WHU
Otto Beisheim School of Management

JENNIFER WEINGARTEN

THE VALUE OF CUSTOMER BEHAVIOR IN SUPPLY CHAIN MANAGEMENT: PREDICTIVE ANALYTICS APPLICATIONS IN DEMAND FORECASTING

Dissertation
for obtaining the degree of Doctor of Business and Economics
(Doctor rerum politicarum - Dr. rer. pol.)

at WHU - Otto Beisheim School of Management

October 2, 2021

First Advisor: Prof. Dr. Stefan Spinler

Second Advisor: Prof. Dr. Christian Schlereth

To my family and Philipp
for their love and support.

ABSTRACT

This dissertation investigates the value of customer behavior in supply chain management through the application of (big) data analytics in demand forecasting. The use of advanced analytics in supply chain management is not novel. However, the growing expansion of data volumes provides companies with new opportunities to optimize their supply chain. Despite the rising interest from both academia and practice and the recent increase in publications in this area, empirical insights are still limited. At the same time, changes in customer expectations towards instant product delivery require companies to rethink their supply chain, where accurate demand forecasts are often at the core of enabling efficient and flexible processes. This makes the development of demand prediction models that can be used in practice especially relevant.

We[1] analyze the use of customer behavior in demand forecasting in three separate research papers. Leveraging data from research partners in the online fashion and construction industry, we assess the potential of the developed prediction models in three areas of application in supply chain management, namely order fulfillment, order picking, and inventory planning.

In the first paper, we develop a prediction model for anticipatory shipping in the fashion industry, which predicts customers' online purchases with the aim of shipping products in advance, and subsequently minimizing delivery times. Using various forecasting methods and data on customers' behavior on the website, we test if, and how early, it is possible to predict online purchases. Results indicate that customer purchases are, to a certain extent, predictable, but anticipatory shipping comes at a high cost due to wrongly sent products.

The second paper assesses the extent to which clickstream data can improve forecast accuracy for fashion products. Specifically, we assess which clickstream variables are most suitable for predicting demand, and identify the products that benefit most from this. Results indicate that clickstream data is especially useful for forecasting medium- and certain intermittent-demand products. A simulation of order picking for these products shows that using clickstream data in the forecast substantially decreases picking times.

The third paper investigates how sequential pattern mining can be used to determine products with correlated demand, and how to leverage this as an input into forecasting for a supplier in the construction industry. We find that sequential pattern mining may be beneficial when used in combination with traditional forecasting methods, and that support vector regression models seem especially suited to forecast intermittent-demand products. An application to inventory planning shows that our developed forecasting model might reduce the company's costs of inventory holding and lost sales by up to 6.9%.

Overall, our research highlights the value of using customer behavior to enhance demand forecasting and the benefit of using improved forecasts in various applications in supply chain management.

[1] Referring to the authors of the respective chapters as noted at the beginning of each chapter.

ACKNOWLEDGMENTS

This quasi-cumulative dissertation was prepared at the Kühne Institute for Logistics Management at WHU – Otto Beisheim School of Management between 2018 and 2020.

First and foremost, I would like to thank Prof. Dr. Stefan Spinler—my first advisor at the Otto Beisheim School of Management—for his continuous support throughout my doctoral studies. He helped shape my research efforts, always had an open mind for my ideas, and gave invaluable advice and feedback whenever needed.

I would also like to thank Prof. Dr. Christian Schlereth for serving in the capacity of my second advisor and for providing helpful perspectives on the topic of this thesis.

Further, I would like to thank the entire team at the Kühne Institute for Logistics Management. Even as an external doctoral student, I always felt part of the team and could ask for guidance and support whenever needed. I am especially grateful to Andreas Faber, for always taking the time when I needed advice, to Sven Falkenberg, for helping me solve problems as if it was his own research, and to Adrian Viellechner, for sharing a similar research journey and making this a whole lot of fun through countless discussions on machine learning algorithms and programming code.

I would also like to thank the two case study companies that I had the honor to work with throughout my doctoral studies. Without their dedication to provide me with two very unique datasets and their endless support, this dissertation would not have been possible.

Lastly, I want to thank my family, especially my mother, Philipp, and my close friends for supporting me through all the 'ups' and also 'downs' in the last years and for making this an incredible journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACF	Autocorrelation function
ADI	Average demand interval
AIC	Akaike information criterion
AP	Across all products
ARIMA	Autoregressive integrated moving average
AS	Anticipatory shipping
AUPR	Area under the precision-recall curve
BDA	Big data analytics
B2B	Business-to-business
B2C	Business-to-consumer
Croston	Croston's method
CV2	Squarred coefficient of variation
DHR	Dynamic harmonic regression
DR	Dynamic regression
ERNN	Elman recurrent neural network
ETS	Exponential smoothing
FFNN	Feed-forward neural networks
GDP	Gross domestic product
GDPR	General Data Protection Regulation
h	Hours
ID	Identifier
IDC	International Data Corporation
IT	Information technology
JRNN	Jordan recurrent neural network
k	Thousand
LG	Logistic regression
LSTM	Long short-term memory networks
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MASE	Mean absolute scaled error
ML	Machine learning
NN	Neural networks
PCA	Principal component analysis

PP	Per product
PS	Pocket sorter
r	Pearson correlation
RF	Random forest
RFID	Radio-frequency identification
RMSE	Root-mean-square error
RNN	Recurrent neural networks
SCM	Supply chain management
SES	Simple exponential smoothing
SKU	Stock-keeping unit
SMA	Simple moving average
SPADE	Sequential pattern discovery using equivalence classes
SSE	Sum of squared errors
SVM	Support vector machine
SVR	Support vector regression

1 INTRODUCTION

1.1 BIG DATA ANALYTICS

In 2018, a study from the International Data Corporation (IDC) (Reinsel et al. 2018) predicted that the collective world's data would grow from 33 to 175 zettabytes[1] by 2025. By that time, 75% of the world's population is expected to interact with data on a daily basis, fueling the trend of massive data generation. The capability to draw insights from big data has the potential to be a competitive asset for many companies (McAfee et al. 2012, Waller and Fawcett 2013). Already today, companies have been leveraging large data volumes to create new and better customer services, improve their processes and operations, guide strategic and also day-to-day decision-making. This data-driven approach has enabled companies to increase both their productivity and profitability as research shows (McAfee et al. 2012). With large and complex data volumes being widely available today and becoming less costly to store and analyze (McAfee et al. 2012), in combination with a growing understanding of how to leverage this data in actual business applications (Brown et al. 2011), big data is, not surprisingly, receiving increasing attention from both academia and the industry (Akter and Wamba 2016).

While various definitions of big data exist, big data is typically characterized by large quantities of heterogeneous data from a variety of different sources (e.g., social media, sensors), often generated in (near) real-time. These data volumes are usually difficult to store, manage, and analyze with traditional software tools (Nguyen et al. 2018, Vassakis et al. 2018). To extract insights from big data, big data analytics (BDA) has emerged, which refers to the application of advanced analytics techniques, including artificial intelligence, to gain insights from big data, and ultimately leverage these insights to enhance decision-making (Wang et al. 2016). Nowadays, BDA can be found in every industry and economy, with many enterprises focusing on developing the skillset to gain knowledge from big data (Vassakis et al. 2018). BDA can generally be applied to provide insights into past events (descriptive analytics), enable the prediction of things before they happen (predictive analytics), and develop recommendations for future actions to support decision-making (prescriptive analytics) (Nguyen et al. 2018). Especially predictive analytics can provide companies with a competitive edge, as it helps them to define future opportunities and risks. A typical application of predictive analytics is the prediction of customer behavior, enabling companies to optimize marketing efforts, product development, and operations, amongst others (Vassakis et al. 2018).

[1] One zettabyte is approximately equal to a billion terabytes.

1.2 BIG DATA ANALYTICS IN SUPPLY CHAIN MANAGEMENT

In supply chain management (SCM), analytics and data-driven decision-making are not novel topics. Complex analytics techniques are frequently used to optimize the supply chain (Tiwari et al. 2018). However, the growing expansion of data volumes generated from end-to-end supply chain management creates new opportunities for companies (Kache and Seuring 2017). Given the advancements in BDA, scientific research of SCM might be at a tipping point (Sanders and Ganeshan 2018). In addition to traditional supply chain data points (e.g., sales, inventory), information from unstructured sources is collected (e.g., online reviews, tweets, or website clickstream data). At the same time, this data is becoming available on a more granular and real-time level (Sanders and Ganeshan 2018). This provides companies with valuable information on topics such as consumer sentiment, sales trends, and real-time inventory availability. As supply chain performance depends to a large degree on information, BDA could be especially beneficial for SCM (Tiwari et al. 2018). Considering that publications in this area have only recently increased, there has been no clear terminology on the use of big data in SCM yet. Scholars refer to it as SCM data science (Waller and Fawcett 2013), supply chain analytics (Wang et al. 2016), or predictive analytics (Schoenherr and Speier-Pero 2015). However, all these terminologies are similar in the sense that they refer to the use of advanced analytics to utilize big data in SCM (Brinch et al. 2018).

The benefits of BDA in SCM are vast and include improvements in supply chain efficiency, enhanced supply chain planning, and a reduction of overall supply chain cost (Schoenherr and Speier-Pero 2015). Potential applications of BDA in SCM have been investigated in various studies. Sanders (2016) outlines areas of application across the SCM categories source, make, move, and sell. Choi et al. (2018) review various BDA techniques and how they can be applied in different areas of operations management, such as forecasting, inventory management, and transportation. Wang et al. (2016) review literature on the application of BDA in logistics and supply chain management and distinguish areas of application by the nature of BDA (i.e., descriptive, predictive, or prescriptive) and whether the application focus is on a strategic or operational level. In a similar manner, Nguyen et al. (2018) also distinguish applications of BDA in SCM by the nature of BDA as well as the methods applied (e.g., optimization, simulation) and outline applications by supply chain function.

