection indicates the specific product lines of a brand. The subsubcategory, which is hierarchy the category below the subcategory, takes the second place for both. On the other hand, permuting the category hardly changes the accuracy or the RMSE. This means that random categories apparently provide a comparable performance than the

actual categories. The supplier has also a little impact on the performance of the prediction of the total demand, especially when compared to company A and B. The brand and brand collection as well as the category, subcategory, and subsubcategory are hierarchical characteristics, which are handled as separate features in the Random Forest (RF) algorithms. Hence, two products with the same category and subcategory, and a different subsubcategory, have two of these three features overlapping. The features with the lowest hierarchy (i.e., the brand collection and subsubcategory) show the highest importance. This indicates that more specific features can improve the predictions.

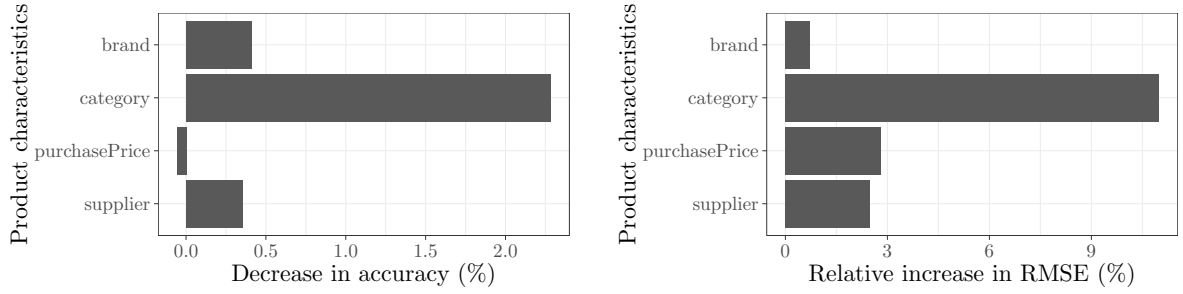


(a) Profile (b) Demand

Figure 5.4: Permutation feature importances for company C

Company D

The feature importance for company D shows other interesting insights (Figure 5.5). In both predictions, the category is the most important. In Figure 5.5a, we see that the category is the most important feature, but only changes the accuracy with around 2.5%. On the other hand, permuting the price results in an increase of the accuracy, possibly by chance since features are random permuted. For the total demand (Figure 5.5b), the price is of more importance, whereas the brand only has a small contribution. Compared to the other companies, the relative increases in the RMSE are low. The maximum increases are about 11%, whereas the data of other companies showed increases of more than 30%.

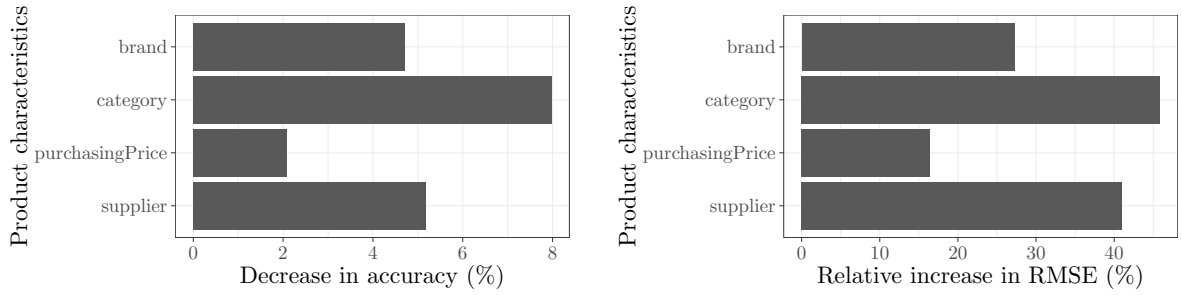


(a) Profile (b) Total demand

Figure 5.5: Permutation feature importances for company D

Company E

For company E, the category is also the most significant feature (Figure 5.6). After the category, the supplier is the most important feature for the profile (Figure 5.6a), whereas the price is the least important. The total demand in Figure 5.6b shows comparable results, with the category as the most important and the price the least important.



(a) Profile (b) Total demand

Figure 5.6: Permutation feature importances for company E

5.1.3 Conclusion on tuning and feature importance

In this section, we trained the individual algorithms of the proposed method for each data set. The first step was to find the best value for *mtry* to achieve the highest performance. We analysed the Out-Of-Bag performance of the algorithms for all possible values of *mtry* for each algorithm and each data set. The *mtry* was for the majority of the algorithms equal to 1 or 2.

Besides tuning the algorithms, we analysed the feature importances. We applied the permutation importance approach, which determines the effect of the individual features on the Out-Of-Bag performance. As expected, each data set showed different results. It is hard to draw any general conclusion, but the overall important features were mostly the supplier and the category. For company C, there was one important product characteristic, the brand collection. The prices did not always have large impact on the accuracies of the models for the different data sets, although we saw certain patterns between the price and the demand in the data analysis of Section 2.4. Moreover, the features had little impact on predicting the profile of company B. Also the features seemed to have little impact on the predictions for company D. When all features have a low importance, it can be difficult to achieve a high predictive performance. To investigate this, we will assess the performance with the test data set in the next section.

5.2 Testing the methods with unseen data

Now we determined the best value for *mtry* for each algorithm and data set, we can analyse the actual performance of the method with the test set. The test set is the data set with introduced products that is not used for training the algorithms. Hence, we can regard this as new products. For these products, we generate predictions to assess the performance of the methods.

5.2.1 Predicting the profile

First, we employ the Random Forest algorithms trained for predicting the profile. After predicting the profile, we determine which profiles have the lowest Euclidean distance to the actual demand pattern and determine the accuracy and kappa. The accuracy and kappa score indicate the predictive performance for the profile, but these results cannot be compared to the performance of the average profile. Therefore, we calculate the RMSE between the predicted profiles and the actual profiles for each data set. As benchmark, we determined the average profiles of the training sets and calculate the RMSE between these average profiles and the actual demands. Table 5.2 shows the results for the test data. The best results per data set for the RMSE are in bold.

Data set		Synthetic	A	B	C	D	E
RF profile	Accuracy	0.824	0.802	0.746	0.568	0.826	0.770
	Kappa	0.736	0.534	0.200	0.197	-0.007	0.489
	regular RMSE	**0.021**	**0.055**	0.093	0.137	0.095	**0.109**
	cumulative RMSE	**0.089**	**0.139**	0.210	0.330	0.204	**0.226**
Average profile	regular RMSE	0.026	0.057	0.093	**0.121**	**0.092**	0.115
	cumulative RMSE	0.125	0.154	**0.205**	**0.267**	**0.194**	0.236

Table 5.2: Results for predicting the demand profiles of the test data

Since the k-means algorithm minimises the within cluster sum of squares given the number of clusters, it is expected that the RMSE of the predicted profiles lower is than the RMSE of the average profile. However, since we predict the profile, we are not sure that we have predicted the right profile and the result may be worse than the average profile. We see that the RMSE for the synthetic data set, company A and E are lower for the predicted profile than for the average profile. For company B, the regular RMSE is equal for both profile methods, and the cumulative RMSE is lower for the average profile. For company C and D, the RMSE of the predicted profile is greater than the RMSE of the average profile. Interestingly, the same pattern is seen with the kappa values. The kappa for the synthetic data set, company A and E are all relatively high, while the kappa value of company B, C, and D are at most 0.200. For the accuracy, we do not observe such an effect, probably because it does not consider the imbalanced classes of the profiles. Especially the profiles of company D is imbalanced with one predominant profile. This resulted in a high accuracy but a very low kappa. There seems to be a link between the feature importance and these results. The feature importance for the profile of company B and D are all quite low (max 2.5%). Moreover, company C has only 1 feature which is particularly important, while the others are also relatively low.

The differences in predictive performance and clustering quality indicate that this method of predicting a profile with k-means and Random Forest might not always be beneficial to the forecasting accuracy. For these data sets, a kappa with a value between 0.2 to 0.4 seems appropriate as a threshold whether to apply a profile or not. Since these profiles only concern the normalised demand, we will also analyse the forecast quality of the predicted and average profiles when combined with the prediction of the total demand. First, we focus on the prediction of this total demand.

5.2.2 Predicting the total demand during the introduction period

We predicted the total demand and the 90% prediction interval with the Quantile Regression Forest algorithm, the extended Log-Normal and Gamma distribution, and the benchmark methods. For these predictions, we determine the RMSE, the PICP, and the PINAW. These results are displayed in Table 5.3.

In the table, we see that demandForest achieves a lower RMSE than the Proximity and ZeroR method in almost all cases. Only for the data set of company D, the Proximity method is more accurate. Regarding the extended methods, the Log-Normal distribution results in less accurate predictions than the standard demandForest, whereas the Gamma distribution improves the accuracy for all data sets except C. The PICP and PINAW should be considered together. The PICP is ideally around or above the 90%, whereas the PINAW should be small. For the synthetic data, demandForest + Gamma shows the best results, with a PICP of around 90% and one of the smallest PINAWs. The PINAW of demandForest is smaller, but the PICP is less than 90%. demandForest + Gamma is also the best for all company data sets. For all these data sets, the PICP is around 90% with a low PINAW. DemandForest + Log-Normal is also accurate for

Data set		Synthetic	A	B	C	D	E
RMSE	demandForest	120.6	7.58	399.5	17.7	15.1	20.0
	dF + Log-Normal	120.0	7.81	400.9	19.0	15.6	22.9
	dF + Gamma	119.7	7.50	386.0	17.8	14.8	19.7
	Proximity	161.8	8.14	529.0	23.2	14.2	27.3
	ZeroR	212.6	8.77	485.6	18.6	17.4	27.4
PICP	demandForest	87.6%	93.3%	84.9%	94.6%	93.9%	91.5%
	dF + Log-Normal	90.6%	94.0%	86.7%	93.2%	91.5%	90.3%
	dF + Gamma	90.6%	94.9%	87.6%	93.9%	92.2%	91.5%
	Proximity	96.8%	91.2%	82.5%	78.4%	81.2%	81.2%
	ZeroR	93.2%	90.0%	91.0%	95.3%	95.2%	95.2%
PINAW	demandForest	23.4%	4.5%	12.6%	22.8%	15.5%	27.4%
	dF + Log-Normal	25.0%	4.2%	15.9%	18.4%	14.6%	34.0%
	dF + Gamma	23.6%	4.0%	11.9%	18.6%	13.3%	25.0%
	Proximity	76.3%	5.1%	15.5%	18.2%	11.0%	21.6%
	ZeroR	55.9%	5.2%	19.3%	29.1%	14.4%	32.8%

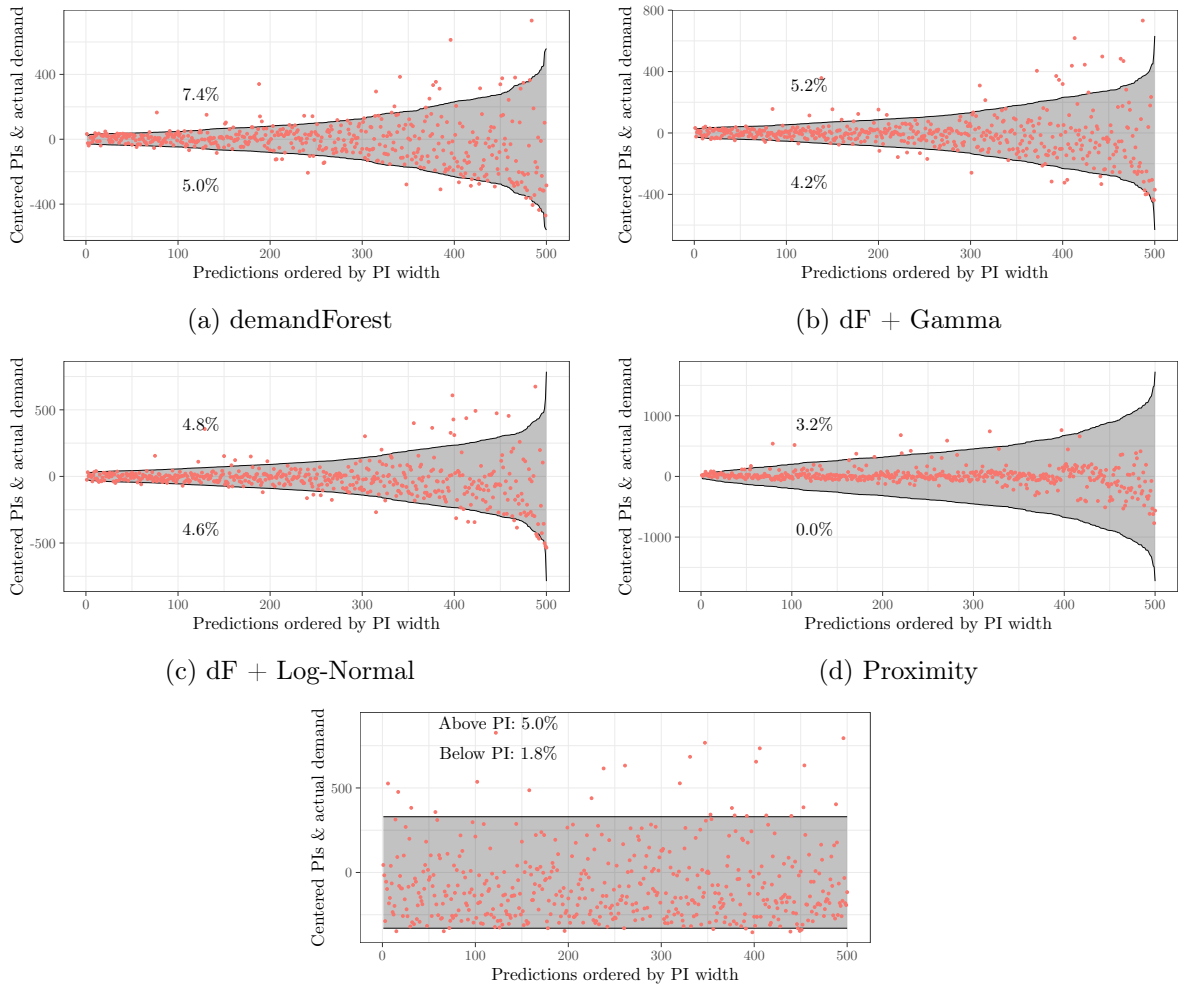
Table 5.3: Results for predicting the total demand

company C, with a comparable PICP and PINAW. The proximity method shows small values for PINAW for company C, D, and E, but the PICP for these data sets is only around 80%.

We plot the prediction intervals and actual demand for each method, based on graphical representation suggested by Meinshausen (2006). In Figure 5.7, we see the different prediction intervals for the synthetic data set. The figure shows the predictions ordered by the width of the prediction intervals. The red dots are the actual demand values. The graphs also show the percentage of predictions above and below the intervals.

In the first four figures, Figure 5.7a to 5.7d, we see that the lengths of the prediction intervals vary within a data set. This means that the demand of some products can be predicted more accurately than other products. Especially for the demandForest methods, we see that the outliers above and below the interval are larger when the width is larger. The prediction intervals of the Proximity method, Figure 5.7d, are clearly too wide, which we also observed for the PINAW. The ZeroR method just uses the overall 0.05 and 0.95 quantiles of the training set, hence the width of the 90% interval is the same for each product (see Figure 5.7e).

The percentages of predictions above and below the intervals should all be around 5%. Nevertheless, the percentage above the interval for the regular demandForest (see Figure 5.7a) is slightly higher. The extended demandForest methods improve this value. The percentages for the Proximity method are on the low side. The percentage below the interval for the ZeroR method is with 1.8% also rather low.



(a) demandForest

(b) dF + Gamma

(c) dF + Log-Normal

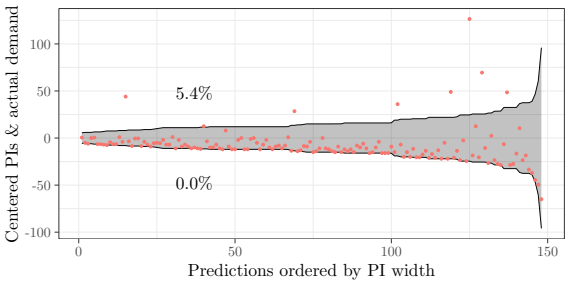
(d) Proximity

(e) ZeroR

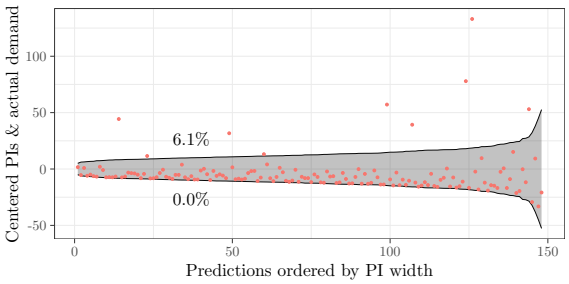
Figure 5.7: Centered 90% prediction intervals for each method for the synthetic data

Besides the interval plots of the synthetic data, Figure 5.8 shows the prediction intervals for company C. We observe similar results for the intervals of company C compared to the synthetic data. Clearly, some articles can be forecasted more accurately than others. The results of the different demandForest methods are comparable, with slightly higher percentages above and below the interval for the extended methods (see Figure 5.8b and 5.7c). Moreover, the widest intervals of the extended methods are around the 100, whereas the widest intervals of the demandForest method are almost 200 in Figure 5.8a. For all three methods we observe that the actual demand points are skewed to the lower prediction interval. This may also explain the low percentages below the prediction intervals.