Despite the expected value of BDA in SCM, its application in practice can be challenging, especially with respect to data collection and cleaning (Wang et al. 2016). Hazen et al. (2014) point out that the extent to which BDA is useful in SCM hinges to a large degree on the quality of the data collected and processed. According to a study by Rozados and Tjahjono (2014), big data in SCM is often distributed in information silos across non-interconnected business functions and external sources. Without connected data sources creating end-to-end visibility of the supply chain, it can be difficult to generate valuable insights from big data, which could explain the limited use of BDA in practice. Also, as Brinch et al. (2018) point out, studies providing insights on where big data may be most beneficial within SCM are needed.

While the application of BDA in SCM is on the rise in both academia and practice, researchers and practitioners have yet to discover its true potential (Schoenherr and Speier-Pero 2015, Srinivasan and Swink 2018, Tan et al. 2015). Overall, big data in SCM is a relatively new area that lacks empirical insights (Matthias et al. 2017, Tan et al. 2015). A recent literature review by Choi et al. (2018) specifically calls for in-depth case studies applying BDA in SCM.

1.3 THE RELEVANCE OF DEMAND FORECASTING IN SUPPLY CHAIN MANAGEMENT

Through the overlap of our fast-paced, data-driven digital world with the physical reality, customer expectations are being reset, causing a shift towards the need for real-time services and instant product delivery (Reinsel et al. 2018). This poses a challenge for companies' supply chains, where accurate demand forecasts are often at the core of enabling supply chain efficiency and flexibility (Hofmann and Rutschmann 2018). Accurate demand forecasts can prevent costs from holding excess inventory as well as lost revenue due to stockouts. According to Kremer et al. (2016), every percentage improvement in forecast accuracy results in a similar percentage improvement in terms of reduced safety stock, without a negative impact on customer service. Despite the volume of research on demand forecasting, recent studies suggest that practical applications of forecasting techniques still lag behind academic developments (Fildes et al. 2019, Syntetos et al. 2016). Specifically, current scientific contributions in this area are often mathematically complex and require expert knowledge (Hofmann and Rutschmann 2018), potentially hindering the application of the developed models in practice.

Historically, the prediction of future demand relies heavily on past sales, market information, and input from experts. Big data provides companies with the opportunity to leverage even more data points from various sources in the forecasting process (Choi et al. 2018). Specifically, unstructured data, such as website clickstream data, enable companies to better understand their customers' behavior. While many studies have aimed at modeling and predicting customer behavior, their application is usually marketing-related. To improve targeted marketing, Kim et al. (2005), for instance, use artificial neural networks to understand which products and services households might be interested in. Moe and Fader (2004) use online customers' clickstream data to predict the purchase probability of a customer visiting a product site in order to move customers that are likely to make a purchase to a better performing server. While these are just a few examples in the marketing domain, applications that leverage data related to customer behavior to improve supply chain performance (e.g., through enhanced demand forecasts) remain scarce (Cirqueira et al. 2020).

1.4 CONTRIBUTION OF THIS DISSERTATION

We[2] contribute to the existing literature by applying predictive analytics using data on customer behavior to improve demand forecast accuracy. From the

[2]Referring to the authors of the respective chapters as noted at the beginning of each chapter.

existing literature on predictive analytics in SCM, a limited number of studies have been dedicated to demand management (Nguyen et al. 2018). Studies that do investigate predictive analytics to forecast future demand are typically focused on applications in revenue management and marketing, for instance, to provide personalized customer services (Choi et al. 2018). Moreover, most studies in predictive analytics, independent of their area of application, focus on the predictive performance of the developed models, with limited insights on both their ease of implementation and actual impact in a business setting. Also, as previously mentioned, the utilization of BDA in SCM is still limited (Nguyen et al. 2018, Waller and Fawcett 2013). To add to these research gaps, we use predictive analytics to forecast future demand with the aim of improving processes within the supply chain. Specifically, we evaluate the impact of our forecast models on warehouse operations.

This dissertation builds on three research papers, covering various aspects of predictive analytics in demand forecasting, partially with the specific challenge of forecasting intermittent demand. While the first two papers focus on customer order and demand forecasting at a large European business-to-consumer (B2C) e-commerce player, the third paper aims to improve demand forecasts at a leading business-to-business (B2B) supplier in the construction industry.

- Chapter 2 investigates how big data can be used to optimize delivery times for customers. We use the dataset from a European online fashion retailer to predict demand on an individual product-customer level to enable anticipatory shipping, which refers to the advanced shipment of products (i.e., before customers place their orders). Ideally, by the time an order is placed, the respective product is already stored at a location close to the customer to minimize delivery time. While Amazon already introduced the concept of anticipatory shipping years ago, limited research exists that investigates whether anticipatory shipping is actually possible and at what cost. Using a dataset containing both structured (e.g., customer gender and age) and unstructured data (e.g., clickstream data) and applying various machine learning forecasting methods, we first identify which method is best suited for our dataset. In a second step, we identify how early in the customer-product interaction it is possible to make accurate predictions (e.g., on the day customers viewed a product for the first time, or right after they added a product to their shopping cart). In the last step, we translate the result of our model into products (in-)correctly sent in advance and assess in a simulation of inventory planning and order fulfillment how anticipatory shipping would reduce delivery times and at which cost.

- Using the same dataset, Chapter 3 investigates to what extent clickstream data can improve product-level forecasts. While several studies have already assessed the value of clickstream data in predicting individual customer orders, typically for applications in marketing, this chapter focuses on the forecast of the aggregated product-level demand for a large product assortment. We add to the existing literature by specifically investigating the effect of single clickstream variables and use feature engineering

to define novel variables in this context. Using clustering techniques and further in-depth analysis, we also outline for which products the use of clickstream data in forecasting is especially beneficial. To assess the impact of our research in a supply chain context, we simulate warehouse order picking using the results of our forecast models with the aim of minimizing picking times across orders.

- To investigate the application of advanced analytics techniques in a B2B context, Chapter 4 uses the dataset from a leading supplier in the construction industry to assess how data mining and machine learning methods can be used to automatically identify relevant information for forecasting, which is typically provided by company experts. Limited research has investigated how 'soft data' typically identified by human judgment can be detected automatically and incorporated into forecasts. To add to this research gap, this chapter investigates how sequential pattern mining can be used to determine products with correlated demand, and how to leverage this information as an input into forecasting. The dataset used is characterized by a very heterogeneous product portfolio, with many products showing intermittent demand behavior, making forecasting especially challenging. Both traditional (i.e., time series methods) and machine learning forecasting methods combined with hierarchical clustering are applied to test the extent to which sequential pattern mining may help improve forecast accuracy. In an application to inventory planning, we assess how improvements in forecast accuracy impact the company's costs of inventory holding and lost sales resulting from stockouts.

To conclude this dissertation, Chapter 5 provides a condensed summary of the findings of Chapters 2-4. Further, we discuss potential avenues for relevant future research.

2

SHORTENING DELIVERY TIMES BY PREDICTING CUSTOMERS' ONLINE PURCHASES: A CASE STUDY IN THE FASHION INDUSTRY

The following chapter is based on Weingarten and Spinler (2021).[1]

2.1 INTRODUCTION

In recent years, the interest in big data has been growing from both academia and the industry (Akter and Wamba 2016). Big data is often defined in terms of 5 V's: volume, variety, velocity, veracity, and value (Wamba et al. 2015). Volume refers to the quantities of data, which require a massive amount of storage. Variety refers to the diverse types of data collected, which can be structured (e.g., customers' demographic data) and unstructured (e.g., likes, tweets) (Akter and Wamba 2016). Velocity stands for the speed of data generation and processing in (near) real-time. Veracity stresses the importance of data quality. Lastly, value relates to the process of extracting value from big data to aid decision-making (Akter and Wamba 2016, Nguyen et al. 2018). Big data provides tremendous opportunities as it is widely available and nowadays much less expensive to access and store (McAfee et al. 2012). Due to the large volume of data, the variety of data sources, and the speed at which data needs to be collected and analyzed, big data analytics (BDA) has emerged. BDA involves the application of advanced analytics techniques, such as statistics, simulation, or optimization, to gain insights from big data to enhance decision-making and increase business value and firm performance (Tiwari et al. 2018). Businesses that already use BDA report a 5% increase in productivity and a 6% increase in profitability, compared to those that do not (McAfee et al. 2012). In supply chain management (SCM), analytics and data-driven decision-making are not novel. Techniques such as statistics and simulation have frequently been used in the past to optimize the supply chain (Tiwari et al. 2018). However, the exponential increase in big data generated from end-to-end supply chain management creates new opportunities, as well as challenges, as companies are faced with the difficulty of mining large datasets (Tiwari et al. 2018). As supply chain performance depends to a large degree on information, BDA could be especially beneficial for SCM. Nevertheless, the research on the application of BDA in SCM is still in its infancy (Kache and Seuring 2017).

BDA has also been emphasized in the e-commerce context, where big data allows online sellers to track each customer's behavior, which provides companies with opportunities such as real-time customer service, dynamic pricing, or personalized promotion activities (Akter and Wamba 2016). While the time

[1] This manuscript has been published in the Information Systems Management journal in the special issue "Managing the Marketing-Operations Interface in Omnichannel Retail". A previous version of the manuscript has been published in the proceedings of the 53rd Hawaii International Conference on System Sciences (Weingarten and Spinler 2020a).

between purchase and product arrival used to be the main disadvantage of e-commerce players compared to brick and mortar stores, last-mile solutions, such as same-day or 2-hour delivery, enable almost instant gratification for consumers (Voccia et al. 2019). To enable nearly instant delivery services, products need to be stored close to the consumer (Hu and Monahan 2016). The large assortment of many e-commerce players, such as Amazon or Alibaba, makes this especially challenging. While many online retailers have been forward-deploying inventory to enable fast delivery (Hu and Monahan 2016), Amazon has been using BDA to predict customers' purchase behavior and as a result, ship products closer to the customers before they place their order online. Amazon has patented this approach as anticipatory shipping (AS) (Spiegel et al. 2013). Whether Amazon is successful with this method, and whether predicting costumers' purchase behavior is possible to the extent that it enables the successful shipping of products in advance, is, to the best of our knowledge, not known. To better understand the possibilities for AS, this paper investigates the predictability of customers' purchase behavior using BDA. Subsequently, we test how AS would impact delivery times using a simulation of inventory and order fulfillment at a large European online fashion retailer. Specifically, structured data (e.g., customer age and gender), as well as unstructured data (e.g., customers' online browsing behavior) are used to predict customers' purchases. An earlier version of this paper was published in the proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS).

The research questions that guide this study are:

1. To what extent can customer information and browsing behavior be used to anticipate consumer purchases to ship products in advance and subsequently decrease delivery time?
2. What is the optimal point in time to predict customer purchases?
3. What is the operational value of using predicted purchases for AS?

The structure of the paper is as follows: Section 2.2 reviews literature related to this study. Section 2.3 explains the applied research approach and methodology. Section 2.4 introduces the case study context and dataset. Section 2.5 presents the results and discusses managerial implications. Section 2.6 concludes the paper and gives an outlook on areas of future research.

2.2 LITERATURE REVIEW

Our research is related to four streams of the literature, namely, (i) the application of BDA in SCM, (ii) research aiming to understand and model customer behavior, with a specific focus on using BDA in e-commerce to predict customers' purchase behavior, (iii) literature assessing the application of various methods to predict customer purchases, and (iv) research regarding approaches for anticipatory shipping.

2.2.1 *Big data analytics and its application in supply chain management*

A widely adopted taxonomy of BDA classifies data analytics into descriptive, predictive, and prescriptive analytics (Nguyen et al. 2018). Descriptive analytics gives insights into past events, predictive analytics makes predictions about future events and prescriptive analytics gives recommendations for future actions to support decision-making (Nguyen et al. 2018). In the literature, BDA is currently a vividly discussed topic among scholars due to its wide area of application. Its usage in SCM still provides many areas for future research, although applications of all three types of BDA can be found across the entire spectrum of SCM. According to Nguyen et al. (2018), most studies regarding BDA in SCM are related to logistics or manufacturing. In logistics, big data collected from players in the distribution network, such as carriers and logistics service providers, can be leveraged to optimize transportation systems, for instance through the usage of radio-frequency identification (RFID) tags or mobile devices (Wang et al. 2016). In manufacturing, applications of BDA range from production to quality control, maintenance, and energy management, amongst others. Especially the presence of sensors in production facilities provides opportunities for BDA (O'Donovan et al. 2015). Fewer studies seem to have been dedicated to demand management, procurement, and inventory management (Nguyen et al. 2018). In demand management, one main application of BDA is demand forecasting. Cui et al. (2018), for instance, use social media data to improve demand forecast accuracy at an online retailer. A second, increasingly important topic in demand management that leverages BDA is demand shaping. As data on individual customers are becoming available, companies are able to make predictions on an individual level, allowing them to offer personalized promotions or prices and essentially influence customer demand (Feng and Shanthikumar 2018). In procurement, BDA can be used to manage sourcing risk as well as supplier performance and selection (Nguyen et al. 2018). An example of the latter can be found in Choi et al. (2018), who use BDA to prioritize information technology (IT) service procurement in the public sector. Lastly, in inventory management, BDA can help optimize stock levels to decrease inventory holding cost, backorders, and lost sales, for instance through improved demand forecasts or data sharing among players in the supply chain (Kache and Seuring 2017). For a more detailed list of BDA applications in SCM in both the scholarly and applied literature, see Nguyen et al. (2018). Our research adds to this stream of literature by investigating one particular application of BDA in SCM, namely anticipatory shipping, for which only limited research exists.

Despite the high expectations of BDA in SCM, its use in practice is still limited. According to a study by Rozados and Tjahjono (2014), big data in SCM is often distributed in information silos across non-interconnected business functions and external sources. Without connected data sources creating end-to-end visibility of the supply chain, it can be difficult to generate valuable insights from big data. This challenge is in line with our experience. Obtaining data from different sources to get visibility on the customers that were purchasing products, their browsing behavior on the website, and the subsequent delivery process of their orders, was one of the main difficulties of this study. We believe

that this reflects a common challenge of many SCM departments when using BDA, potentially being one reason for its limited use in practice.

2.2.2 *Understanding and modeling customer behavior*

Many studies have focused on understanding and also predicting customer behavior to improve customer-centric processes, but not necessarily with a focus on big data. The application of advanced analytics techniques, including machine learning, is frequently used to gain insights into customers' behavior, needs, and preferences (Lessmann and Voß 2009), especially in the field of marketing and customer relationship management. Typical applications are customer value optimization (e.g., Gessner and Volonino 2005), customer churn prediction (e.g., Chen et al. 2012, De Caigny et al. 2018), marketing response modeling (e.g., Kim et al. 2005), and customer segmentation (e.g., Mizuno et al. 2008), amongst others.

A specific stream of research focuses on customer behavior in e-commerce. E-commerce players typically deal with two types of data: structured (e.g., customer age, gender) and unstructured (e.g., clicks, likes, tweets), where the challenge of BDA lies in creating meaningful insights from the combination of the two (Akter and Wamba 2016). Typical applications of BDA in e-commerce are the identification of customer needs, market segmentation, or making relevant information available at the right time (Akter and Wamba 2016). An example of the latter is Amazon's recommendation system, which recommends products to customers based on an understanding of their preferences (Zhao et al. 2015). Unstructured data, such as clickstream data, typically find applications in demand forecasting (e.g., Cui et al. 2018, Yang et al. 2014) and marketing, for instance, to offer personalized services (e.g., Huang and Van Mieghem 2014). Using BDA to understand and predict the purchase behavior of online customers is particularly relevant for many e-commerce players as it can help to improve conversion rates, which refers to the share of website visits resulting in a purchase (Van den Poel and Buckinx 2005). Several studies have attempted to predict online purchase behavior, with many specifically using clickstream data (e.g., Huang and Van Mieghem 2014, Kim et al. 2005, Lo et al. 2016, Moe and Fader 2004, Montgomery et al. 2004, Nishimura et al. 2018, Sismeiro and Bucklin 2004, Van den Poel and Buckinx 2005, Xu et al. 2014), but usually not with the aim of improving supply chain performance. Moe and Fader (2004) present a model to predict purchase probabilities for a given site visit and re-direct visits with a high purchase probability to a better performing server. Sismeiro and Bucklin (2004) use Bayesian methods to predict the completion of tasks in the online purchase process, indicating that customers' browsing behavior is a relevant predictor for online purchases. Lo et al. (2016) predict whether a user is a purchaser or a non-purchaser as their day of purchase approaches. Van den Poel and Buckinx (2005) are the first to use data from various sources in their prediction of customer purchase probabilities, namely clickstream data, customer demographics, and historical purchase behavior. Specifically, they predict if a purchase is made during a customer's next website visit. They provide a detailed list of variables used in their research, as well as other papers,

and whether they were found to have a statistically significant effect. Building on this research, our study includes a similarly high variety of data in the prediction. However, we focus on events taking place during a product site visit, such as clicks on images, instead of clickstream data regarding detailed historical browsing behavior, such as the type of previous pages visited.

A detailed literature overview regarding the prediction of customer purchases in e-commerce can be found in Cirqueira et al. (2020). As their literature review outlines, applications, in which a combination of structured and unstructured data has been used to predict customers' online behavior and subsequently improve supply chain performance, are scarce. One example can be found in Huang and Van Mieghem (2014), who use clickstream and historical purchase data to predict the quantity and timing of offline orders to improve inventory management.

2.2.3 *Methods for predicting customers' online purchases*

Various forecasting methods have been applied to predict customers' online purchasing behavior as outlined in Cirqueira et al. (2020). One process that can be found in many studies includes the application of classification methods, where a qualitative output variable takes on values in one of N different classes (Hastie et al. 2009). Classification models aim to determine to which class a new observation belongs, for instance, whether a customer will purchase a product (yes or no), based on a set of training data for which the class is known. While many models for classification are well-known in the literature and easy to implement, some studies develop rather complex models to optimally predict purchase behavior. Those models could be complicated to replicate in a business context. Moe and Fader (2004), for instance, develop a comprehensive conversion model that decomposes a customer's conversion behavior to predict purchases, while Montgomery et al. (2004) develop a dynamic multinomial probit model to predict purchase conversion.

In the context of classification, several types of methods exist. Verbeke et al. (2012) summarize them into seven categories, namely decision trees, ensemble methods, neural networks, statistical classifiers, nearest neighbor methods, support vector machine (SVM) based methods, and rule induction methods. Decision trees recursively partition training observations into subsets according to a certain function of the input variable values. Essentially, decision trees consist of three main building blocks, namely nodes, which test the value of input variables, branches, representing the outcome of this test and connecting the tree to the next node, and lastly, terminal nodes, which predict the outcome (i.e., class) of an observation. Decision trees have relatively fast training times and are usually easier to understand than black-box models, such as neural networks (Maimon and Rokach 2010). Ensemble methods (e.g., random forests, bagging, or boosting) leverage the power of multiple decision trees to improve prediction performance (James et al. 2013). Neural networks are inspired by biological neural networks and consist of a network of neurons connected by functions and weights, which are estimated to fit the network to the training data (Maimon and Rokach 2010). While neural networks have been shown to

perform well throughout literature, they require long training times and are difficult to interpret (Maimon and Rokach 2010). Statistical classifiers (e.g., logistic regression) use an underlying probability model that assesses the relationship between the feature set and the output variable (Verbeke et al. 2012). Statistical classifiers are widely used in the literature and score high on comprehensibility and ease of implementation. Nearest neighbor methods measure the similarity of observations using various distance measures to classify observations. The disadvantage of these methods is that they do not work well with large datasets as they calculate the distance to all observations for each new observation to be classified. Moreover, nearest neighbor methods do not build a final model from the training data that can be used for prediction (James et al. 2013). SVM-based methods are often used for binary classification. SVMs plot each observation as a point in an n-dimensional space, where 'n' is the number of variables. They create an optimal hyperplane that separates data points into two classes. If the data points are not linearly separable, SVMs map the data points to a higher dimensional space to enable separation (James et al. 2013). Similar to neural networks, SVMs are difficult to comprehend (James et al. 2013). Lastly, rule induction techniques generate if-then rules to make predictions for the minority class, while observations are assigned to the majority class per default (Verbeke et al. 2012).