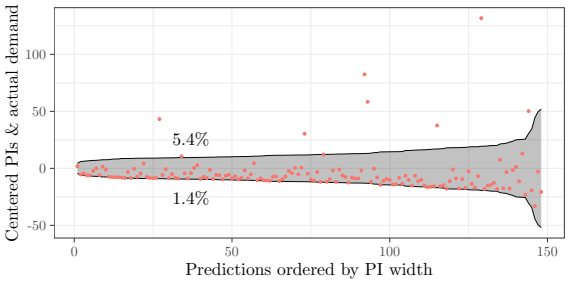
The Proximity intervals are in this case also not very accurate, since the percentage above is 21.6% (see Figure 5.8d), which is not even close to 5%. On the other hand, the percentage below is 0%. Additionally, the widest intervals of the predicted demand are more than 350 wide, while the actual demand for these widest intervals are relatively low. The ZeroR method is again very simplistic and the actual demands are again skewed to the lower prediction interval. For the interval graphs of the other data sets we refer to Appendix D.



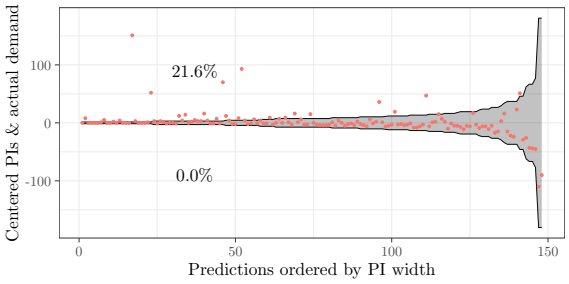
(a) demandForest



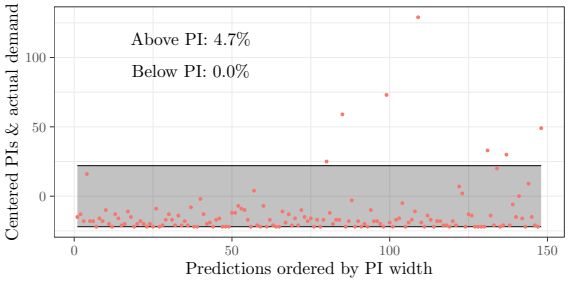
(b) dF + Gamma



(c) dF + Log-Normal



(d) Proximity



(e) ZeroR

Figure 5.8: Centered 90% prediction intervals for each method for company C

5.3 Forecast accuracy of the proposed methods

Now we combine the predicted profiles and the predicted total demand. We assess the forecast quality with the RMSE of the regular forecast and the cumulative forecast. The results of all methods for each data set can be found in Table 5.4.

Data set		Syn	A	B	C	D	E
Regular forecast	dF & predicted profile	10.8	0.681	32.5	2.22	3.59	3.61
	dF & average profile	11.8	0.681	34.3	1.92	3.44	3.80
	dF + LN & predicted profile	10.7	0.683	32.6	1.83	3.03	3.31
	dF + LN & average profile	11.7	0.688	33.8	1.79	3.04	3.50
	dF + G & predicted profile	10.8	0.676	32.0	2.15	3.48	3.56
	dF + G & average profile	11.7	0.678	33.7	1.89	3.34	3.77
	Proximity & avg profile	13.2	0.705	40.2	2.57	4.11	5.36
	ZeroR	15.2	0.784	36.4	1.78	3.15	4.24
Cumulative forecast	dF & predicted profile	84.0	5.65	218.6	11.1	13.0	16.2
	dF & average profile	86.2	5.66	228.1	10.8	12.8	16.5
	dF + LN & predicted profile	83.5	5.82	217.5	11.6	12.6	18.2
	dF + LN & average profile	85.9	5.83	217.3	11.5	12.8	18.7
	dF + G & predicted profile	83.7	5.61	211.2	11.1	12.7	16.2
	dF + G & average profile	85.8	5.61	217.7	10.9	12.5	16.4
	Proximity & avg profile	106.8	6.01	308.8	15.6	13.2	23.7
	ZeroR	135.3	7.34	258.8	11.0	14.1	23.1

Table 5.4: RMSE for regular and cumulative forecasts

The forecasts of the demandForest methods overall have a lower forecast error. The results of company C are the only exception with ZeroR as lowest RMSE for the regular forecast and one of the lowest for the cumulative forecast. Comparing the methods with the predicted profile and with the average profile, we see that the predicted profiles have a lower RMSE for the regular forecast, except for company C and D. This does not come as a surprise, since the normalised profiles were also worse for these data sets. This supports the results of the profiles, where the average profiles were also better for company C and D. Only the demandForest + Log-Normal provides better results with the predicted profile than the average at company D. In Section 5.2.1, the RMSE of the average and predicted profile of company B were comparable. For the combined forecasts, company B provides similar results. Overall, the predicted profile is better, but the average profile has a slightly lower RMSE for the cumulative forecast of demandForest + Log-Normal.

Considering the profiles and forecast accuracy, the forecast accuracy provides better results when the profiles are clear and can be predicted relatively good (i.e., lower RMSE for the predicted profile than the average profile and kappa with value above 0.4). The predicted profile and the Log-Normal or Gamma extension of demandForest provide the best forecasts overall. To further analyse the performance of demandForest and the other methods, we will analyse the application of the methods into an inventory setting.

5.4 Inventory performance of the methods

In this inventory case, we do not use the average forecasts, but employ the corresponding quantiles to define the service level and corresponding order levels. We analyse the Cycle Service Levels, Fill Rates, and the inventory costs for the quantiles 0.50 to 0.99, as discussed in Section

4.4.3. There are multiple objectives for this inventory case: comparing the consistency between the quantiles and the actual CSLs, determining the expected costs at different service levels, finding quantiles with the lowest costs, and analysing these results for different settings of the lead time. We discuss these results in the next subsections.

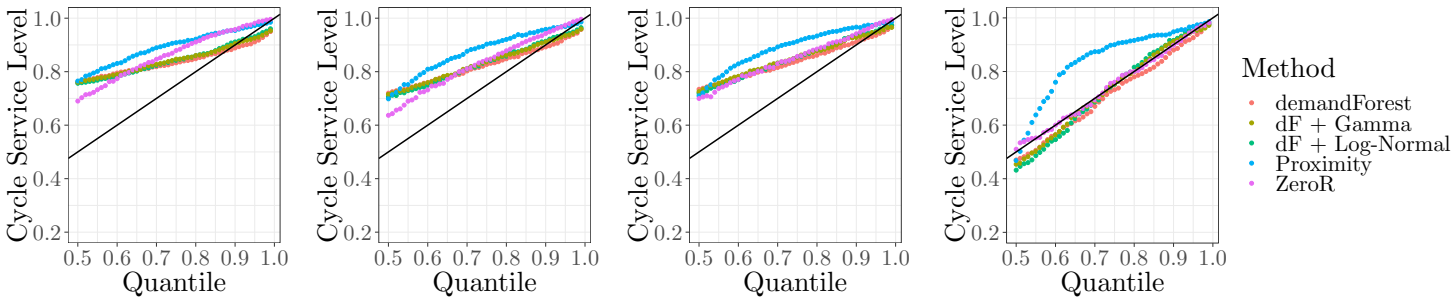
5.4.1 Consistency between quantiles and CSLs

In this section, we elaborate on the consistency between the quantiles, which are the known values between 0.50 and 0.99, and the Cycle Service Level (CSL)s, which are the actual ratios of not having a stock out after applying the quantile-based forecasts. The CSLs should theoretically be equal to the quantiles. For the cases with a one-time order, this is likely. However, with the replenishment cases, we multiply the quantiles with the predicted or average profiles. Hence, the actual CSLs might differ from the given quantile. The method is more reliable when the deviations between the quantiles and the CSLs are small, because then the company can achieve a CSL similar to the target service level. On the other hand, significant deviations indicate an unreliable method, which complicates companies to achieve the target service level. We discuss the results per data set.

Synthetic data

The results of the quantiles and corresponding CSLs of the synthetic data set are shown in Figure 5.9. The quantiles (i.e., the target service levels) are on the horizontal axes and the CSLs (i.e., the actual service levels) are on the vertical axes. The black lines in the figures are reference lines, which illustrate the ideal result for the methods. Each graph represents a case with different lead time for the products.

All methods, except Proximity, show reliable CSLs for the one-time order case (Figure 5.9d). The CSLs for the proximity method are only comparable to the quantiles around 0.50 and near to 1. For the replenishment cases, we observe comparable results. The proximity method still shows the high CSLs, whereas the service levels of the other methods also increase, especially with lower quantiles. For the lead time of zero weeks is the ZeroR method also a lot higher above quantile 0.75. Hence, the demandForest methods seem to provide the most reliable results, with no distinctive differences between the alternatives. Nevertheless, for the quantiles below 0.85, the methods result in higher service levels than intended.

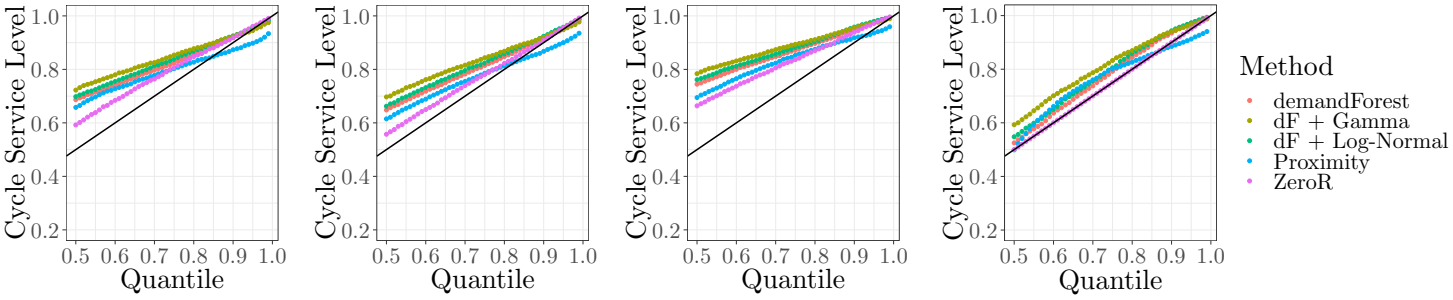


(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.9: Quantiles and Cycle Service Levels for the synthetic data

Company A

Figure 5.10 shows the results of the first company data set, which appear to be very comparable for each case. In all cases, the Proximity method resulted in lower service levels than expected for the highest quantiles, whereas the demandForest + Gamma method overestimates the CSLs the most over the complete range of service levels. The other methods are quite reliable for the one-time order case (see Figure 5.10d), with the ZeroR as most reliable. The CSLs for the replenishment methods are higher than the quantiles, with reliable results for higher quantiles. The proximity has the most unreliable patterns, since it overestimates the CSLs at low quantiles and underestimates the CSLs for higher quantiles. The most reliable method for all cases is the ZeroR, followed by the regular demandForest method, which all have upside deviations that increase with lower quantiles.

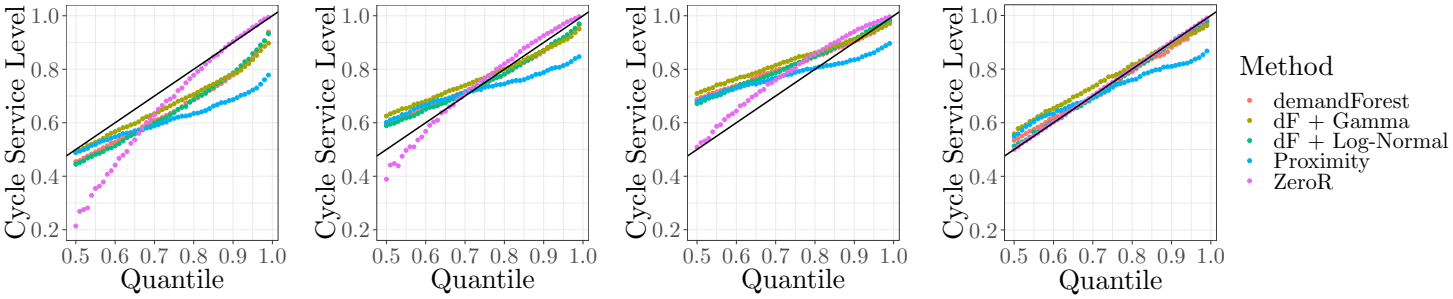


(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.10: Quantiles and Cycle Service Levels for company A

Company B

In Figure 5.11, we again observe reliable results for most methods except the Proximity method. The Proximity method is unreliable for all cases, even more clear than we observed at company A. The demandForest methods all have comparable results in each case, which are more reliable than the benchmark methods. However, for the lead time of zero weeks, the demandForest shows methods deviations up to 10%, see Figure 5.11a. Additionally, the demandForest + Gamma method results in slightly higher CSLs. For the lead time of zero weeks, Figure 5.11a, the ZeroR method is the most reliable for the highest quantiles. However, for quantiles below 0.75, the CSLs drop to 20% instead of the target of 50%.

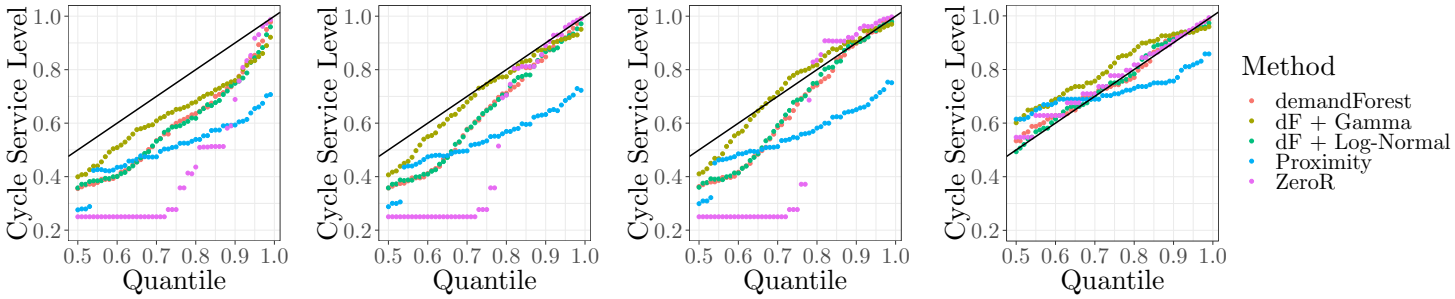


(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.11: Quantiles and Cycle Service Levels for company B

Company C

In the subfigure with the one-time order, Figure 5.12d, the Proximity method is again underperforming. The demandForest and demandForest + Log-Normal are most reliable, whereas the ZeroR shows very small upside deviations and the demandForest + Gamma some larger upside deviations. For the replenishment cases, the CSLs of the demandForest + Gamma are most similar to the quantiles (i.e., the target CSLs), followed by the other demandForest methods. The CSLs of the Proximity method are too low for all replenishment quantiles. The results of the ZeroR method are very low for lower quantiles, but comparable for quantiles above 0.8 and the lead times of two and six. The low and step-wise CSLs of the ZeroR method are a result of similar values of the quantiles in the training set. Since this data set contains a lot of products with a weekly demand of zero, a lot of forecasts are also zero for the ZeroR method. This changes for quantiles above 0.75. For the one-time order, these low step-wise CSLs do not occur since it uses the quantiles over the complete introduction period, instead of the weekly quantiles.