Verbeke et al. (2012) compare the performance of various classification methods from these seven categories and discover that most methods do not perform significantly different. This is comparable to the findings from other studies (e.g., Baesens et al. 2003, Lessmann et al. 2008).

2.2.4 *Anticipatory shipping*

Amazon has patented an approach for AS, in which the company uses big data, including order history and data from its e-commerce portal, to predict a customer's online purchases and ship products to a geographical area close to the customer. The final delivery address is not completely specified until the customer places the order online (Spiegel et al. 2013). Not much research regarding AS can be found in the literature. Lee (2017) presents a model for AS in an omnichannel context. The study uses associate rule mining based on the Apriori algorithm to predict orders within pre-defined clusters of demand points to ship products to the nearest distribution center in advance. A genetic algorithm is then applied to optimize AS in the distribution network. Viet et al. (2020) present a model for AS in the agro-food industry. They also apply associate rule mining but add a time threshold to take product perishability into account. Both papers use historical orders as input to associate rule mining to identify potential products and volumes for AS, assuming that association rules (e.g., 'if product A is purchased, product B is likely to be purchased later as well') found in the historical data are applicable to future orders. We believe that this approach is not suitable for the fashion industry where retailers have enormous, frequently changing assortments with few data points (e.g., past orders) available for each product, limiting possibilities to find association rules.

2.3 METHODOLOGY

2.3.1 *Research approach*

This paper follows a three-step approach (Figure 2.1) to predict online purchases as early and accurately as possible to enable advanced shipment of products while minimizing the number of products that are erroneously sent in advance with no subsequent purchase. For this, we use a dataset provided by a European online fashion retailer containing information on customers, their browsing behavior, as well as purchase history, spanning over a period of one year. As we want to predict whether an observation, referring to the interaction between a customer and a product page, will result in a purchase at a certain point in time, the response variable 'purchase decision' is defined as a binary variable that falls into one of two categories, yes (1) or no (0). We are thus faced with a binary classification problem. In the first step of our research, the data is split into training, validation, and test data, and several forecasting methods are applied to different datasets to evaluate which forecasting method and dataset yield the best results in terms of prediction accuracy. The first dataset consists of all observations, the second one contains only observations from customers with frequent purchases (at least 12 per year), and lastly, customers are split into clusters and each cluster is predicted separately. In this first step, the whole time period of the dataset is used, which essentially means that the prediction for purchases is made at the end of the one year time period. To actually achieve delivery time savings, the best performing forecasting method and dataset from step one are used to predict purchases at an earlier point in time, namely (i) at the end of the first day a customer viewed a product and (ii) right after a first 'add to cart' click occurred. Also, we investigate differences in predicting purchases in various product categories separately, to identify product categories that are easier to predict. Lastly, to estimate the impact of the forecasting methods, we first translate predictions into packages sent (in-) correctly. Afterward, we apply anticipatory shipping predictions to inventory planning and order fulfillment to simulate how much delivery time savings would be achieved and at which cost, measured in terms of products wrongly sent in advance without any subsequent purchases from customers in the same geographical area.

This paper assumes that the purchase behavior of a customer does not substantially change over time, hence one observation could be assumed to be from a time period outside that of the dataset. That is why the split into training, validation, and test data in this paper does not take into account any temporal order of observations, as would be done for time series forecasting.

2.3.2 *Forecasting methods and accuracy measure*

In the following, the applied forecasting methods are described. This is followed by an explanation of how variables are selected (feature selection) and lastly, how the accuracy of the methods is assessed and which assumptions are made.

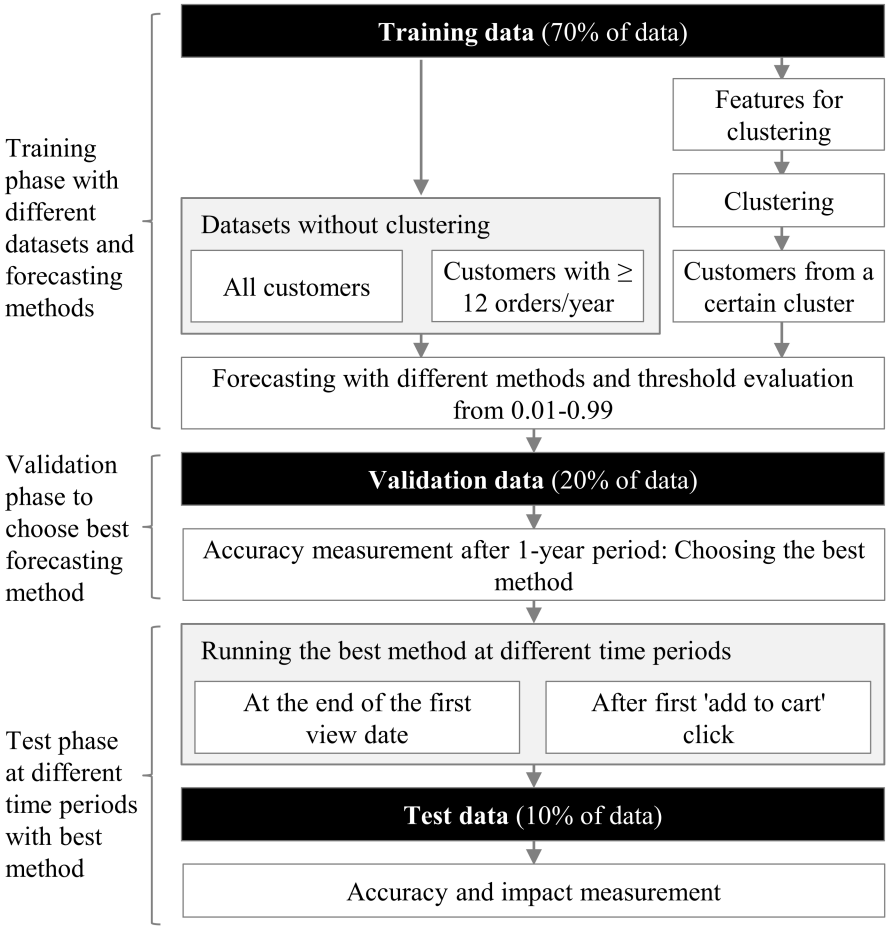

Figure 2.1: Research approach

2.3.2.1 *Forecasting methods and parameter selection*

As our research is largely motivated by practice, we focus on forecasting methods that score high on ease of implementation and comprehensibility. Therefore, we choose to apply already available and well-known classification methods for online purchase prediction and refrain from the development of a more complex prediction model.

As many classification methods perform similar in terms of forecast accuracy (Verbeke et al. 2012), we choose to apply only one method from each of the different types of classification methods outlined by Verbeke et al. (2012), except for rule induction and nearest neighbor. As mentioned in Section 2.2.4, we do not believe that meaningful rules can be derived for specific products due to the frequently changing assortment of online fashion retailers and the resulting lack of historical data per product. Nearest neighbor methods are excluded from this research as they have difficulty handling large datasets and do not result in a final model for prediction. From the decision tree and ensemble methods, random forest (RF) is applied, which is a popular learning method as it is simple to train while yielding high accuracy (Hastie et al. 2009). RF constructs an ensemble of decision trees, where each tree casts a vote for the predicted class and a majority vote is taken (Breiman 2001). In the class of

neural networks (NN), the multilayer perceptron is applied, which is the most commonly used form of neural networks. A multilayer perceptron consists of multiple neurons (nodes) arranged in several layers. It learns the relationship between the variables and the output variable through backpropagation (Hastie et al. 2009). In the field of statistical classifiers, logistic regression (LG) is selected, which is one of the most well-known methods in classification as it is much easier to understand and implement compared to many other methods. LG uses the logistic function to give outputs between 0 and 1, representing the probability that an observation belongs to class 0 or 1 (James et al. 2013). From the SVM-based methods, SVM with a linear and radial kernel is applied, where the kernel governs how the training data is mapped to a higher-dimensional space.

Most methods for classification do not work well if any class is heavily undersampled. This is often the case for data regarding purchase behavior as most website visits do not result in purchases (Moe and Fader 2004). For this reason, one-class SVM is tested as an additional method. One-class SVMs construct a decision boundary around the majority class to differentiate it from observations in the minority class, which are considered outliers or anomalies (Khan and Madden 2014).

All methods are implemented in R-3.5.1 and constructed to predict the probability that an observation belongs to one of the two classes. The threshold, which determines at which probability an observation is considered to lead to a purchase, is evaluated between 1-99% to determine which value leads to the highest overall accuracy. All predictor variables are standardized, except for RF. To ensure reproducibility, a summary of all hyperparameters tested for each forecasting method, as well as the functions and packages used in R, can be found in Appendix A in Table A.1. As all applied methods are well-known, we refrain from providing more detailed notations and refer to the relevant literature (Hastie et al. 2009, James et al. 2013, Maimon and Rokach 2010).

2.3.2.2 *Feature selection*

Correlation analysis is performed to remove highly correlated variables (pearson correlation (r) < 0.7). Additionally, lasso linear regression and RF are performed. Lasso uses a penalty term (lambda) for regression coefficients and can drive coefficients of non-relevant variables to zero, hence essentially excluding variables. The importance measure of RF assesses the mean decrease in accuracy if a variable is excluded. Additionally, a set of variables is selected that excludes categorical variables with many categories as they substantially increase training times. This results in five different sets of variables that are tested with each forecasting method:

- Set 1: All variables
- Set 2: All variables except categorical variables with more than 53 categories
- Set 3: All variables in the lasso output using lambda.min (minimum error observed)

- Set 4: All variables in the lasso output using lambda.1se (error is within 1 standard error of minimum error)
- Set 5: All variables in the RF output with a mean decrease in accuracy > 0.01% (as this already leads to a reduction of 50% of variables)

2.3.2.3 *Accuracy measure*

A confusion matrix is often used to evaluate the performance of classification models. In this study, five measures from the confusion matrix are used to assess model performance: accuracy, sensitivity, specificity, precision, and prevalence. Accuracy measures the overall proportion of correct classifications. Sensitivity assesses the proportion of observations resulting in a purchase that the classifier correctly predicted as such, while specificity measures the proportion of observations not resulting in purchases that the classifier correctly predicted as such. Precision measures from all observations the classifier predicted as purchase, the proportion that resulted in a purchase. Lastly, prevalence assesses the proportion of observations that resulted in a purchase. As the accuracy measured with a confusion matrix is often not appropriate for imbalanced datasets, the area under the precision-recall curve (AUPR) is calculated as an additional performance measure, which assesses the trade-off between precision and sensitivity (also called recall) (Davis and Goadrich 2006).

2.3.3 *Clustering*

According to Chen and Lu (2017), clustering, which is a method to partition datasets into homogenous subsets, can improve the performance of classification models. In this research, we aim to group customers with similar purchasing behavior to improve forecast accuracy. Various clustering methods exist, of which K-means and hierarchical clustering are two of the most well-known approaches. K-means is known for being a very efficient algorithm for clustering. It estimates the best way to divide the dataset into a pre-specified number of clusters by minimizing the within-cluster variation. To do this, K-means initially assigns each observation to a cluster in a random manner. After this initial assignment, the algorithm calculates the mean, also called centroid, of each cluster and reassigns each observation to the closest cluster based on the Euclidean distance to the cluster mean. This process is repeated until the cluster assignment stops changing. Hierarchical clustering, on the other hand, develops a hierarchical decomposition of the data. This enables the assessment of the obtained clustering for different numbers of clusters (James et al. 2013). Both methods have their advantages and disadvantages. For K-means clustering, we need to know in advance how many clusters to construct. Additionally, the algorithm has difficulty reaching a global optimum, caused by the random initial assignment of observations. Although hierarchical clustering is often able to provide better results, it is computationally expensive and therefore known to not work well with big data in comparison to K-means clustering (Maimon and Rokach 2010). Despite its shortcomings, several studies have successfully applied K-means clustering to improve forecast accuracy (e.g., Chang et al. 2009,

Chen and Lu 2017, Thomassey and Fiordaliso 2006). As we aim to develop a method for AS that can be implemented at a company that deals with a tremendous amount of data, we choose to apply K-means clustering.

To develop clusters of customers that differ in their purchasing behavior, we only use those numerical predictor variables for clustering that describe the customer as such. To apply K-means clustering, the data is standardized and the variables assessed for correlation. To determine the optimal number of clusters, we compute the within-cluster variation for 1-10 clusters. Moreover, to overcome the issue of local optima, we use 10 different initial cluster assignments that are randomly chosen. Subsequently, forecasting methods are used to train and predict each cluster separately.

2.3.4 *Impact estimation: application to inventory planning and order fulfillment*

To understand the impact of anticipatory shipping, the prediction results are translated into the number of products sent (in-)correctly. To further investigate the effect on delivery times, we simulate the application of anticipatory shipping for a subset of premium customers from the online fashion retailer. For this, we use the prediction results for the test dataset to simulate inventory and order fulfillment for two consecutive weeks. For the premium customers, we use the predictions to send products in advance to achieve delivery time savings. In the simulation, if a product purchase is predicted, the product is sent to the warehouse closest to the customer for whom the prediction was made, unless the warehouse already stocks the product. In order to assess this, the closest warehouse is determined for each zip code. Moreover, the product is reserved for 48 hours (h) for the respective customer.

As a lot of information regarding inventory and returns at the company is not known to us, five assumptions are made for the simulation:

1. Inventory can be picked, packaged, distributed to warehouses, and delivered to customers 24 h per day (including weekends).
2. We choose a supply chain network with specialized warehouses, meaning that each product is stored in only one warehouse at the beginning of the planning horizon.
3. Returns are not taken into account in the inventory development during the two weeks. However, to incorporate inventory from potential returns, the inventory at the start of the planning horizon is assumed to be equal to the fulfilled demand (i.e., sales) in the two weeks, plus the inventory needed for anticipatory shipping.
4. Shipping a product to another warehouse is assumed to take 8 h on average.
5. If an order can be fulfilled from the warehouse closest to the customer, delivery time is estimated to be 8 h on average. For all other warehouses, it is approximated at 16 h on average.

To evaluate the results from anticipatory shipping, we first establish a baseline by estimating how inventory and order fulfillment develop over the two weeks

without any predictions. The simulation algorithm for the baseline as well as the application of anticipatory shipping can be found in Appendix A.2.

2.4 CASE STUDY CONTEXT AND DATA

The data for analysis is provided by an online retailer in Europe that mainly sells fashion items. Like most online retailers, the case company tries to minimize delivery time. Consequently, they are interested in using predictive analytics to explore opportunities to decrease delivery times. For confidentiality reasons, no further information regarding customer and warehouse locations can be given. The data received[2] includes five types of datasets, which can all be linked via pseudonymized customer identification numbers:

- Customer information: gender, sign-up year, segment (mainly dependent on profitability)
- Order information: order date, products ordered, total number of orders per customer
- View information: number of product page visits of a customer, date, and length of visit
- Event information (information on where a customer clicked on a product page): event type (e.g., 'click on image', 'add to cart'), event date, total number of clicks per customer
- Product information: product category

A few additional variables are calculated based on the dataset. Comparing customer orders and product page views, different patterns are noticeable on a seasonal, monthly and weekly level (Figure 2.2). In terms of seasons, fall is the time with the lowest amount of views, but with the second-highest number of orders. This could potentially indicate that customers' purchase behavior differs across seasons, for instance taking faster decisions, in terms of product page views, to order products in the fall. On a monthly level, orders and views show slightly different patterns, especially between November and January, potentially driven by different customer behavior in the weeks leading up to Christmas. On a weekly level, most product page views occur on Sundays, while it is not the weekday with the most orders. Perhaps customers preferably browse through product sites on Sundays with no subsequent purchase, or subsequent purchase in the following days. Due to this, we add the season, month, and weekday on which a customer viewed a product for the first time as additional variables to the dataset. Moreover, for each customer, the number of times a product page was opened, the total number of events that occurred on a product page, the order frequency (number of orders per month) and lastly, the average decision time, which is the average time between the first date a

[2]The data protection principles of the General Data Protection Regulation (GDPR) are strictly followed so that any personal data received is in a form which does not permit identification of data subjects. Data is maintained and encrypted using Advanced Encryption Standard 256-bit encryption.

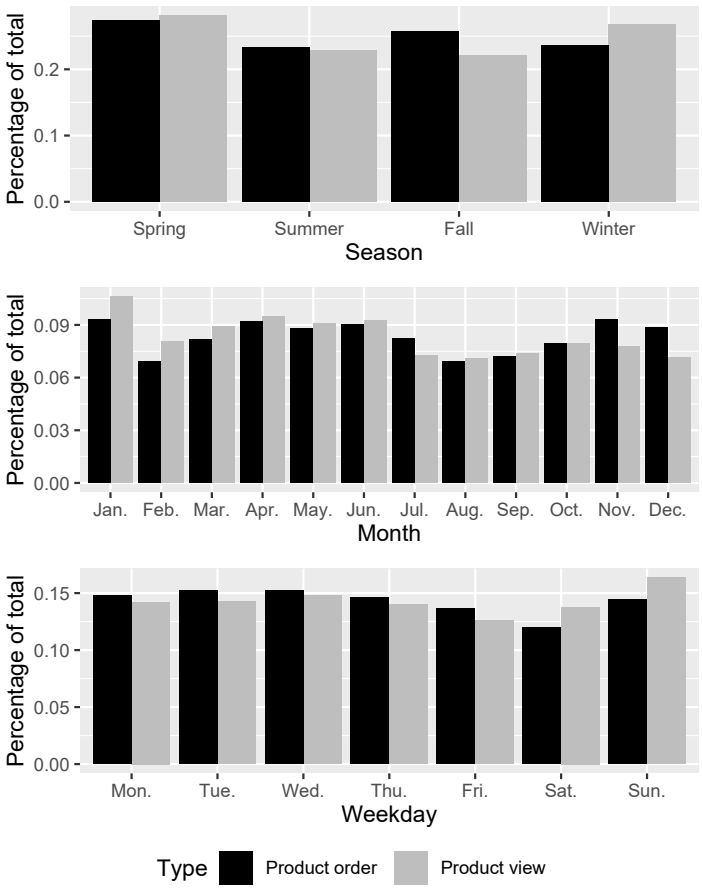

Figure 2.2: Difference between product order volume and product view volume per season, month, and weekday

product was viewed and the order date, is calculated. Interestingly, there is a large difference in the average decision time between women and men. Women take on average 1.74 days between the first time they view a product and subsequent product order, while men take 1.37 days. To test whether the difference is statistically significant, we use an unpaired two-sample Wilcoxon rank test, also referred to as Mann Whitney test, as the data is not normally distributed. The p-value of the test is below the significance level alpha = 0.05, indicating that men's average decision time is significantly different from women's average decision time. This could have substantial implications for any type of anticipatory action in the supply chain.

Overall, most of the variables used, especially gender, order frequency, number of product page visits of a customer, length of a visit and the total number of clicks per customer, are all found to have a statistically significant effect in the study of Van den Poel and Buckinx (2005). A full list of all variables used can be found in Table 2.1. Five variables are not included in this list as they resemble click types unique to the case study partner's website and are therefore confidential.

Also for confidentiality reasons, information regarding product price was not provided by the case study company. A lower product price could potentially result in higher purchase probability or a faster average decision time, which

both would have important implications for anticipatory shipping. However, due to the lack of data, this hypothesis cannot be investigated but should be subject to future research.

As the decision for repeat purchases (e.g., due to the wrong size) is assumed to be different from the decision to order a product for the first time, all data that occurred after a customer purchased a product for the first time is excluded from the dataset.

To predict whether a customer will buy a product, a prediction on a customer-product level is made. One observation in the constructed dataset thus contains the views and events that happened between a customer and a product page, combined with the general information of that specific customer and product. Due to the size of the dataset, a random sample of 100 thousand (k) customers is selected, resulting in a total of 8.3 million observations. From those, only 3.8% resulted in a purchase, indicating that the dataset is imbalanced as the classification categories are not equally represented. Due to this imbalance, techniques for dataset balancing, such as over- and undersampling, could be investigated. Dataset balancing would cause an increased bias towards the minority class. However, the aim is to predict as many purchases as possible while trying to predict observations that do not result in a purchase as accurately as possible. The latter is necessary to avoid erroneously sending a large number of products in advance. In this case, a bias towards the majority class is beneficial, which is already achieved by the currently imbalanced dataset (Maimon and Rokach 2010). Balancing techniques are hence not applied.