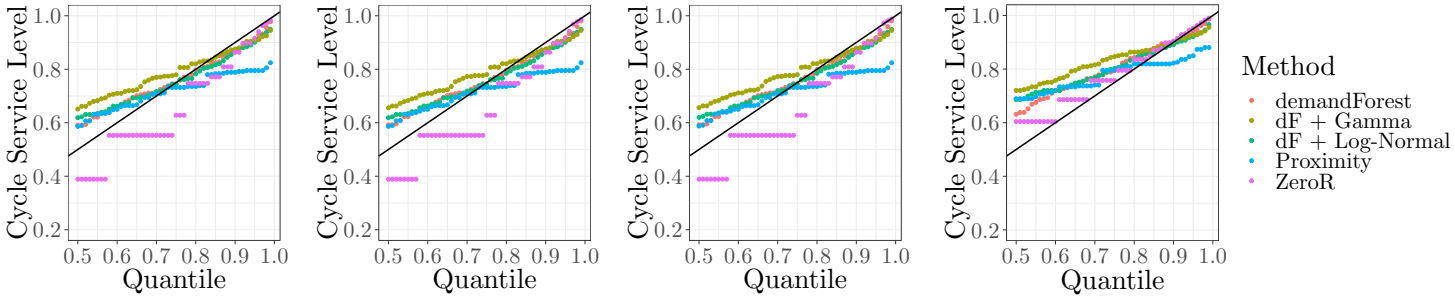


(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.12: Quantiles and Cycle Service Levels for company C

Company D

The Proximity measure results again in unreliable service levels, with too low CSLs for the higher quantiles in all plots in Figure 5.13. The ZeroR method results in step-wise CSLs, similar to the results of company C. These CSLs are consistently lower than the quantiles for replenishment (see Figure 5.13a to 5.13c) and higher for the one-time order (see Figure 5.13d). The observed service levels for the demandForest + Gamma method are mostly the quantiles. The demandForest and demandForest + Log-Normal seem to provide the best results.

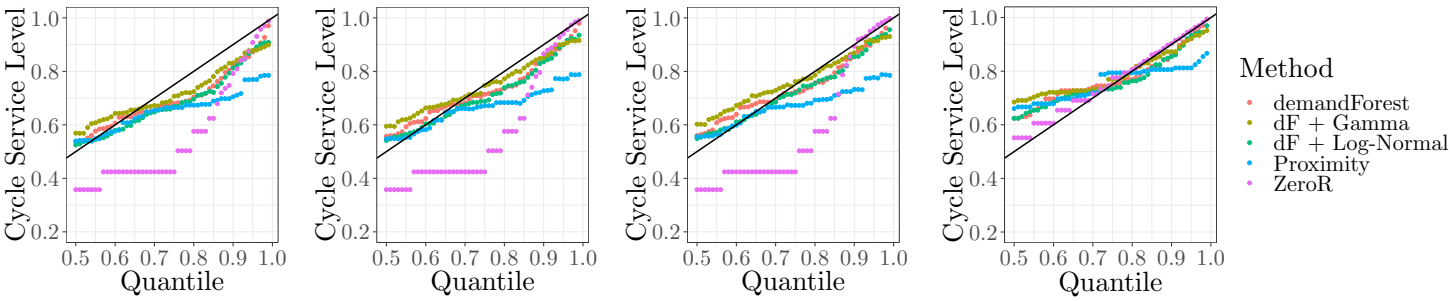


(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.13: Quantiles and Cycle Service Levels for company D

Company E

The results for the data of company E are comparable to the results obtained from company D. In Figure 5.14, the Proximity method results in too low CSLs for the higher quantiles and shows the ZeroR method large deviations for the replenishment cases (see Figure 5.14a to 5.14c). All three demandForest methods have consistent results. For the replenishment cases, the demandForest methods results are somewhat low, with the demandForest + Gamma as best. Again, the demandForest + Gamma method is slightly higher than the other demandForest methods.



(a) Replenish, LT = 0 (b) Replenish, LT = 2 (c) Replenish, LT = 6 (d) One-time order

Figure 5.14: Quantiles and Cycle Service Levels for company E

Overall observations

Considering the results of the quantiles and CSLs of all data sets, we observe overall comparable and robust results for the demandForest methods. The demandForest + Gamma method results overall in slightly higher CSLs than the regular demandForest and Log-Normal method, whereas these methods are comparable. The overall most unreliable method is the Proximity method. This method provides in some cases too high service levels and especially with higher quantiles too low service levels. For the last three data sets (i.e., company C, D, and E), the ZeroR method is also unreliable for lower quantiles, while ZeroR provided accurate and robust results for the first three data sets (e.g., synthetic, company A and B). The main characteristics from these data sets is that the first three data sets contain a larger number of products and higher demand volumes. The last three data sets contained a lower number of products and also a larger number of products with a total demand of only 1.

Comparing the forecasts of Section 5.3 with the results of the service levels, we observe the similarity. For the forecasts, we already observed that most prediction intervals of the proximity were below 90%. We observe the same effect for the actual Cycle Service Levels that are lower for the Proximity method. Additionally, the demandForest methods are all quite reliable for quantiles above 0.80. Hence, also the 95% quantile, used for the 90% prediction interval, provides good results.

Besides the consistency between the quantiles (e.g., the target service levels) and the actual CSLs, it is also important that the methods provide good financial results for the company. Hence, we analyse the inventory costs in the next subsection.

5.4.2 Inventory costs for different service levels

In this subsection, we evaluate the total inventory costs corresponding to the Cycle Service Levels. Similar to the previous subsection, we plot four figures for the different cases of each data set. We plot the total inventory costs against the actual CSLs. We also show vertical lines at the service levels of 0.50 and 0.99, representing the bounds of the quantiles, which should be

equal to the limits of the service levels. Nevertheless, in the previous section we already showed that this is not always the case. We discuss the results for each data set individually. We plot the costs against the actual CSL, not against the quantiles. In the previous subsection we observed that the quantiles are not completely similar to the CSLs. We use in this subsection the CSLs, since these are observed by the customers of the different companies and we evaluate the costs of the different methods corresponding with these observed service levels.

Synthetic data

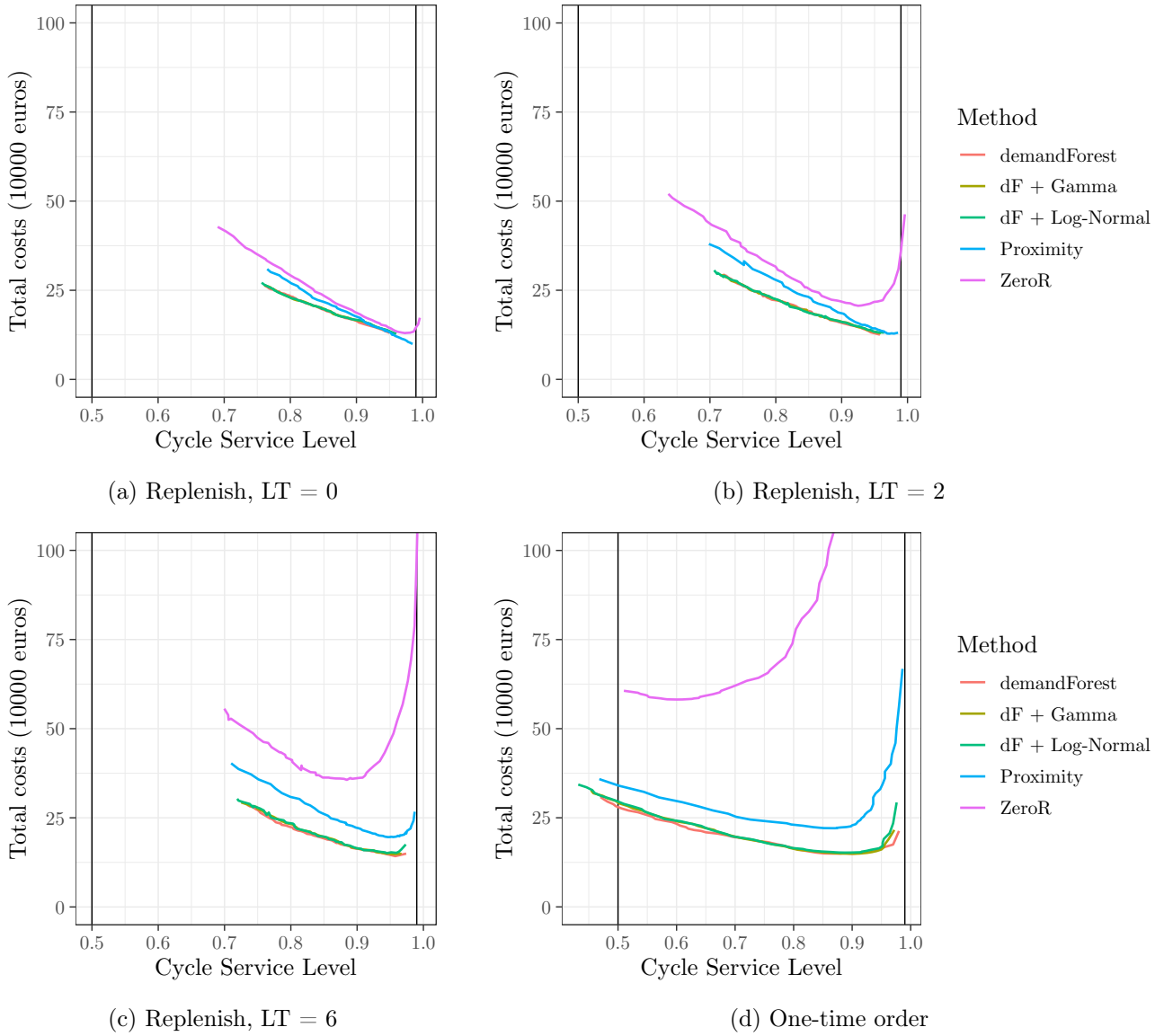
For the replenishment cases, we observe that the service levels are not always equal to the quantiles. The service levels have a narrower range than the targeted quantiles of 0.50 to 0.99 (see Figure 5.15a to 5.15b). As discussed in Section 4.4.3, planners should find a balance between service levels and inventory costs. More products in inventory (i.e., with larger quantiles/order levels/CSLs) result in less lost value and order costs, but increase the (excess) holding costs. Hence, the results of the inventory costs can differ from the forecasting results. Whereas the RMSE penalises each deviation from the actual demand, the inventory costs can decrease due to a higher or lower quantile. Usually there is a service level where there is a balance between the costs such that the combined costs are the lowest. We observe this effect clearly for the ZeroR methods in the subfigures of Figure 5.15, where the costs decrease to a minimum and thereafter increase significantly as the quantile/service levels approach 1.

In all subfigures of Figure 5.15, we observe that the resulting costs of the demandForest methods are below the benchmark methods. However, for the lead time of zero weeks (see Figure 5.15a) the costs of the Proximity method drop below the demandForest methods for the highest service levels. This occurs since the Proximity method has a CSL of 0.985, while the CSL of the demandForest methods is around 0.96 instead of the targeted 0.99. For these two cases, the lowest costs are achieved by the Proximity method with quantile 0.99 as input, see Table 5.5. These costs are lower than the lowest costs of the other cases. For the lead time of two and six weeks and the one-time order case, the lowest costs are obtained by the demandForest method.

A key finding is that the quantile/CSL with the lowest costs becomes higher when the lead time decreases. Furthermore, the costs of the different methods become more comparable. Additionally, the holding costs, excess holding costs and lost value decrease when the lead time is shorter. The order costs generally increase. This makes sense, since shorter lead times result in lower order levels and more frequent ordering. These lower order levels avoid large excess stocks at the end of the introduction period, while more frequent ordering prevents lost sales. In other words, shorter lead times makes it easier to anticipate the demand. And this anticipation comes at the costs of ordering.

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	Proximity	demandForest	demandForest	demandForest
Quantile	0.99	0.99	0.98	0.92
CSL	0.985	0.958	0.958	0.900
Total costs (€)	99,397	125,602	142,512	149,138
Order costs (€)	68,050	69,000	43,900	25
Holding costs (€)	20,445	22,522	41,657	82,287
Excess holding costs (€)	4,634	6,129	24,626	30,358
Lost value (€)	6,268	27,951	32,328	36,468

Table 5.5: Results of the methods with lowest costs per case for the synthetic data



(a) Replenish, LT = 0

(b) Replenish, LT = 2

(c) Replenish, LT = 6

(d) One-time order

Figure 5.15: Cycle Service Levels and inventory costs for the synthetic data

Company A

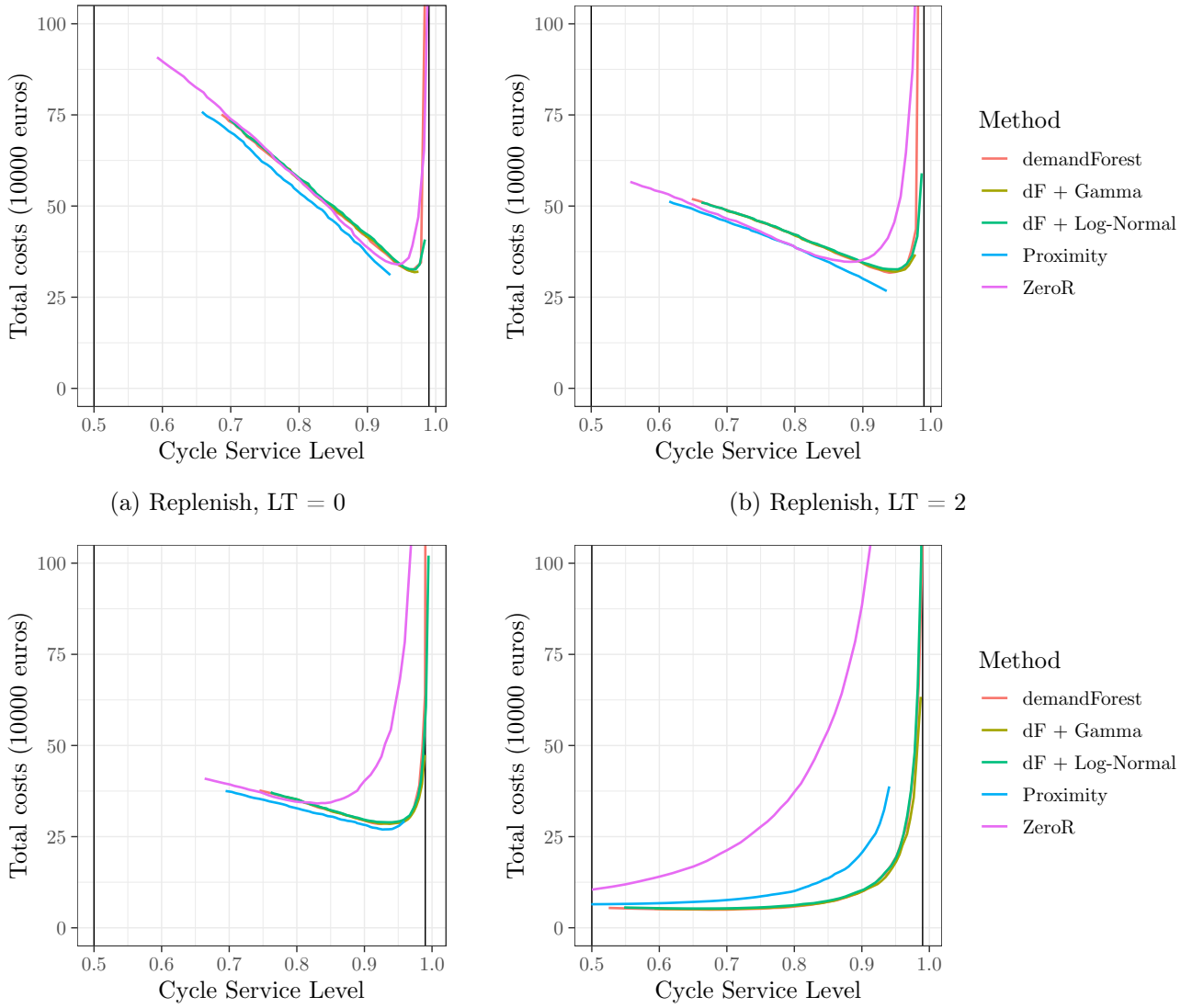
The results of the replenishing case with lead times of zero shows comparable results between all methods, see Figure 5.16a). However, the Proximity method achieves the lowest for all service levels, although it has a limited range of s. With a service level of 0.934, the Proximity method is slightly cheaper than the demandForest methods at the CSLs around 0.97. The demandForest methods are only better than the benchmarks at service levels higher than 0.95.

For the lead time of two weeks, Figure 5.16b, the Proximity method again provides the lowest costs. Also the ZeroR method obtains lower costs than the demandForest methods for service levels below 90%. Only for the highest service levels, which are not achieved by the Proximity method, the demandForest methods are better. The lowest costs are obtained by the Proximity method at its highest CSL (see Table 5.6).