2.4.1 *Feature selection*

The correlation matrix can be found in Table 2.2. Six variables show high correlation: the number of times a customer visited a certain product page and the total time a customer viewed the product page (r = 0.78), the total number of events and the total number of product page visits of a customer during the one-year time period (r = 0.80), and the number of times a customer opened and closed the image gallery on a product page (r = 0.84). Lasso using lambda.1se is most aggressive in terms of feature selection and results in a set of 23 variables, while lasso using lambda.min results in 34 variables. To perform RF, the variables relating to product category have to be reduced to 53 categories for implementation in R. Those product categories with a small number of observations are hence removed until the maximum number of 53 categories is reached, resulting in a 25% decrease in dataset size. Applying RF results in 24 variables with a mean decrease in accuracy above 0.01%. The variable with the largest decrease in accuracy is, as can be expected, the 'add to cart' click (2.23%). As incorporating the product category variables with more than 53 categories leads to such a substantial decrease in dataset size when limiting the number of categories to 53, those variables are excluded from analysis using RF, resulting in variable set 5b.

Table 2.1: List of variables used

Number	Variable	Variable type
1	Total time a customer viewed a product page[1,2,3,4,5]	Numerical
2	Total number of product page visits of a customer[1,2,3,4,5,6]	
3	Number of times a customer opened a certain product page*	Numerical
4	Average decision time (average time between the first date a product was viewed and order date)[1,2,3,4,5,6]	Numerical
5	Sign-up year of the customer (ranging from 1-10)[1,2,3,4,5,6]	Numerical
6	Order frequency (average number of orders/month)[1,2,3,4,5,6]	Numerical
7	Total number of events (product page clicks) of a customer*	Numerical
8	Number of events (product page clicks) of a customer on a specific product page[1,4,5]	Numerical
9	'Add to cart' click[1,2,3,4,5]	Numerical
10	'Add to wishlist' click[1,2,3,4,5]	Numerical
11	Thumbnail click[1,4,5]	Numerical
12	Description box click[1,2]	Numerical
13	Main image click[1,4,5]	Numerical
14	'Choose size' click[1,2,3,4,5]	Numerical
15	Matching products click[1]	Numerical
16	'Change colour' click[1,2,3,4,5]	Numerical
17	Thumbnail arrow click[1,2]	Numerical
18	Measures box click[1,2,3,4,5]	Numerical
19	'Open image gallery' click[1,2,3]	Numerical
20	Thumbnail gallery click[1,2]	Numerical
21	Delivery details click[1,2,3]	Numerical
22	'Send size request' click[1,2]	Numerical
23	Size dropdown menu click[1,2,3,4,5]	Numerical
24	Size dropdown menu click (only in app)[1,2,3,4,5]	Numerical
25	Materials box click[1,2,3]	Numerical
26	'Close image gallery' click*	Numerical
27	'Open size request' box click[1,2,3]	Numerical
28	Size overview click[1,2]	Numerical
29	Similar product click[1,2]	Numerical
30	Brand click[1,2]	Numerical
31	'Show all similar products' click[1]	Numerical
32	Materials box click (only in app)[1]	Numerical
33	Measures box click (only in app)[1]	Numerical
34	'Close box' click (only in app)[1]	Numerical
35	Purchase decision average (share of a customer's observations resulting in a purchase)[6]	Numerical
36	Customer gender[1,4,5]	Categorical
37	Customer segment (based on customer profitability)[1,2,3,4,5]	Categorical
38	Product category 1 (e.g., dress, t-shirt, etc.)[2,3,4]	Categorical
39	Product category 2 (e.g., 'women denim')[2,3,4]	Categorical
40	Weekday a customer viewed a product for the first time[1,2,4,5]	Categorical
41	Month a customer viewed a product for the first time[1,2,3,4,5]	Categorical
42	Season a customer viewed a product for the first time[1,2,4,5]	Categorical

*Excluded from analysis due to high correlation.
1 Variable set 2: All variables except categorical variables with more than 53 categories.
2 Variable set 3: All variables in lasso output using lambda.min.
3 Variable set 4: All variables in lasso output using lambda.1se.
4 Variable set 5: All variables in random forest output with mean decrease in accuracy $> 0.01\%$.
5 Variable set 5b: Variable set 5, excluding product category variables.
6 Variable used for clustering.

For clustering, the choice of using numerical variables that describe the customer as such results in a set of four variables as input to clustering (Table 2.1), after previously removing highly correlated variables. To represent the relation between the set of 100k customers and 8.3 million observations, a variable determining each customer's purchase decision average is used as additional input. It assesses the share of a customer's observations that led to a purchase. All five variables show a low correlation ($r < 0.50$), which is why no further steps for decorrelation are performed.

2.5 RESULTS

2.5.1 *Results from different forecasting methods and datasets*

In the following, we will outline the results of using various forecasting methods to predict purchases using different datasets.

2.5.1.1 *Dataset 1: all customers*

The size of the training dataset is too large for most of the forecasting methods. Therefore, we estimate the required training size to obtain meaningful results using logistic regression, resulting in ~115k observations. Using fewer observations leads to an outcome where each observation is predicted to not result in a purchase. To be more conservative, a training size of 175k observations is used across all methods. Afterward, training size is increased for each method, using the optimal choice of parameters (Table A.1) and variable set, until no more significant improvements in accuracy are achieved or model training results in an error, for example, due to non-convergence of algorithms.

The first results from one-class SVM indicate that the model is not appropriate for this particular binary classification problem, as prediction accuracy is exceptionally low. A large fraction of non-purchases is not identified, leading to low specificity. One-class classification is typically used if one class is sampled well, while the other class is heavily undersampled (Tax and Duin 2004). While our dataset is imbalanced, the minority class still has a large number of observations due to the size of the dataset. This could explain why other models showed better performance. Moreover, observations from the minority class might be too similar to the majority class to be considered as outliers in a one-class SVM model. One-class SVM is therefore not further applied in the analysis.

If all observations are predicted to be non-purchases, an accuracy of 96.23% would be achieved. Any method resulting in accuracy above that is thus considered to be adding value. The best results are achieved by RF, which also has the fastest training times (Table 2.3). RF achieves an accuracy of 96.95% (AUPR: 58.83%), using 10 variables available for splitting at each tree node and 500 trees. The model is able to predict almost 48% of all purchases and almost 99% of all non-purchases correctly. From all 'yes purchase' predictions, the model is correct approximately 63% of the time. However, the model seems to be overfitting, as the accuracy of the training data prediction is 100%. Accord-

Table 2.2: Pairwise correlations (numerical variables)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.00																	
2	-0.02	1.00																
3	0.78	0.09	1.00															
4	-0.02	-0.02	0.00	1.00														
5	-0.01	0.02	0.02	0.14	1.00													
6	0.03	0.46	0.05	-0.11	-0.01	1.00												
7	0.01	0.80	0.08	-0.04	-0.05	0.57	1.00											
8	0.66	-0.01	0.64	-0.02	-0.04	0.05	0.07	1.00										
9	0.39	-0.03	0.37	-0.03	-0.05	0.02	-0.02	0.49	1.00									
10	0.29	0.01	0.32	0.01	0.01	0.02	0.04	0.34	0.09	1.00								
11	0.29	-0.02	0.26	0,00	-0.02	0.01	0.04	0.59	0.11	0.13	1.00							
12	0.36	0.07	0.36	-0.01	-0.01	0.09	0.10	0.31	0.15	0.11	0.12	1.00						
13	0.32	0.10	0.60	-0.01	0.01	0.06	0.10	0.41	0.16	0.20	-0.01	0.21	1.00					
14	0.22	-0.07	0.03	-0.03	-0.08	0.01	-0.03	0.35	0.26	0.10	0.10	-0.02	-0.05	1.00				
15	0.17	0.02	0.20	0.00	0.01	0.01	0.02	0.14	0.07	0.05	0.06	0.08	0.09	-0.01	1.00			
16	0.19	-0.07	0.05	-0.02	-0.02	-0.01	-0.03	0.23	0.04	0.02	0.07	-0.02	-0.05	0.16	-0.01	1.00		
17	0.08	-0.02	0.02	0.00	-0.02	0.01	0.03	0.33	0.02	0.04	0.08	0.00	0.00	0.07	0.00	0.06	1.00	
18	0.16	-0.04	0.03	-0.01	-0.01	0.00	-0.01	0.26	0.05	0.06	0.06	0.00	-0.02	0.17	0.00	0.08	0.07	1.00
19	0.12	-0.03	0.03	-0.01	-0.01	0.00	-0.01	0.22	0.03	0.04	-0.01	0.00	-0.01	0.11	0.00	0.07	0.00	0.12
20	0.07	-0.01	0.02	-0.01	0.00	0.00	0.00	0.25	0.02	0.03	0.00	0.00	0.01	0.05	0.00	0.04	0.01	0.08
21	0.09	-0.02	0.02	-0.01	0.00	-0.01	-0.01	0.18	0.04	0.04	0.00	0.00	-0.01	0.10	0.00	0.04	0.00	0.56
22	0.13	-0.02	0.04	-0.01	-0.01	0.01	0.00	0.17	0.03	0.04	0.04	0.00	-0.01	0.07	0.00	0.10	0.03	0.07
23	0.22	-0.06	0.04	-0.02	-0.04	0.00	0.00	0.42	0.16	0.11	0.20	-0.01	-0.04	0.49	-0.01	0.17	0.13	0.19
24	0.51	0.04	0.61	-0.01	-0.01	0.04	0.05	0.52	0.44	0.15	0.11	0.24	0.28	-0.02	0.11	-0.02	-0.01	-0.01
25	0.13	-0.03	0.02	-0.01	-0.01	0.00	-0.01	0.19	0.04	0.04	0.04	0.00	-0.02	0.14	0.00	0.05	0.04	0.49
26	0.10	-0.02	0.03	-0.01	-0.01	0.00	0.00	0.22	0.03	0.04	-0.01	0.00	0.00	0.09	0.00	0.06	0.00	0.12
27	0.13	-0.03	0.04	-0.01	-0.02	0.00	-0.01	0.17	0.02	0.03	0.04	0.00	-0.02	0.08	0.00	0.13	0.03	0.06
28	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.00	0.02	0.00	0.00	0.02	0.00	0.01	0.02	0.02
29	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.06	-0.20	-0.05	-0.19	-0.20	0.07	-0.14	0.07	0.15	-0.02	0.01	-0.02	-0.07	0.19	-0.02	0.08	0.01	0.04

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19	1.00																
20	0.58	1.00															
21	0.09	0.08	1.00														
22	0.05	0.02	0.04	1.00													
23	0.10	0.09	0.15	0.23	1.00												
24	0.00	0.00	0.00	0.01	-0.01	1.00											
25	0.10	0.04	0.33	0.06	0.11	-0.01	1.00										
26	0.84	0.58	0.09	0.05	0.10	0.00	0.09	1.00									
27	0.04	0.02	0.03	0.64	0.22	0.01	0.05	0.04	1.00								
28	0.00	0.00	0.01	0.01	0.04	0.00	0.02	0.00	0.01	1.00							
29	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	1.00						
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	1.00					
31	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.12	0.06	1.00				
32	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00			
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	1.00		
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	
35	0.03	0.01	0.03	0.03	0.15	0.00	0.03	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 2.3: Results dataset 1: all customers (175k training observations)

		Validation data						Training data			
Method	Variable selection	Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR	Accuracy	Sensitivity	Specificity	Precision
RF	Set 5b	96.95%	47.51%	98.89%	62.57%	3.77%	58.83%	100.00%	99.94%	100.00%	100.00%
NN	Set 2	96.58%	38.74%	98.84%	56.73%	3.77%	48.95%	97.13%	45.63%	99.09%	65.61%
SVM	Set 2	96.51%	24.06%	99.35%	59.00%	3.77%	45.73%	97.46%	40.89%	99.61%	79.98%
LG	Set 5	96.42%	27.76%	99.11%	54.94%	3.77%	45.90%	96.57%	28.42%	99.16%	56.32%
RF	Add to cart	96.24%	3.47%	99.87%	51.56%	3.77%	7.60%	96.35%	3.76%	99.87%	52.75%

Table 2.4: Results dataset 1: all customers (larger training data size)

			Validation data						Training data			
Method	Variable selection	No. of training observations	Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR	Accuracy	Sensitivity	Specificity	Precision
RF	Set 5b	~870,000	97.20%	50.36%	99.04%	67.22%	3.77%	63.79%	99.98%	99.49%	100.00%	99.94%
NN	Set 2	~230,000	Algorithm does not converge									
SVM	Set 2	~350,000	96.54%	27.67%	99.24%	58.78%	3.77%	46.89%	97.33%	41.00%	99.51%	76.27%
LG	Set 5	~2,300,000	96.42%	28.23%	99.09%	54.95%	3.77%	46.48%	96.41%	28.04%	99.08%	54.38%

ing to Breiman (2001), RFs always converge so that overfitting is not an issue. To further assess this, we increase the minimum size of the terminal nodes. This lowers the prediction accuracy of the training data but does not improve the validation data prediction accuracy. We conclude that overfitting indeed seems to be no issue. NN results in 96.58% accuracy (AUPR: 48.95%), using one layer of five hidden neurons. SVM achieves 96.51% accuracy (AUPR: 45.73%), using a radial kernel, cost of 1, and gamma of 0.01, while LG results in an accuracy of 96.42% (AUPR: 45.90%). As the variable 'add to cart' is determined most important by the RF importance measure, we use RF to test whether solely using this variable would be sufficient to achieve high prediction accuracy. However, this leads to a much lower accuracy (96.24%) and an AUPR of 7.6%, indicating that the remaining list of predictor variables add substantial value in combination with 'add to cart'.

After increasing training data size, the accuracy of RF improves to 97.20% (AUPR: 63.79%) using ~870k observations (Table 2.4). The performance of NN cannot be improved, as larger training data sizes do not produce algorithm convergence. Increasing training size for SVM only shows a small improvement, resulting in 96.54% accuracy (AUPR: 46.89%). Lastly, the accuracy of LG remains as before (96.42%), with a slight increase in AUPR (46.48%). As RF outperforms all other models, it is used for the remaining analyses.

2.5.1.2 *Dataset 2: customers with high order frequency*

Prediction of customers with high order frequency using RF with an increased training data size shows an overall accuracy of 96.96% and an AUPR of 67.31% (Table 2.5). The accuracy should not be compared to the accuracy of dataset 1 as this dataset has a much higher prevalence, meaning more observations lead to purchases. Instead, AUPR is used for comparison, showing a higher value than for dataset 1 (Table 2.4), indicating that customers with high order frequency are easier to predict. In the subsequent sections, however, we continue to use dataset 1 to further test the application of AS across all customers.

2.5.1.3 *Dataset 3: impact of clustering on prediction accuracy*

Assessing the within-cluster variation shows that for more than five clusters, there is only a small reduction in within-cluster variation. K = 5 is thus chosen as the optimal number of clusters. Predicting five clusters separately leads to an overall prediction accuracy of 97.16% (AUPR: 63.29%) (Table 2.6), indicating that clustering does not improve model performance. Figure 2.3 shows how customers in those five clusters differ. For confidentiality reasons, the variable sign-up year is adjusted so that the earliest sign-up year corresponds to 'year 1'. Cluster 1 contains customers that signed-up several years ago and show an average order frequency. Decision time ranges from slow to fast. Cluster 2 consists of rather new customers that have not been buying much yet, while cluster 3 is composed of customers with high order frequency who make fast purchase decisions. The remaining two clusters are hardly noticeable in Figure 2.3 as cluster 4 contains customers that have not purchased anything yet and viewed very few products, and cluster 5 consists of customers with few product

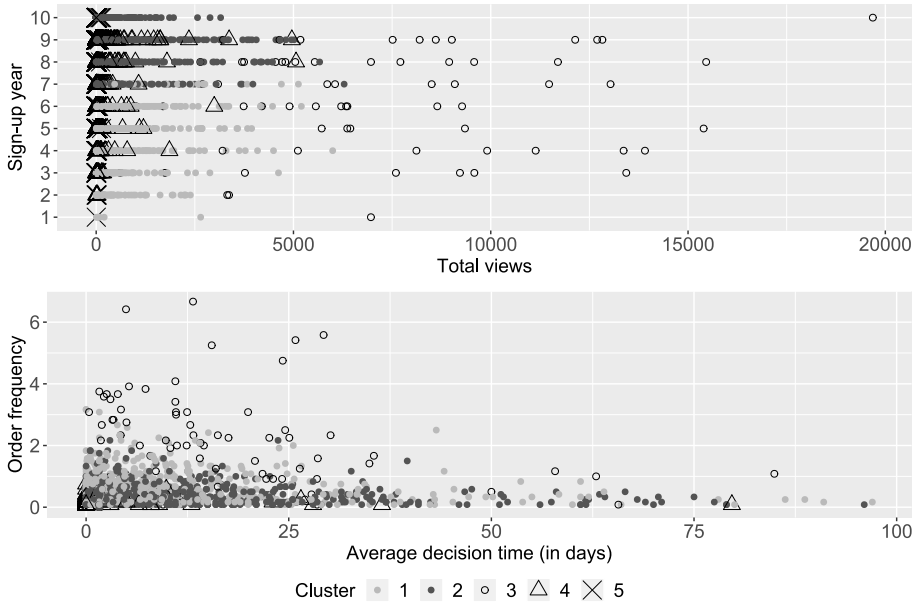

Figure 2.3: Cluster differences

views who bought 1-2 products on the same day they viewed the products for the first time.

2.5.2 *Prediction at different points in time and across product categories*

Successfully predicting customer purchases at the end of the first view date is not possible according to the results (Table 2.7). The prevalence of this dataset is much lower as the data does not contain orders that occurred on the same day as the first view date. Model accuracy is only 98.44% (AUPR: 20.83%), which is almost the same as predicting 'no purchase' for all observations.

Predicting a purchase right after an 'add to cart' click yields much better results (Table 2.7). Accuracy is 76.08% (AUPR: 76.46%) compared to an accuracy of 59.28% if 'no purchase' is predicted for all observations. In this dataset, prevalence is much higher as the data only consists of observations that contain an 'add to cart' click. To further improve model performance, an additional variable computing the 'add to cart' conversion is added, which measures the proportion of a customer's 'add to cart' clicks that led to a purchase. This results in an accuracy of 77.56% (AUPR: 81%). Those values outperform all previous results. In all three cases, clustering does not improve model performance.

Using the RF importance measure to investigate the mean decrease in accuracy per variable shows that 'add to cart' conversion is ranked as the most important variable, with a mean decrease in accuracy of 17.2%. The remainder of the five most important variables consists of the month a customer viewed a product for the first time, the number of product page visits, the customer's segment, and order frequency. Interestingly, most variables contributing to the prediction accuracy of RF are variables describing the customer (e.g., segment,

Table 2.5: Results dataset 2: customers with high order frequency (RF with variable set 2)

No. of training observations	Validation data						Training data			
	Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR	Accuracy	Sensitivity	Specificity	Precision
~870,000	96.96%	56.87%	98.79%	68.32%	4.37%	67.31%	99.10%	82.36%	99.87%	99.61%

Table 2.6: Results dataset 3: clusters (RF with variable set 5b)

Cluster	No. of training observations	Validation data						Training data			
		Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR	Accuracy	Sensitivity	Specificity	Precision
1	~210,000	95.74%	54.38%	98.31%	66.68%	5.85%	64.41%	99.97%	99.54%	100.00%	99.95%
2	~230,000	97.41%	44.28%	99.14%	62.70%	3.16%	56.15%	99.99%	99.59%	100.00%	100.00%
3	~220,000	98.11%	44.25%	99.36%	61.61%	2.26%	55.86%	99.99%	99.63%	100.00%	99.96%
4	~200,000	100.00%	25.00%	100.00%	50.00%	< 0.01%	19.51%	100.00%	100.00%	100.00%	100.00%
5	~12,000	87.44%	93.71%	74.31%	88.42%	67.68%	95.17%	99.00%	99.79%	97.42%	98.71%
Overall results		97.16%	46.61%	98.92%	63.08%	3.77%	63.29%	99.98%	99.61%	99.99%	99.97%

Table 2.7: Results at the end of first view date and after 'add to cart' click (RF with variable set 5b)

		Test data						Training data			
Dataset	No. of training observations	Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR	Accuracy	Sensitivity	Specificity	Precision
First view date	~580,000	98.44%	1.39%	99.99%	61.97%	1.56%	20.83%	99.47%	66.00%	100.00%	99.98%
Add to cart	~440,000	76.08%	68.18%	81.51%	71.69%	40.72%	76.46%	99.23%	98.73%	99.57%	99.36%
Add to cart*	~440,000	77.56%	65.89%	85.57%	75.82%	40.72%	81.00%	99.17%	98.42%	99.68%	99.53%

*Including variable 'add to cart' conversion.