For the lead time of six weeks, the costs of the Proximity method become similar to the demandForest methods. The ZeroR method achieves only comparable results for the lower service levels, see Figure 5.16c. For the case without replenishment are the demandForest methods the best. It must be noted that the costs of the demandForest methods rapidly increase for service

levels near 100%. The lowest costs are achieved by the demandForest method with a CSL of 0.675. Nevertheless, all costs with a service level between the 0.50 and 0.90 are quite similar.

Regarding the costs of the best performing methods in Table 5.6, we see that the order costs have a significant impact on the costs. Mainly due to these order costs, the total costs of the one-time order case are around six times smaller than the other cases. Additionally, the excess holding costs decrease for longer lead times, while the lost value increases.



(a) Replenish, LT = 0

(b) Replenish, LT = 2

(c) Replenish, LT = 6

(d) One-time order

Figure 5.16: Cycle Service Levels and inventory costs for company A

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	Proximity	Proximity	Proximity	demandForest
Quantile	0.99	0.99	0.93	0.64
CSL	0.934	0.935	0.927	0.675
Total costs (€)	310,892	266,843	269,666	49,694
Order costs (€)	285,375	211,800	186,675	25
Holding costs (€)	2,715	4,624	5,781	3,720
Excess holding costs (€)	13,430	37,754	60,566	11,599
Lost value (€)	9,372	12,665	16,643	34,351

Table 5.6: Results of the methods with lowest costs per case for company A

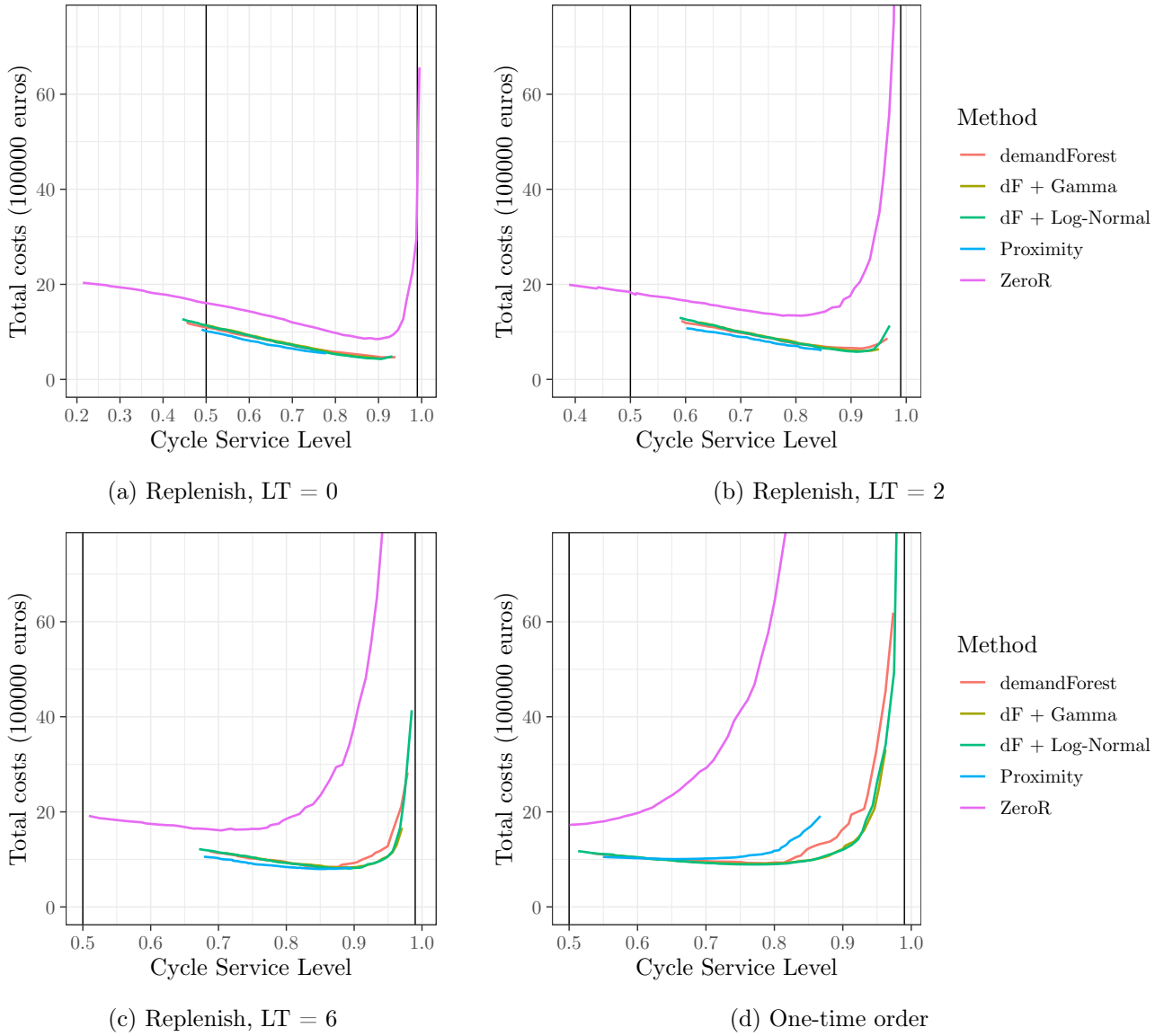
Company B

Interestingly, for the data set of company B, the Proximity method is again performing well. It is the method with the lowest costs for the replenishment (see Figure 5.17a to 5.17c). Unfortunately, this method does not achieve higher service levels than 0.78 for the lead time of zero weeks to a maximum of 0.90 at the case with lead times of six weeks. Also for the case with the one-time order, the Proximity method is comparable to the demandForest methods, but it is not able to achieve a service level higher than 0.87. For three of the four cases, the demandForest methods are able to achieve the lowest costs, see Table 5.7. Nevertheless, for the lead time of six weeks, the Proximity method is better. Overall, the Proximity method and demandForest methods are very comparable, while the demandForest methods obtain a wider and more reliable range of service levels.

For the methods with the lowest costs per case in Table 5.7, the holding costs, excess holding costs, and lost value seems to decrease with shorter lead times. Moreover, the order costs have a smaller impact on the total costs with this data set compared to company A. Therefore, the replenishment cases result in lower costs than the one-time order case.

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	dF + Log-Normal	dF + Log-Normal	Proximity	dF + Log-Normal
Quantile	0.98	0.94	0.92	0.77
CSL	0.906	0.910	0.846	0.766
Total costs (€)	430,809	580,707	798,536	893,481
Order costs (€)	72,875	85,850	61,675	25
Holding costs (€)	49,041	52,754	76,767	94,949
Excess holding costs (€)	103,628	125,382	169,709	172,323
Lost value (€)	205,265	316,721	490,385	626,184

Table 5.7: Results of the methods with lowest costs per case for company B



(a) Replenish, LT = 0

(b) Replenish, LT = 2

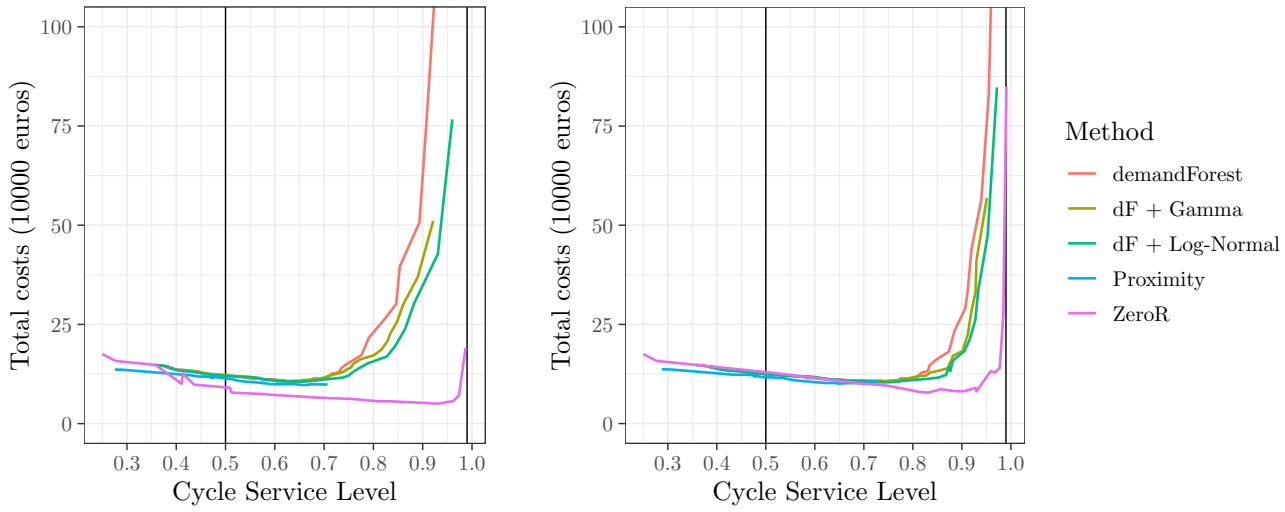
(c) Replenish, LT = 6

(d) One-time order

Figure 5.17: Cycle Service Levels and inventory costs for company B

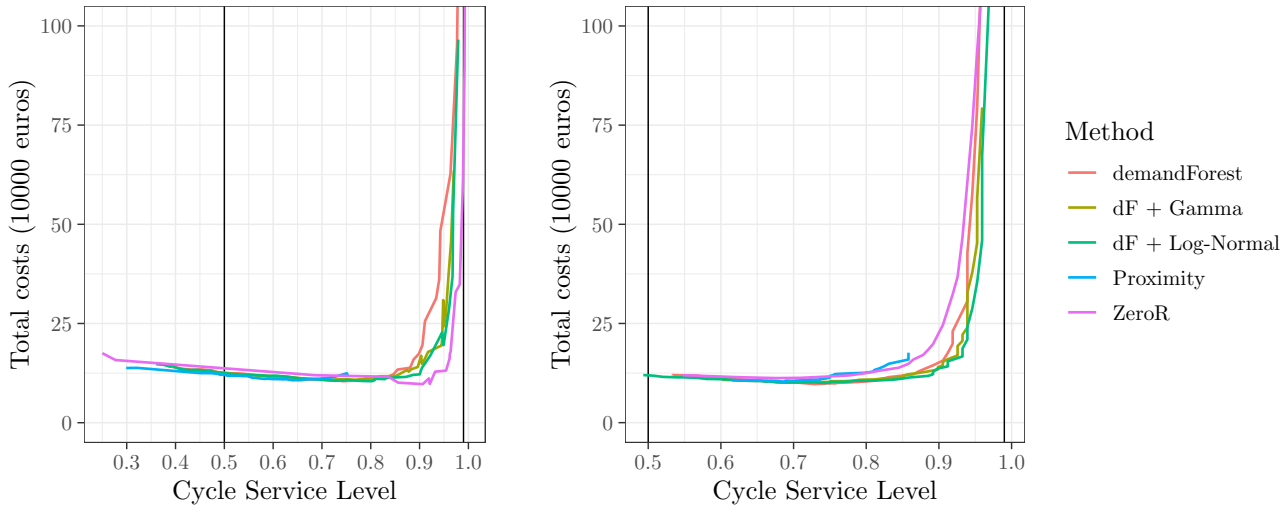
Company C

For company C, the ZeroR method performs surprisingly good for all cases, much better than for the other companies. It provides the lowest inventory costs for all replenishment cases (see Table 5.8) at relative high quantiles and service levels. For lower service levels at these replenishment cases, the Proximity method results in the lowest costs (see Figure 5.18a to 5.18c). In Figure 5.18c with the lead time of six weeks, the demandForest methods perform better for some service levels around the 0.80. Only for the one-time order case, Figure 5.18d, the demandForest methods are better. Regarding the lowest costs and corresponding service levels in Table 5.8, when the lead time decreases, the quantile and service level with the lowest costs tend to increase, while the costs decrease. These costs mainly consist of the excess holding costs and lost value. The replenishments enable to apply a higher service level, which decreases the lost value, while also decreasing the (excess) stocks. The (excess) stocks can be decreased due to ordering for smaller time windows. Similarly to company B, the overall lowest costs are obtained at the case with the shortest lead times.



(a) Replenish, LT = 0

(b) Replenish, LT = 2



(c) Replenish, LT = 6

(d) One-time order

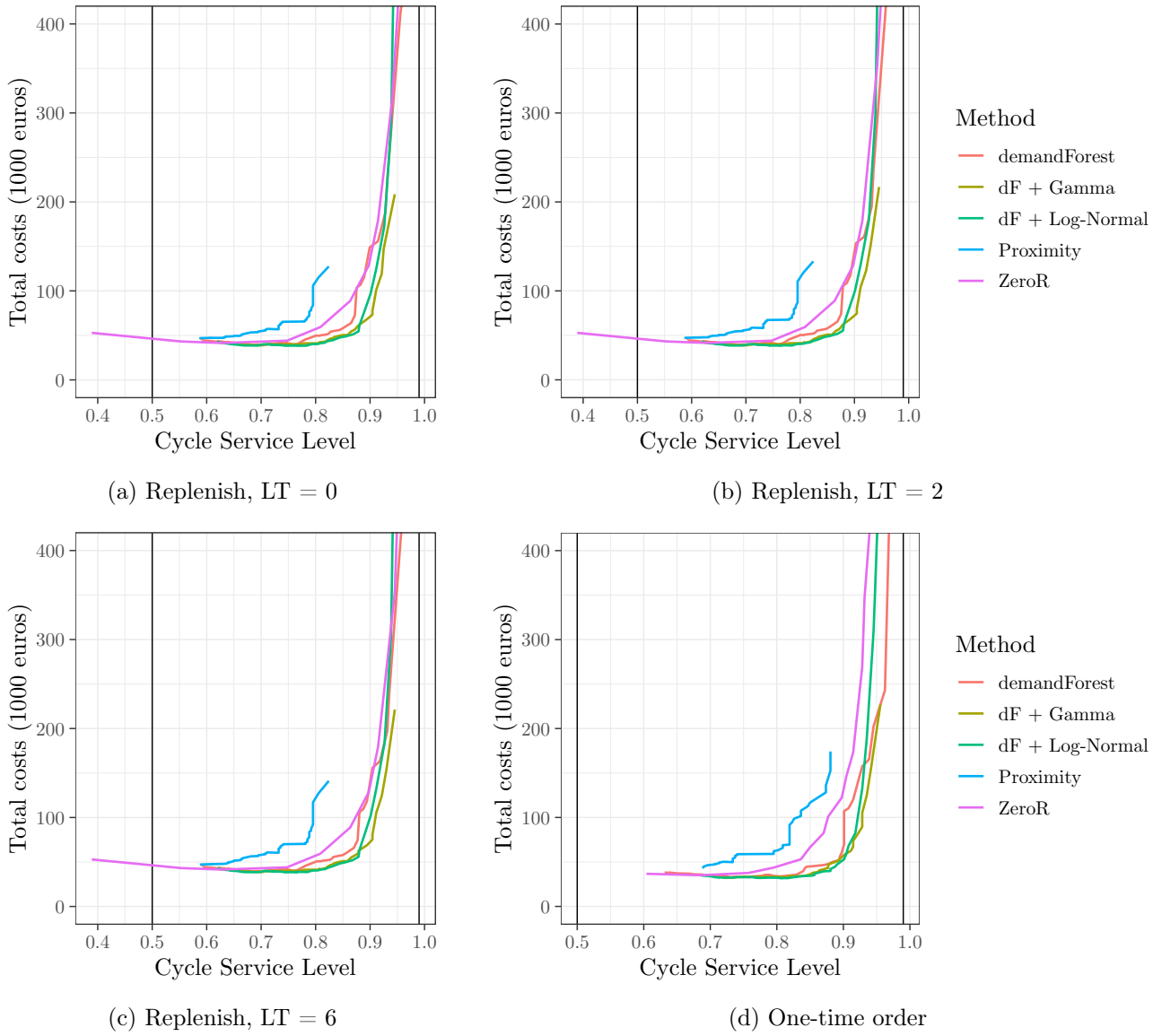
Figure 5.18: Cycle Service Levels and inventory costs for company C

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	ZeroR	ZeroR	ZeroR	demandForest
Quantile	0.96	0.88	0.84	0.73
CSL	0.931	0.831	0.907	0.730
Total costs (€)	50,180	77,904	97,119	97,284
Order costs (€)	9,425	10,000	7,825	25
Holding costs (€)	6,750	4,962	9,676	7,977
Excess holding costs (€)	11,975	7,450	26,710	12,623
Lost value (€)	22,030	55,493	52,908	76,659

Table 5.8: Results of the methods with lowest costs per case for company C

Company D

As can be seen in Figure 5.19, the results are for all cases for company D remarkably similar. Over the range from about 0.75 to 0.90, the demandForest methods result in the lowest costs. The Proximity method only covers a small range of service levels, with higher costs than the other methods. For CSLs above the 0.90, the ZeroR method and demandForest methods increase significantly. The demandForest methods are the methods with the lowest costs, with demandForest as the overall lowest for the replenishment cases at quantile 0.79, with CSLs around 0.77 (see Table 5.9). Regarding the underlying costs, these are also very similar for each case. This may be explained due to the fact that 68.2% of the data only has sales in the first week, as discussed in Section 2.2. Since the one-time order case has less order costs, it obtaines the overall lowest costs.



(a) Replenish, LT = 0

(b) Replenish, LT = 2

(c) Replenish, LT = 6

(d) One-time order

Figure 5.19: Cycle Service Levels and inventory costs for company D

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	dF + Log-Normal	dF + Log-Normal	dF + Log-Normal	dF + Log-Normal
Quantile	0.79	0.79	0.79	0.76
CSL	0.766	0.766	0.766	0.805
Total costs (€)	38,294	38,409	38,412	31,666
Order costs (€)	7,400	7,400	7,400	25
Holding costs (€)	2,853	2,878	2,887	3,733
Excess holding costs (€)	6,045	6,203	6,248	5,597
Lost value (€)	21,996	21,929	21,877	22,311

Table 5.9: Results of the methods with lowest costs per case for company D

Company E

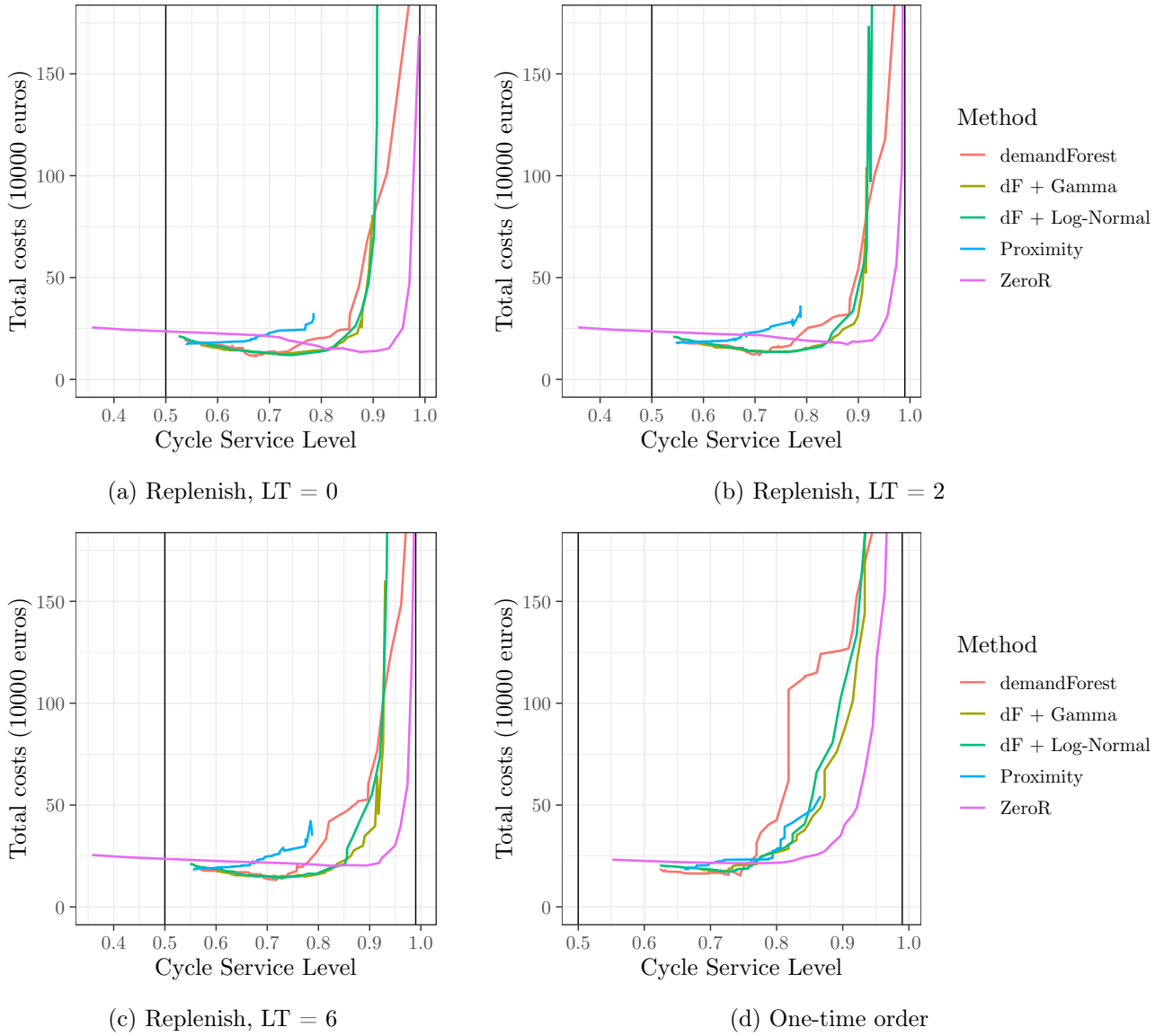
Less than the results of company D, but the costs for the different cases of company E, illustrated in Figure 5.20, are very similar. Presumably because 50.3% of the products only have sales in the first week, as discussed in Section 2.2. This percentage is somewhat lower than the 68.2% of company D.

In all cases, the regular demandForest results in the lowest costs with quantiles of 0.72/0.73 and service levels between 0.67 and 0.75 (see Table 5.10). The lost value of these results is the most significant part of the total inventory costs. The demandForest methods are overall best methods for the service levels between 0.55 and 0.77/0.85. However, the ZeroR method scores a lot better for the higher CSLs in each case. The ZeroR method obtains the lowest costs for CSLs of all cases above the 0.85.

The proximity method covers a small range of service levels with costs similar or higher than the others. Only for some CSLs around 0.55, the Proximity is the lowest. Comparing the demandForest methods, we observe that the costs of the regular demandForest start increasing at a lower CSL than the extended methods. Nevertheless, it increases slower and result in lower costs for the highest service levels. All three demandForest increase significantly at CSLs above 0.90, while ZeroR tends to increase more after 0.95.

	Replenish, LT=0	Replenish, LT=2	Replenish, LT=6	One-time order
Method	demandForest	demandForest	demandForest	demandForest
Quantile	0.73	0.72	0.72	0.72
CSL	0.674	0.708	0.717	0.745
Total costs (€)	114,022	118,849	131,514	153,978
Order costs (€)	6,550	5,750	5,175	25
Holding costs (€)	5,121	6,042	8,114	14,606
Excess holding costs (€)	18,731	15,187	25,013	57,905
Lost value (€)	83,619	91,870	93,212	81,442

Table 5.10: Results of the methods with lowest costs per case for company E



(a) Replenish, LT = 0

(b) Replenish, LT = 2

(c) Replenish, LT = 6

(d) One-time order

Figure 5.20: Cycle Service Levels and inventory costs for company E

Overall observations

Regarding the overall observations, we see that the demandForest are the best methods for the one-time order cases. Nevertheless, none of the different demandForest methods in clearly better than the other. The main difference between the demandForest methods that can be identified is that the demandForest + Gamma method generally increases less significant for the highest quantiles. For these one-time order cases, no profile was necessary and only the predictions from the QRF algorithms were used.

Similar to the consistency within the cases (Section 5.4.1), demandForest performed less compared to the benchmark methods for the cases with shorter lead times. Nevertheless, for the synthetic data set and the data of company B and D, the demandForest methods were overall the best. For the results of company A, the Proximity method was usually better. For company C and E, the ZeroR method often achieved better results, especially for higher service levels. These results are not surprising for company C, because the ZeroR method also provided accurate forecasting results in Section 5.3. Furthermore, the total costs were often the lowest for the replenishment case with a lead time of zero. This suggests that anticipating the demand is

useful. With decreasing lead times, holding costs, excess holding costs and the lost value usually decreased, while order costs logically increased.

Although the Proximity method was less reliable in terms of achieving the target service levels (see Section 5.4.1), it was often competitive in terms of costs. Especially for company A were the costs often lower than the other methods. For the synthetic, company A and B, the differences between the methods became smaller when the lead time decreased. This means that the method of forecasting becomes less important for the actual inventory performance. This makes sense, since one can anticipate quicker to potential deviations between the forecast and the actual demand, and the deviations are less severe than the cases with longer lead times. Nevertheless, one should keep attention to the order costs when anticipating with short lead times, to prevent that the costs significantly increase due to a lot of orders.

5.5 Conclusion

In this section, we applied the different methods on a synthetic data set and five data sets from industry partners. For these data sets, we analysed the predictive performance and forecasting quality of the individual algorithms, the combined methods and we evaluated the inventory performance. With all these steps applied at different data sets, we aimed to obtain a proper overview of the quality of the forecast and performance of the methods in practice.

In the first step, we trained the RF and QRF algorithms to find the best value for *mtry* on each individual data set. Additionally, we analysed the feature importance. This importance analysis provided useful information about which product characteristics have predictive value for the profile and demand of a new product. For the synthetic data set, it provided the expected results. The brand and category were important for the profile, whereas the colour and price were important for the demand. As could be expected, the industry data sets usually showed less dominant importance. The overall main predictive features were the supplier and the category of a product. More specifically, the most importance feature for the data set of company C was the brand collection. This brand collection indicates the specific product serie of a brand and was important for predicting both the profile and the demand.

After training the demand, we could use the algorithms for predicting new products. We found that the predictions of the profile were not very accurate in some cases. Predicting the profile was only better than the average profile for the algorithms with a kappa greater than 0.4. Hence, when demandForest might be applied to other data sets, it may not be beneficial to determine profiles if the kappa of the prediction is below 0.4. For predicting the demand, we observed that extended demandForest methods provided improved results. The demandForest + Gamma made improvements regarding the RMSE. The PICP and PINAW seemed to improve for both extensions, where especially the demandForest + Log-Normal improved the regular demandForest results. The benchmark methods, both the Proximity and the ZeroR, showed worse results. The RMSEs were larger, the PICP of the Proximity method varied between 78.4% and 96.8%, and the PINAWs are generally larger. Only the PINAWs of the Proximity method were the lowest for three data sets, but at the costs of too low PICPs. In the interval plots we also observed that the intervals of the Proximity method were often too wide or too too narrow.

By combining the profiles and demand predictions, we assembled a forecast for the introduction period. By analysing the RMSE of the forecast and the cumulative forecast, we found that the demandForest methods overall provided a better forecasting quality than the benchmark methods. There were some exceptions, ZeroR had one of the lowest RMSEs of the regular forecasts for company C and D. These companies were also the companies for which the profile prediction was worse than the average profile.

With these promising results, we analysed the inventory cases. For these inventory cases, we evaluated not only the predictions or 90% prediction intervals, but also evaluated the quantiles between 0.50 and 0.99 with a step size of 0.01. The cases we defined were three cases with replenishment lead times of zero, two and six weeks. In the fourth case, only one batch of products was ordered at the beginning of the introduction.

In the first inventory analysis, we compared the quantiles with the Cycle Service Levels. For the one-time order cases, the CSLs were reliable, especially for the demandForest methods and ZeroR. For the replenishment cases, the CSLs were usually higher than the quantiles. The demandForest methods were all quite comparable, with the demandForest + Gamma as the method with a slight upward bias. This bias was for some data sets an advantage, and sometimes not. Compared to the benchmark methods, the demandForest methods were more reliable for the different companies. The data sets of company C, D and E showed that the ZeroR method provides much lower quantiles than it should have provided for quantiles below 0.80. The Proximity method on the other hand provided different results for each data set. For the synthetic data, the service levels were too high, and company C had too low service levels. For the other data sets both too high and too low CSLs were observed. Hence, the Proximity method is the most unreliable method regarding the service levels.

Although the service levels were unreliable, the inventory costs of the Proximity method were quite good and often comparable to the demandForest methods. The ZeroR method provided also competitive results. For the products of company C and for the highest service levels of company E, ZeroR provided the lowest costs. Nevertheless, in most other cases the demandForest methods provided the lowest costs. Especially for the one-time order cases. Hence, demandForest is most useful in cases with a one-time order or longer lead times. In the replenishment cases, when the profiles were employed, the methods were performing less. Nevertheless, they still provided the overall most robust results, while the benchmark methods (i.e., the Proximity method and ZeroR) varied more in their performance, which sometimes resulted higher costs.

That the costs of the Proximity method were often comparable to the demandForests methods, despite its lower forecast accuracy, can be explained by multiple differences between forecasting and inventory management and the differences between the demandForest method and the Proximity method. In the case of forecasting, the RMSE penalises each deviation from the actual demand. However, this changes for inventory costs. A higher forecast/quantile usually increases holding costs and decreases ordering costs and lost value. Furthermore, the excess costs are only determined at the end of the introduction period. These are likely to increase with higher forecasts/quantiles. Hence, these costs are more balanced than the RMSE, which penalises each deviation. Nevertheless, one should expect that more accurate forecasts (e.g., the demandForest methods) decrease the costs compared to less accurate forecasts (e.g., the Proximity method). However, we use the quantiles instead of the mean in the inventory case. For the Proximity method is quantile 0.50 equal to the mean, but for the demandForest methods are the means larger than the median, since the distributions are right-skewed. These quantiles are less dependent on the mean than the quantiles of the proximity method. Hence, a good forecast accuracy cannot also be interpreted as a good inventory performance. The estimated quantiles of demandForest might not yet be optimal based on cost, yet they provide more reliable service levels than the Proximity method with comparable costs.

To conclude, despite the fact that the benchmark methods are in some cases and data sets better for predicting the demand and providing good service levels at low costs, the overall best methods remain the demandForest methods. These methods provide the highest forecasting quality, reliable service levels and one of the lowest inventory costs. Large differences between the demandForest methods were not observed. Considering the forecasting quality, the demandForest + Gamma was the most competitive. For the inventory costs, mostly the lowest costs were provided by the regular demandForest, thereafter by the demandForest + Log-Normal.

6 | Conclusion and recommendations

In Section 1.4, we defined research questions to obtain a satisfactory result towards the research objective. In the subsequent chapters, we answered these questions. In Chapter 2, we explored the data of five industry partners and found certain patterns and relations in the data of new products. We reviewed current literature about new product forecasting, machine learning algorithms and performance metrics in Chapter 3. With this knowledge, we designed our forecasting method in Chapter 4. In that chapter, we also defined two benchmark methods and proposed a synthetic data set to evaluate the performance of the proposed method. Finally, we analysed the performance of the methods in Chapter 5, for both forecasting and inventory management.

In this chapter, we discuss the most important findings and implications of this research. In Section 6.1, we discuss the most important findings, contributions to literature, and limitations of this research. In Section 6.2, we elaborate about the practical implications of this research and provide our recommendations for Slimstock. Finally, in Section 6.3 we provide directions for future research.

6.1 Conclusion

We developed a novel approach called demandForest that provides pre-launch forecasts for the first four months of demand of new products. The Random Forest and Quantile Regression Forest algorithms utilise product characteristics of new and existing products to generate a profile and the total demand of these new products during the first four months. These forecasts are based on the historical demand of existing comparable products. In this way, it overcomes the challenge of new product forecasting: the lack of historical data. With the Quantile Regression Forest algorithm, DemandForest also provides prediction intervals and quantiles. These intervals and quantiles estimate the uncertainty of the demand and can be used in inventory management. Furthermore, by fitting the Gamma and Log-Normal distributions to the empirical distributions of the Quantile Regression Forest algorithms, we extended demandForest and tried to improve the estimated quantiles.

To evaluate the forecasts and applicability to inventory management of demandForest, we defined two benchmark methods. The first benchmark method, ZeroR, simply took the average or quantiles of all existing products. The second benchmark, Proximity, is defined with the aim to imitate the current forecasts and decisions of supply chain planners. With the proximity measure from the QRF algorithm, it determines the most similar product and uses this value as total demand. For the profile, Proximity uses the average profile of the existing products. The demandForest methods and the benchmark methods were evaluated on different data sets, namely a synthetic data set and data of industry partners from retail, e-commerce, wholesale and both B2B and B2C markets. In this way, we assessed the quality and performance of the proposed method, and also the general applicability and robustness.

The demandForest methods provided overall the most accurate and robust results. With a few exceptions, the forecasts of demandForest had the highest quality. The extensions of the Log-Normal and Gamma distributions provided slightly better forecasts than the regular empirical distributions of the QRF algorithm of demandForest. The three demandForest methods provided comparable results for the inventory cases. Since the service levels of demandForest are based on the quantiles of the total demand, the results were most consistent for a one-time order. The results of the CSLs became less consistent for the replenishment cases, where the total demand was combined with the profiles. Hence, the applicability and reliability of the profiles remains arguable. Nevertheless, the CSLs of the demandForest methods were usually above the targeted service levels in the replenishment cases. Therefore, the profiles can be used in practice to guarantee certain service levels. Furthermore, they provided more consistent CSLs than the benchmark methods. ZeroR provided very accurate service levels, but not in all cases. The Proximity method resulted in too high service levels for some cases and data sets, and too low service levels for other cases.

Although the benchmark methods were less consistent in their service levels, they provided competitive results for the inventory costs. Both the Proximity method and ZeroR resulted in the lowest costs for several cases. Nevertheless, the demandForest methods were overall better. These provided the lowest costs in most of the cases, and otherwise not much more expensive than the benchmark methods. The extensions of fitting the Log-Normal and Gamma distribution to the empirical distributions of demandForest did not provide significantly better results. All in all, the demandForest method provides a higher forecasting quality, more reliable and consistent prediction intervals and service levels, and comparable or lower costs than the benchmark methods. DemandForest was especially performing well for the one-time order cases.

Besides these promising results of demandForest, it was also possible to extract additional information from the Random Forest and Quantile Regression Forest algorithms. With the feature importance, it was possible to determine the product characteristics which have the most predictive value in a data set. Furthermore, with the Proximity method, we used the most comparable product based on the proximity. With the proximity from the rf, it is also possible to present a supply chain planner with a top 5 most comparable products. Both the feature importance as a top 5 products can be valuable insights when evaluating a certain forecast in detail.

6.1.1 Scientific contribution

This research contributes to the field of new product forecasting in several ways. To the best of our knowledge, this research is the first example of Quantile Regression Forests in new product forecasting and the application to inventory management. In addition, we improved the prediction intervals and quantiles in several cases by fitting theoretical distributions to the quantiles of the QRF algorithm. As a side result, we employed the proximity matrix from the Random Forest algorithm to find the most comparable products. These most comparable products can also provide more insight into the algorithm, besides the analysis of the feature importance. Furthermore, we proposed a synthetic data set. This data set can be used for future research on this topic and can be used to compare current and new methods.

6.1.2 Limitations

Despite the promising results, this study and demandForest comes with limitations. We discuss limitations of the data used in this research, the methodology and the algorithms used.

First, we did not have the actual introduction date for all data sets. Hence, we assumed that the date of the first sold product was the introduction date. This resulted in the fact that,

except for the data of company A, we had at least one product sold in the first week. When defining the profiles, we found this effect in one of the two profiles of each data set. Moreover, since we used the first sale as introduction date, each product in the data set was sold at least once. Products which may never be sold were not considered.

Second, another limitation of the data occurred at for company A. For this data, the demand of numerous shops were aggregated. Hence, we did not consider differences between the shops. In reality, there may exist important differences that should be considered when disaggregating forecasts to the shop level.

Third, the forecasts of new products of all methods are only based on previously introduced products and a limited number of product characteristics. However, the actual demand may also depend on trends in the market (e.g., consumer expectation or competition), assortment management of a company (e.g., size of the product category, shelve space, or page rank in a web shop), marketing (e.g., commercials or promotions), and weather and seasonality (e.g., sunny weather, Christmas, etc.). Hence, a forecast based on product characteristics can support a supply chain planner with the initial inventory decisions, but does not cover all aspects of new product demand. Expert judgment and intuition of planners may still be necessary to anticipate to the factors besides the product characteristics.

Fourth, the inventory case has some limitations. Instead of analysing the products with their own specific review times and lead times, we applied four cases with fixed values of all products. Furthermore, the order costs, holding costs, excess holding costs, and lost value are based on estimations. In a real-world setting, these actual performances and costs differ per product and per company. Nevertheless, these fixed values and estimated provided sufficient into the methods and enabled comparability between different data sets.

Lastly, the Random Forest (RF) algorithms have some limitations. The range of forecasts of a RF algorithm is limited to the lower and upper limit of the training set. Moreover, since the final output of a RF algorithm is the average of all trees, it is difficult to forecast the demand of outliers such as the most successful new products. The forecasts of successful products are likely to be underestimated due to averaging the predictions of individual trees.

6.2 Recommendations

Regarding this study, we have several recommendations for Slimstock:

1. We advise Slimstock and its clients to start keeping track of actual introduction dates. We advice to use the first date at which the product is presented to customers for sale as actual introduction date. With these actual introduction dates, they can improve the historical demand data during the introduction period. It can prevent the bias in the data that each product is at least sold once in the first week, which we had to deal with in this research.

2. Furthermore, we recommend to save the historical forecasts and orders of new products. In that case, Slimstock can evaluate if demandForest also improves the actual forecasts of planners.

3. If possible, more product characteristics that are known before the introduction should be saved in the data bases of the software of Slimstock. Several characteristics that might have been important for the choices of consumers were not available for this research, for example colours, dimensions and materials. With this additional data, forecasts may be improved.

4. Since demandForest performed especially well for the one-time order cases, we advice Slimstock to decide whether they want to use the combined methods for the full four month forecasts or only use it for the first order of a new product (without profiles). In the latter case, the method should be slightly adjusted such that the forecast period can be adjusted to the number of weeks a first order should cover. When demand data becomes available after the first inventory cycle, the current statistical methods can follow up the first predictions of demandForest.

5. We advise to implement the demandForest method in SQL Server with Machine Learning Services as a pilot, such that the data collection and calculations can be executed in-database at clients. With such a pilot, demandForest can be tried out by multiple clients. When this pilot succeeds, the methods can be fully integrated into the software of Slimstock.

6. Before implementing demandForest at a client, we recommend to always perform an analysis with training, tuning and testing the algorithms before employing them for actual forecasts. In this case, there is more insight into the expected performance of the forecasts.

7. Besides providing planners of clients with full forecasts and quantile predictions, Slimstock can also provide them with additional useful insights.

   (a) With the proximity measure from the Random Forest algorithms, it is possible to present the top 5 comparable products. Planners can use this as reference material if they want to adjust the forecasts based on their own insights. It also provides them with evidence for a certain forecast, which can be used during S&OP meetings. Since planners typically have to find comparable products manually, this can save them time.

   (b) The feature importance of the complete data set is also a useful insight for planners. It shows the significance of certain product characteristics on the new product demand.

6.3 Directions for future research

This research did not only found answers for the research questions, but also raised new questions. In this section, we provide directed for future research to improve demandForest.

The demandForest method only generates pre-launch forecasts. However, when actual demand data becomes available, one can update the forecasts or switch to more traditional statistical methods for the forecasts. Future research can build upon demandForest and should investigate ways to update the forecasts such that they become more accurate when new data becomes available.

The combination of the total amount of demand and the profile did not always resulted in the targeted service levels. Hence, other approaches for estimating the uncertainty and improving the reliability of the service levels deserve further exploration. The quantiles might be determined otherwise than from the total demand. Maybe it is better to calculate independent predictions for each week. It may also be beneficial to exploit the variability within a profile. Additionally, it might be valuable to predict the Average inter-Demand Interval (ADI) and Squared Coefficient of Variation (CV^2) and build forecasts based on this information.

In this research, we included all new products introduced by a company into the training phase of the algorithms. However, forecasts may be improved by segmenting specific groups of products based on, for example, categories, prices or an ABC-classification. The performance of forecasts might be improved by using only the segmented input data of such a group of products.

In this study, only product characteristics are used. Future research can consider, for example, market trends, competition, marketing, seasonality, or the weather. This data might be more challenging to obtain, but forecasts may improve when considering a wider range of predictive factors.

We mainly focused on Random Forest algorithms, since these seemed the most suitable for our research. Nevertheless, other methods might be able to achieve better results. Therefore, future work should focus on evaluating other methods such as Artificial Neural Networks, Gradient Boosting, and Support Vector Machines.

Bibliography

Albers, S. (2004). Forecasting the diffusion of an innovation prior to launch. *Albers, S.(Hg.): Cross-functional Innovation Management-Perspectives from different disciplines. Wiesbaden*, 243–258.

Ali, J., Khan, R., Ahmad, N. & Maqsood, I. (2012). Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, *9*(5), 272.

Alpaydin, E. (2010). *Introduction to machine learning*. The MIT Press.

Amasyali, M. F. & Ersoy, O. K. (2009). A study of meta learning for regression. *ECE Technical Reports*, 386.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., PéRez, J. M. & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256.

Assmus, G. (1984). New product forecasting. *Journal of Forecasting*, *3*(2), 121–138.

Baardman, L., Levin, I., Perakis, G. & Singhvi, D. (2018). Leveraging comparables for new product sales forecasting. *Production and Operations Management*, *27*(12), 2340–2343.

Basallo-Triana, M. J., Rodrıguez-Sarasty, J. A. & Benitez-Restrepo, H. D. (2017). Analogue-based demand forecasting of short life-cycle products: A regression approach and a comprehensive assessment. *International Journal of Production Research*, *55*(8), 2336–2350.

Bass, F. M. (1969). A new product growth for model consumer durables. *Management science*, *15*(5), 215–227.

Bass, F. M. (2004). Comments on "a new product growth for model consumer durables the bass model". *Management science*, *50*(12-supplement), 1833–1840.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth and Brooks.

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

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

Breiman, L. (2003). Manual - setting up, using, and understanding random forests v4.0. Retrieved June 23, 2019, from https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf

Bucher, D. & Meissner, J. (2011). Configuring single-echelon systems using demand categorization. In *Service parts management* (pp. 203–219). Springer.

Caliński, T. & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Chaudhuri, P., Loh, W.-Y. et al. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, *8*(5), 561–576.

Cobb, B. R., Rumı, R. & Salmerón, A. (2013). Inventory management with log-normal demand per unit time. *Computers & Operations Research*, *40*(7), 1842–1851.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Costantino, F., Di Gravio, G., Patriarca, R. & Petrella, L. (2018). Spare parts management for irregular demand items. *Omega*, *81*, 57–66.

De Brabanter, K., De Brabanter, J., Suykens, J. A. & De Moor, B. (2010). Approximate confidence and prediction intervals for least squares support vector regression. *IEEE Transactions on Neural Networks*, *22*(1), 110–120.

Delen, D. (2011). Predicting student attrition with data mining methods. *Journal of College Student Retention: Research, Theory & Practice*, *13*(1), 17–35.

Delignette-Muller, M. L., Dutang, C. et al. (2015). Fitdistrplus: An r package for fitting distributions. *Journal of Statistical Software*, *64*(4), 1–34.

Espinoza, M., Joye, C., Belmans, R. & De Moor, B. (2005). Short-term load forecasting, profile identification, and customer segmentation: A methodology based on periodic time series. *IEEE Transactions on Power Systems*, *20*(3), 1622–1630.

Fawagreh, K., Gaber, M. M. & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, *2*(1), 602–609.

Gholami, A. & Mirzazadeh, A. (2018). An inventory model with controllable lead time and ordering cost, log-normal-distributed demand, and gamma-distributed available capacity. *Cogent Business & Management*, *5*(1), 1469182.

Gneiting, T. & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378.

Goodwin, P., Dyussekeneva, K. & Meeran, S. (2013). The use of analogies in forecasting the annual sales of new electronics products. *IMA Journal of Management Mathematics*, *24*(4), 407–422.

Goodwin, P., Meeran, S. & Dyussekeneva, K. (2014). The challenges of pre-launch forecasting of adoption time series for new durable products. *International Journal of Forecasting*, *30*(4), 1082–1097.

Green, K. C. & Armstrong, J. S. (2007). Structured analogies for forecasting. *International Journal of Forecasting*, *23*(3), 365–376.

Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The elements of statistical learning*. Springer series in statistics New York.

Hibon, M., Kourentzes, N. & Crone, S. F. (2013). New product forecasting and inventory planning using time series clustering. *The 33rd Annual international Symposium on Forecasting, Seoul*.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278–282). IEEE.

Huber, J., Gossmann, A. & Stuckenschmidt, H. (2017). Cluster-based hierarchical demand forecasting for perishable goods. *Expert systems with applications*, *76*, 140–151.

Hwang, C., Hong, D. H. & Seok, K. H. (2006). Support vector interval regression machine for crisp input and output data. *Fuzzy Sets and Systems*, *157*(8), 1114–1125.

Hwang, J. G. & Ding, A. A. (1997). Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, *92*(438), 748–757.

Hyndman, R. J. et al. (2006). Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, *4*(4), 43–46.

Hyndman, R. J. & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, *31*(8), 651–666.

Kabaila, P. (1999). The relevance property for prediction intervals. *Journal of Time Series Analysis*, *20*(6), 655–662.

Kahn, K. B. (2002). An exploratory investigation of new product forecasting practices. *Journal of Product Innovation Management*, *19*(2), 133–143.

Kahn, K. B. (2006). *New product forecasting: An applied approach*. New York: M.E. Sharpe, Inc.

Kahn, K. B. (2014). Solving the problems of new product forecasting. *Business Horizons*, *57*(5), 607–615.

Kahneman, D. & Tversky, A. (1977). *Intuitive prediction: Biases and corrective procedures*. Decision Research: A branch of perceptronics.

Khosravi, A., Nahavandi, S. & Creighton, D. (2010). Construction of optimal prediction intervals for load forecasting problems. *IEEE Transactions on Power Systems*, *25*(3), 1496–1503.

Khosravi, A., Nahavandi, S. & Creighton, D. (2013). A neural network-garch-based method for construction of prediction intervals. *Electric Power Systems Research*, *96*, 185–193.

Khosravi, A., Nahavandi, S., Creighton, D. & Atiya, A. F. (2010). Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE transactions on neural networks*, *22*(3), 337–346.

Khosravi, A., Nahavandi, S., Creighton, D. & Atiya, A. F. (2011). Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on neural networks*, *22*(9), 1341–1356.

Koenker, R. & Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Lahouar, A. & Slama, J. B. H. (2017). Hour-ahead wind power forecast based on random forests. *Renewable energy*, *109*, 529–541.

Lee, H., Kim, S. G., Park, H.-w. & Kang, P. (2014). Pre-launch new product demand forecasting using the bass model: A statistical and machine learning-based approach. *Technological Forecasting and Social Change*, *86*, 49–64.

Lee, J., Boatwright, P. & Kamakura, W. A. (2003). A bayesian model for prelaunch sales forecasting of recorded music. *Management Science*, *49*(2), 179–196.

Lee, W. Y., Goodwin, P., Fildes, R., Nikolopoulos, K. & Lawrence, M. (2007). Providing support for the use of analogies in demand forecasting tasks. *International Journal of Forecasting*, *23*(3), 377–390.

Loureiro, A., Miguéis, V. & da Silva, L. F. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, *114*, 81–93.

Lu, C.-J. & Kao, L.-J. (2016). A clustering-based sales forecasting scheme by using extreme learning machine and ensembling linkage methods with applications to computer server. *Engineering Applications of Artificial Intelligence*, *55*, 231–238.

Lu, C.-J. & Wang, Y.-W. (2010). Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting. *International Journal of Production Economics*, *128*(2), 603–613.

Mahajan, V. & Peterson, R. A. (1978). Innovation diffusion in a dynamic potential adopter population. *Management Science*, *24*(15), 1589–1597.

Mas Machuca, M., Sainz Comas, M. & Martınez Costa, C. (2014). A review of forecasting models for new products. *Intangible capital*, *10*(1), 1–25.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983–999.

Milligan, G. W. & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*(2), 159–179.

Nagy, G. I., Barta, G., Kazi, S., Borbély, G. & Simon, G. (2016). Gefcom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach. *International Journal of Forecasting*, *32*(3), 1087–1093.

Namit, K. & Chen, J. (1999). Solutions to the< q, r> inventory model for gamma lead-time demand. *International Journal of Physical Distribution & Logistics Management*, *29*(2), 138–154.

Neelamegham, R. & Chintagunta, P. (1999). A bayesian model to forecast new product performance in domestic and international markets. *Marketing Science*, *18*(2), 115–136.

Nenes, G., Panagiotidou, S. & Tagaras, G. (2010). Inventory management of multiple items with irregular demand: A case study. *European Journal of Operational Research*, *205*(2), 313–324.

Olden, J. D., Joy, M. K. & Death, R. G. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, *178*(3-4), 389–397.

Oshiro, T. M., Perez, P. S. & Baranauskas, J. A. (2012). How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition* (pp. 154–168). Springer.

Pedro, H. T., Coimbra, C. F., David, M. & Lauret, P. (2018). Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts. *Renewable Energy*, *123*, 191–203.

Probst, P. & Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, *18*, 181–1.

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

R Core Team. (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from http://www.R-project.org/

Ramaekers, K., Janssens, G. K. et al. (2008). On the choice of a demand distribution for inventory management models. *European Journal of Industrial Engineering*, *2*(4), 479–491.

Rogers, E. M. (1962). *Diffusion of innovations*. New York: The Free Press.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Shrivastava, N. A., Khosravi, A. & Panigrahi, B. K. (2015). Prediction interval estimation of electricity prices using pso-tuned support vector machines. *IEEE Transactions on Industrial Informatics*, *11*(2), 322–331.

Silver, E. A., Pyke, D. F. & Thomas, D. J. (2017). *Inventory and production management in supply chains*. CRC Press.

SQL Server Machine Learning Services. (2019). Retrieved August 23, 2019, from https://docs.microsoft.com/en-us/sql/advanced-analytics/what-is-sql-server-machine-learning?view=sql-server-2017

Starkweather, J. & Moske, A. K. (2011). Multinomial logistic regression. *Consulted page at September 10th: http://www. unt. edu/rss/class/Jon/Benchmarks/MLR_ JDS_ Aug2011. pdf*, *29*, 2825–2830.

Sun, Z.-L., Choi, T.-M., Au, K.-F. & Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, *46*(1), 411–419.

Suykens, J. A. (2003). *Advances in learning theory: Methods, models, and applications*. IOS Press.

Syntetos, A. A., Boylan, J. E. & Croston, J. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, *56*(5), 495–503.

Szozda, N. (2010). Analogous forecasting of products with a short life cycle. *Decision Making in Manufacturing and Services*, *4*(1-2), 71–85.

Taillardat, M., Mestre, O., Zamo, M. & Naveau, P. (2016). Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Monthly Weather Review*, *144*(6), 2375–2393.

Tehrani, A. F. & Ahrens, D. (2016). Enhanced predictive models for purchasing in the fashion field by using kernel machine regression equipped with ordinal logistic regression. *Journal of Retailing and Consumer Services*, *32*, 131–138.

Thomas, R. J. (1985). Estimating market growth for new products: An analogical diffusion model approach. *Journal of Product Innovation Management*, *2*(1), 45–55.

Thomassey, S. & Fiordaliso, A. (2006). A hybrid sales forecasting system based on clustering and decision trees. *Decision Support Systems*, *42*(1), 408–421.

Thomassey, S. & Happiette, M. (2007). A neural clustering and classification system for sales forecasting of new apparel items. *Applied Soft Computing*, *7*(4), 1177–1187.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988–999.

Vaysse, K. & Lagacherie, P. (2017). Using quantile regression forest to estimate uncertainty of digital soil mapping products. *Geoderma*, *291*, 55–64.

Venables, W. N. & Ripley, B. D. (2013). *Modern applied statistics with s-plus*. Springer Science & Business Media.

Voulgaridou, D., Kirytopoulos, K. & Leopoulos, V. (2009). An analytic network process approach for sales forecasting. *Operational Research*, *9*(1), 35–53.

Wright, M. J. & Stern, P. (2015). Forecasting new product trial with analogous series. *Journal of Business Research*, *68*(8), 1732–1738.

Wright, M. N. & Ziegler, A. (2015). Ranger: A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv:1508.04409*.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... Philip, S. Y. et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1–37.

Zamo, M., Mestre, O., Arbogast, P. & Pannekoucke, O. (2014). A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production, part i: Deterministic forecast of hourly production. *Solar Energy*, *105*, 792–803.

Zhang, G. P. (2009). Neural networks for data mining. In *Data mining and knowledge discovery handbook* (pp. 419–444). Springer.

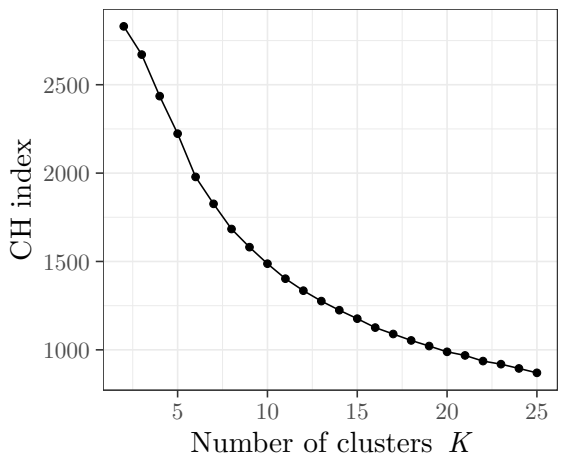
Zhao, J. H., Dong, Z. Y., Xu, Z. & Wong, K. P. (2008). A statistical approach for interval forecasting of the electricity price. *IEEE Transactions on Power Systems*, *23*(2), 267–276.

A | Appendix Data Sets

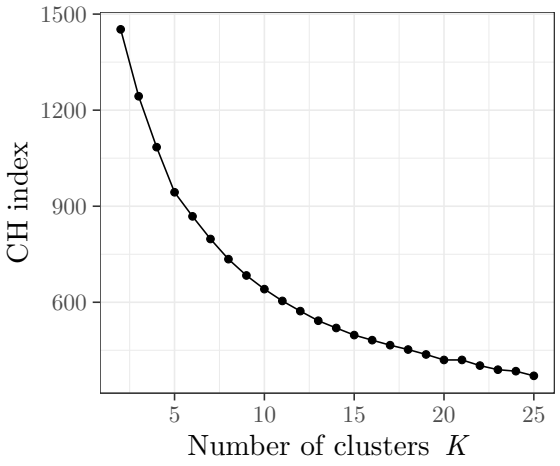
This appendix is confidential.

B | Appendix Clustering

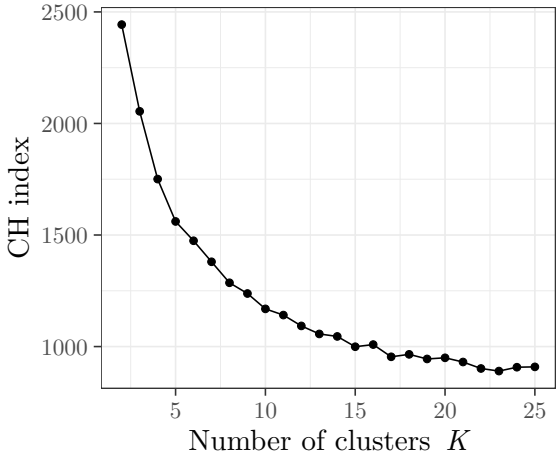
In Section 4.4.2, we described the clustering of demand profiles for all companies. In this appendix, we visualise the values of the CH-index of the k-means clustering for the companies not shown in Section 4.4.2. Figure B.1 visualises the CH-indices for company B (see Figure B.1a), company C (see Figure B.1b), and company D (see Figure B.1c). For all companies, the optimal number of clusters is 2.



(a) Company B



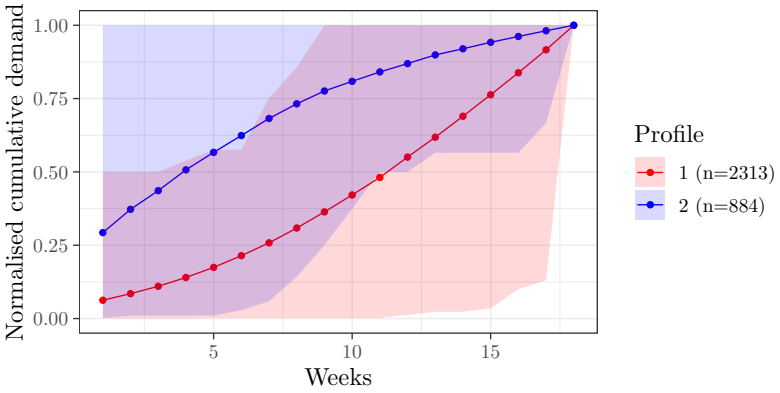
(b) Company C



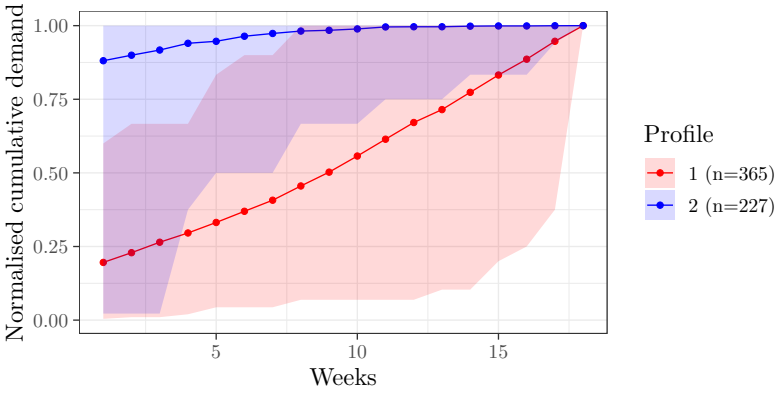
(c) Company D

Figure B.1: Optimal number of clusters

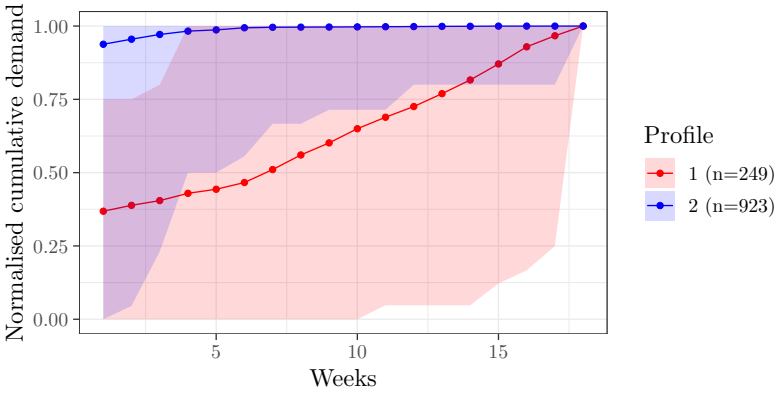
When determining the cluster centres with 2 clusters, we obtain the clusters for the companies. The cluster centres, or profiles, and ranges for company B, C, and D are displayed in Figure B.2a, B.2b, and B.2c respectively. Similar to the results of company A and E, each company has one concave increasing profile, and one rather linear or convex increasing profile. Company B is similar to company A, whereas the high demand in the first week of company C and D more similar is to company E.



(a) Company B



(b) Company C



(c) Company D

Figure B.2: Cluster centres with $k = 2$

C | Appendix Synthetic Data Set

In Section 4.4.4, we proposed a synthetic data set. In this appendix, we provide more details about the synthetic data set and provide additional tables and a figure in the next pages. As already discussed, the price and colours are related to the demand, whereas the category and brand are related to the profile. For all products in this synthetic data set, we first randomly assigned the demand and a profile. The total demand is generated randomly with the Gamma distribution with $\alpha = 2$ and $\beta = 150$. In this case, the average demand will be 300. The profiles have 18 demand points. The profiles are, arbitrarily, one with a 10% exponential increase per demand point, one with a 10% exponential decrease per demand point, and one stable profile. The demand for each demand point is the profile multiplied by the total demand and Normal distributed noise with a coefficient of variation of 0.25 is added to each demand point.

Afterwards, we assigned the characteristics to these products. The relation between the price and demand is: $price = 2000/demand$. This relation is inversely proportional, where a low demand corresponds with a high price, and a high demand with a low price. We add Normal distributed noise to this relation with a coefficient of variation of 0.5. To relate the colour to the demand, we divide the demand into five equal segments based on the percentiles. Each colour relates for 80% of the products to a specific segment, and for 20% to other segments as noise. Table C.1 shows the probabilities of the colour of a product, given the demand percentiles of the Gamma(2, 150) distribution. Marked in bold are the relations which are the most likely.

For the category and brand of a product, we apply the same method as for the colour. The category and brand are for 80% related to a specific profile, and for 20% related to other profiles. The conditional probabilities are shown in Table C.2 and Table C.3 for respectively the category and brand.

Since there are five demand segments and three profiles, there exist 15 types of products. For each demand segment and profile, we have defined more likely and less likely product characteristics. To provide an overview, The most likely characteristics for these type of products are listed in Table C.4.

We also show three example products and their demand profile. These products different profiles and a different total demand range. The characteristics of these products belong to the most likely characteristics. All demand points are generated based on the demand and profile, with a CV of 0.25 for the profile. The demand points are visualised in Figure C.1 and the characteristics of the products are shown in Table C.5. In the figure, we observe the three different profiles with some randomness.

Demand percentiles	Black	Yellow	Green	White	Gray	Orange	Blue	Purple	Brown	Red	*Total*
$P_0 - P_{20}$	**0.40**	**0.40**	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	*1*
$P_{20} - P_{40}$	0.025	0.025	**0.40**	**0.40**	0.025	0.025	0.025	0.025	0.025	0.025	*1*
$P_{40} - P_{60}$	0.025	0.025	0.025	0.025	**0.40**	**0.00**	0.025	0.025	0.025	0.025	*1*
$P_{60} - P_{80}$	0.025	0.025	0.025	0.025	0.025	0.025	**0.40**	**0.40**	0.025	0.025	*1*
$P_{80} - P_{100}$	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	**0.40**	**0.40**	*1*

Table C.1: Probabilities of the colour of a product, given the demand percentiles of G(2,150)

Profile	Kitchen	Smart home	Sound	Television	Accessories	Photography	Tablets	Computers	Games	Telephone	*Total*
Increasing, 10% exponential	**0.211**	**0.211**	**0.211**	**0.211**	0.026	0.026	0.026	0.026	0.026	0.026	*1*
Decreasing, 10% exponential	0.032	0.032	0.032	0.032	**0.258**	**0.258**	**0.258**	0.032	0.032	0.032	*1*
Stable	0.032	0.032	0.032	0.032	0.032	0.032	0.032	**0.258**	**0.258**	**0.258**	*1*

Table C.2: Probabilities of the category of a product, given the profile

Profile	Animity	Mudeo	Octozzy	Outise	Supranu	Transible	Kayosis	Dynotri	Hyperive	Verer	*Total*
Increasing, 10% exponential	**0.211**	**0.211**	**0.211**	**0.211**	0.026	0.026	0.026	0.026	0.026	0.026	*1*
Decreasing, 10% exponential	0.032	0.032	0.032	0.032	**0.258**	**0.258**	**0.258**	0.032	0.032	0.032	*1*
Stable	0.032	0.032	0.032	0.032	0.032	0.032	0.032	**0.258**	**0.258**	**0.258**	*1*

Table C.3: Probabilities of the brand of a product, given the profile

Profile	Demand distribution and percentile range	Most common categories	Most common brands	Most common colors	Most likely price range
Increasing, 10% exponential	Gamma(2, 150), $P_0 - P_{20}$	Kitchen, smart home, sound, television	Animity, mudeo, octozzy, outise	Black, yellow	$[16.17, \infty)$
	Gamma(2, 150), $P_{20} - P_{40}$	Kitchen, smart home, sound, television	Animity, mudeo, octozzy, outise	Green, white	$[9.69, 16.17)$
	Gamma(2, 150), $P_{40} - P_{60}$	Kitchen, smart home, sound, television	Animity, mudeo, octozzy, outise	Gray, orange	$[6.59, 9.69)$
	Gamma(2, 150), $P_{60} - P_{80}$	Kitchen, smart home, sound, television	Animity, mudeo, octozzy, outise	Blue, purple	$[4.45, 6.59)$
	Gamma(2, 150), $P_{80} - P_{100}$	Kitchen, smart home, sound, television	Animity, mudeo, octozzy, outise	Brown, red	$[0, 4.45)$
Decreasing, 10% exponential	Gamma(2, 150), $P_0 - P_{20}$	Accessories, photography, tablets	Supranu, transible, kayosis	Black, yellow	$[16.17, \infty)$
	Gamma(2, 150), $P_{20} - P_{40}$	Accessories, photography, tablets	Supranu, transible, kayosis	Green, white	$[9.69, 16.17)$
	Gamma(2, 150), $P_{40} - P_{60}$	Accessories, photography, tablets	Supranu, transible, kayosis	Gray, orange	$[6.59, 9.69)$
	Gamma(2, 150), $P_{60} - P_{80}$	Accessories, photography, tablets	Supranu, transible, kayosis	Blue, purple	$[4.45, 6.59)$
	Gamma(2, 150), $P_{80} - P_{100}$	Accessories, photography, tablets	Supranu, transible, kayosis	Brown, red	$[0, 4.45)$
Stable	Gamma(2, 150), $P_0 - P_{20}$	Computers, games, telephone	Dynotri, hyperive, verer	Black, yellow	$[16.17, \infty)$
	Gamma(2, 150), $P_{20} - P_{40}$	Computers, games, telephone	Dynotri, hyperive, verer	Green, white	$[9.69, 16.17)$
	Gamma(2, 150), $P_{40} - P_{60}$	Computers, games, telephone	Dynotri, hyperive, verer	Gray, orange	$[6.59, 9.69)$
	Gamma(2, 150), $P_{60} - P_{80}$	Computers, games, telephone	Dynotri, hyperive, verer	Blue, purple	$[4.45, 6.59)$
	Gamma(2, 150), $P_{80} - P_{100}$	Computers, games, telephone	Dynotri, hyperive, verer	Brown, red	$[0, 4.45)$

Table C.4: Most likely characteristics for each profile & demand combination

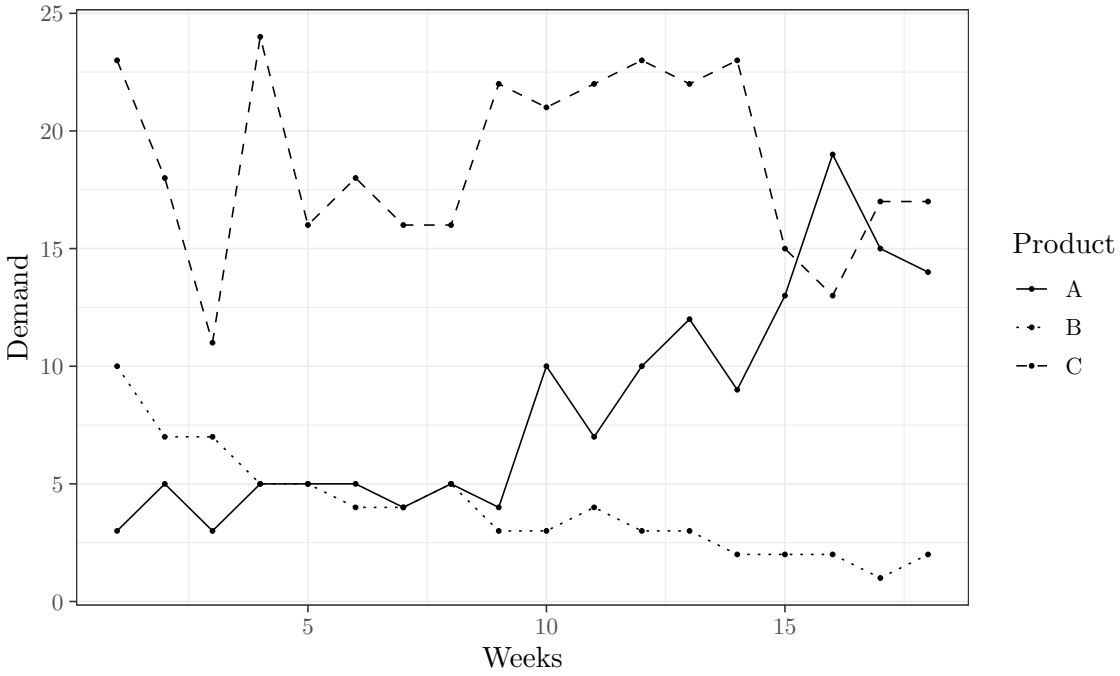


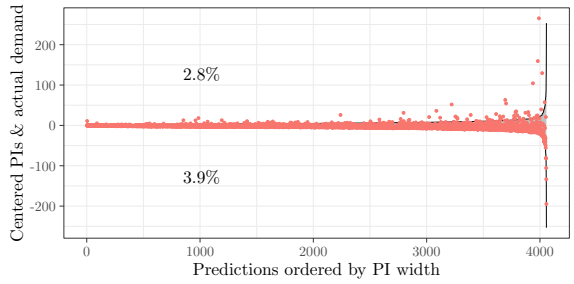
Figure C.1: Demand of three example products from the synthetic data set

Product	Profile	Demand segment	Colour	Category	Brand	Price	Demand
A	Gamma(2, 150), $P_{20} - P_{40}$	Increasing, 10% exponential	White	Sound	Outise	10.8	148
B	Gamma(2, 150), $P_0 - P_{20}$	Decreasing, 10% exponential	Yellow	Photography	Supranu	23.11	72
C	Gamma(2, 150), $P_{60} - P_{80}$	Stable	Blue	Games	Hyperive	5.93	335

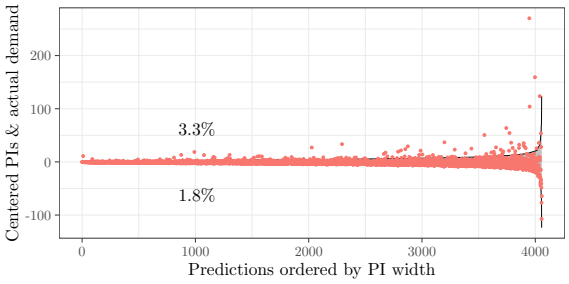
Table C.5: Characteristics of three example products from the synthetic data set

D | Appendix Prediction Intervals

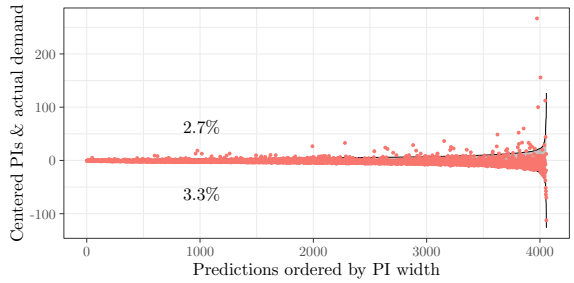
This appendix shows the 90% interval plots for the different methods at company A (see Figure D.1), company B (see Figure D.2), company D (see FigureD.3), and company E (see Figure D.4).



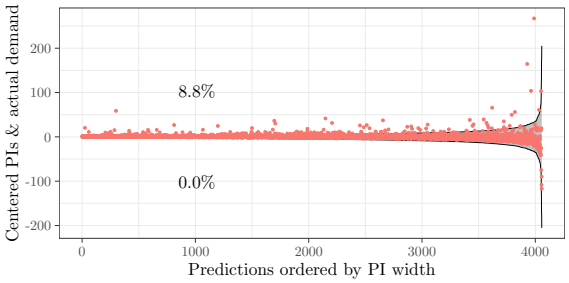
(a) demandForest



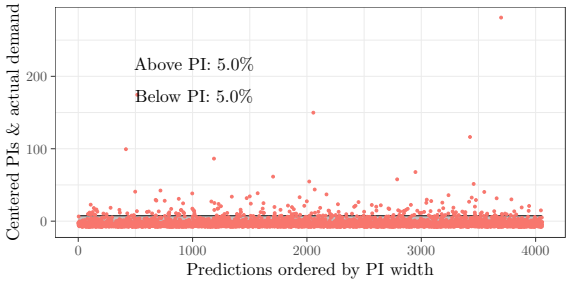
(b) dF + Gamma



(c) dF + Log-Normal

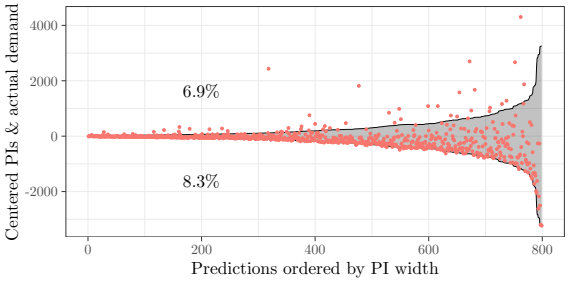


(d) Proximity

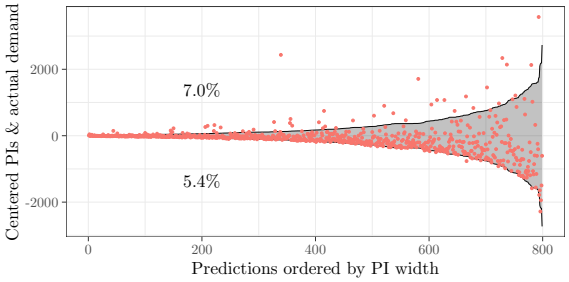


(e) ZeroR

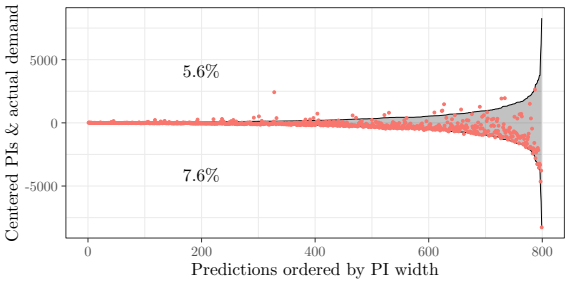
Figure D.1: Centered 90% prediction intervals for each method for company A



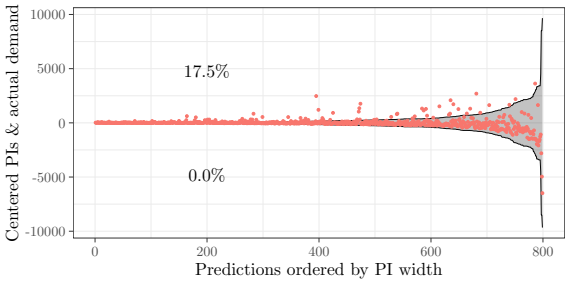
(a) demandForest



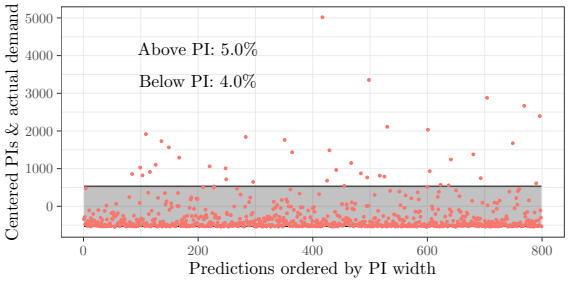
(b) dF + Gamma



(c) dF + Log-Normal

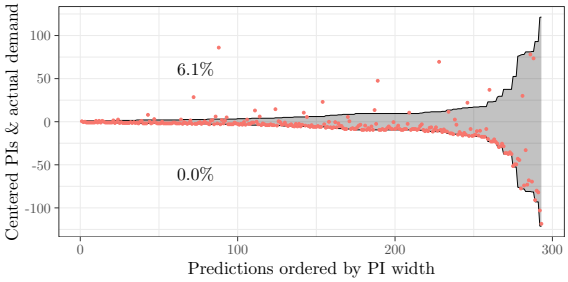


(d) Proximity

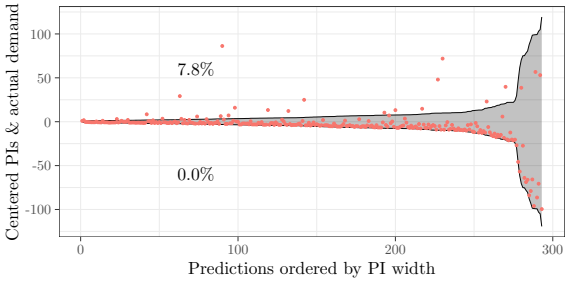


(e) ZeroR

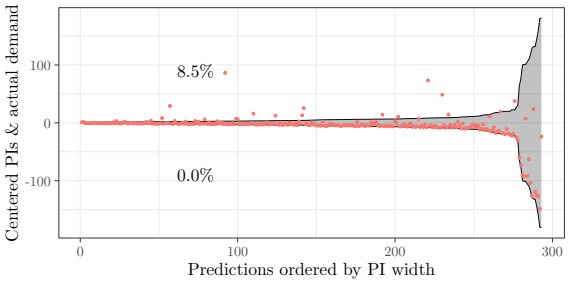
Figure D.2: Centered 90% prediction intervals for each method for company B



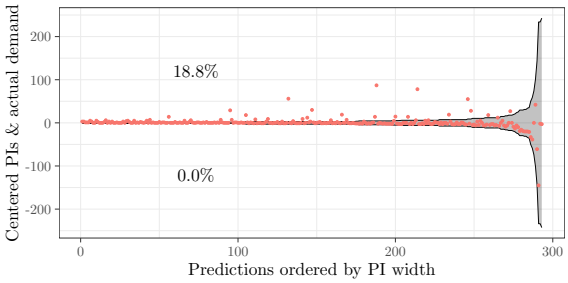
(a) demandForest



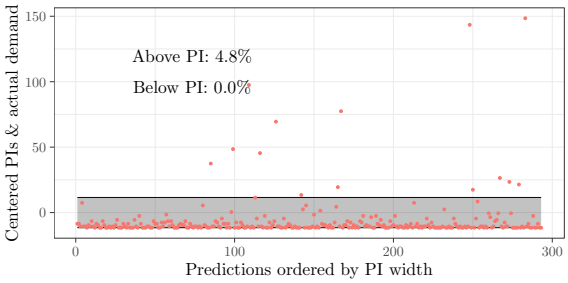
(b) dF + Gamma



(c) dF + Log-Normal

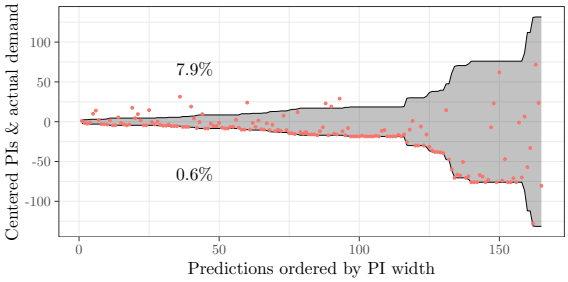


(d) Proximity

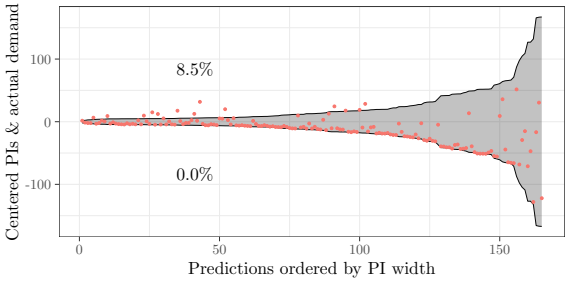


(e) ZeroR

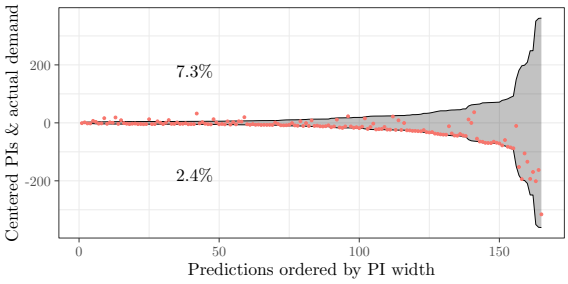
Figure D.3: Centered 90% prediction intervals for each method for company D



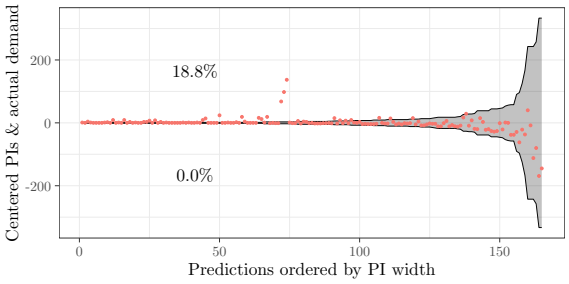
(a) demandForest



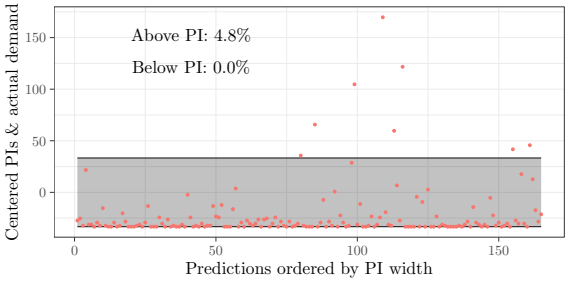
(b) dF + Gamma



(c) dF + Log-Normal



(d) Proximity



(e) ZeroR

Figure D.4: Centered 90% prediction intervals for each method for company E