Table 2.8: Results after 'add to cart' click per product category (RF with variable set 5b)

Product category	Accuracy	Sensitivity	Specificity	Precision	Prevalence	AUPR
Women's clothing	77.81%	65.98%	85.28%	73.89%	38.71%	79.90%
Men's clothing	78.98%	80.49%	77.60%	76.82%	47.98%	87.63%
Textile	77.93%	71.46%	82.58%	74.71%	41.85%	82.41%
Shoes	79.77%	66.40%	88.25%	78.20%	38.83%	84.19%
Loungewear/swimwear	80.45%	78.05%	82.21%	76.24%	42.24%	87.64%
Sports clothing	80.67%	75.08%	84.49%	76.76%	40.56%	87.00%
Accessories	81.49%	67.91%	89.31%	78.54%	36.55%	85.71%

order frequency) or the product viewing process (e.g., time spent on a product site), while most of the variables measuring clicks on a product site, except for 'add to cart', do not seem to be particularly relevant for the prediction. This is an interesting addition to the set of variables tested by Van den Poel and Buckinx (2005), as their list does not cover specific click types or the time of the first view of a product.

So far, we predicted customer purchases across all product categories. To assess whether there is a difference in predictability, we separately use RF after a first 'add to cart' click for different product categories. Table 2.8 shows the results of the test data. When separately forecasting women's and men's clothing, results show that the AUPR for men's clothing is much higher, indicating that these products are easier to predict. Moreover, the data for women's clothing shows a lower prevalence, suggesting that women purchase fewer of the products they viewed. The separation of the product portfolio into the categories textile, shoes, loungewear/swimwear, sports clothing, and accessories, does not show such a substantial difference in AUPR, with loungewear/swimwear and sports clothing having the highest AUPR. The RF importance measure shows that a similar set of variables is relevant for all product categories, except for the month a product was viewed for the first time playing a more important role for loungewear/swimwear.

2.5.3 *Application to the future*

As explained in Section 2.3.1, the split into training, validation, and test data that has been applied so far is based on the fact that we assume that the purchase behavior of a customer does not substantially change over time. To test whether this assumption holds true, the dataset is split into training and test data that respect the temporal order of observations, meaning that the training data now consists of observations with a first view date in the first half of the one-year time period, while the test data contains observations with a first view date in the last quarter. Using RF with the original training data size of 175k observations, an AUPR of 64.60% can be achieved, compared to 58.53% from dataset 1 using RF. This indicates that the prediction model is also applicable to future data.

2.5.4 *How anticipatory shipping improves delivery time*

To estimate the impact, the results first have to be translated into products sent (in-) correctly. For 100k customers, the RF results of dataset 1 would have led to 157k products being sent in advance correctly within one year, and 77k products would have been sent without the customer buying it, creating unnecessary logistics cost. Essentially, for every 100 products sent correctly, 49 are sent erroneously. A share of the latter could eventually be bought by a different customer from a similar region, mitigating the cost of products sent incorrectly. This will be assessed in the simulation of inventory and order fulfillment. Moreover, shipping products to a different location might result in insufficient stock for purchases from the region those products were originally shipped from.

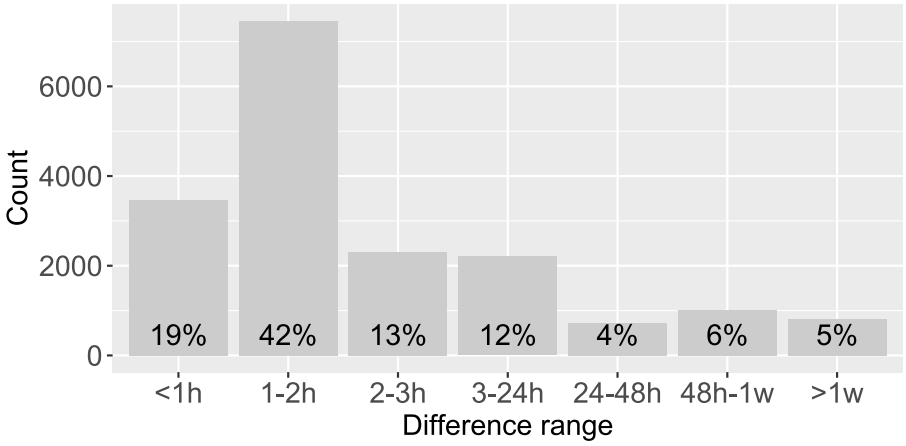

Figure 2.4: Time difference between 'add to cart' click and purchase for correctly predicted purchases

Due to a lack of data regarding historical inventory development, the impact of this cannot be estimated with the given dataset. The best impact is achieved when predicting after an 'add to cart' click (including the new variable 'add to cart' conversion). 169k products would have been sent correctly and 54k incorrectly, translating into only 32 products sent by mistake for every 100 products sent correctly. To lower the cost of erroneously sending products, the impact of different thresholds can be tested. A 90% threshold would have led to only 11 products sent erroneously for every 100 products sent correctly. Increasing the threshold, however, also leads to a smaller number of purchases identified (60k instead of 169k).

When investigating the delivery time savings of predicting after an 'add to cart' click, meaning the time saved between an 'add to cart' click and actual purchase, it becomes apparent that the time difference is often too short to send a product closer to the customer (Figure 2.4). Only ~15% of purchases predicted correctly would have resulted in delivery time savings of more than one day.

To simulate the effect of anticipatory shipping, we use the prediction after an 'add to cart' click in the test dataset as this delivered the best results. The dataset contains approximately 19k customers with an 'add to cart' click. We assume that this subset of customers could represent the case study company's premium customers for which an anticipatory shipping model is implemented. Products can only be sent if the delivery address of the customer is already known. It should be noted that for 60% of the correctly predicted purchases, a delivery address is not known at the time of prediction. The reason for this can be that the delivery addresses of new customers are not known yet. Encouraging website visitors to sign up early and provide a future delivery address, as well as leveraging Google Analytics' location reporting, could mitigate this issue. For premium customers whose delivery address is known, RF predicts 584 purchases from 419 different customers in the two-week planning horizon.

Results show that without anticipatory shipping, the orders of the premium customers in the two weeks have an average delivery time of 14.2 h, with 22% of the orders delivered from the warehouse closest to the customer. With antic-

ipatory shipping and a 48-hour reservation window, the delivery time can be reduced to 13.4 h, with 48% of orders delivered from the warehouse closest to the customer. However, for 47% of those orders, the product was still in transit to the warehouse at the time the order was placed, with an average wait time of 5.5 h until the product arrives in the warehouse closest to the customer. When applying anticipatory shipping only for customers that historically took over 8 h on average between the 'add to cart' click and subsequent purchase, this percentage can be reduced to 43%, but average wait time remains the same. This again highlights the challenge of the close time proximity between prediction and purchase.

From 584 products predicted for purchase, only 25 were sent to another warehouse without subsequent purchase from customers located in zip codes allocated to the same warehouse. This number is quite small and does not even take into account that the products might be purchased soon after the two-weeks. This shows that while anticipatory shipping might lead to many unnecessary products sent in advance when applied to all customers, using the prediction for just a subset of customers could mitigate this disadvantage of AS. This is because many products wrongly sent to another warehouse are in fact purchased by a different customer in the same geographical area.

2.5.5 *Managerial implications*

The results show that while the purchase of a customer is, to a certain extent, predictable, delivery time savings from anticipatory shipping are difficult to achieve due to the short time between prediction and purchase. The location of the retailer's warehouses and customers also plays a major role in this. For retailers with few warehouses and a widespread customer base, resulting in long transportation times, AS could be very difficult to implement. Nevertheless, studies have shown that faster delivery times lead to a lower number of returns (Bergmann 2018). The cost from wrongly predicted purchases or correctly predicted purchases arriving too late at the warehouse could potentially be reduced through savings in product returns. Additionally, time savings from predictions could be leveraged for anticipatory picking and packaging, instead of anticipatory shipping, for instance, to reach same-day delivery cut-off times.

The case company will use the results of this research to estimate the business case for AS for varying threshold levels. Like many fashion retailers, the case company has a large volume of order movements between warehouses to avoid sending orders containing several items in various parcels. Implementing AS could also have a positive effect on the number of order movements.

In terms of application, we believe that the approach should be equally applicable to other online fashion retailers. Variables that were listed as most important by RF are all data points that other fashion retailers should be able to obtain. A limitation here is that retailers need to have sufficient website traffic (i.e., customers, clicks, and orders) to generate enough observations for the algorithms to deliver meaningful results. Being able to forecast customer purchases could have many areas of application besides AS. Especially in the fast fashion industry, it could be leveraged to reorder products, which are likely to

sell out quickly, in advance. Also, it could be used in returns management to redistribute returns to warehouses where sales are likely to occur in the near future. Whether the proposed prediction model can be used in other fields cannot be answered with this study. The purchase decision for a fashion item might be very different from other types of products.

Lastly, it should be noted that the prediction model is fully dependent on the quality and reliability of the input data. Issues in tracking, for example, due to programming errors, customers not being logged in, or click types frequently being renamed, are typical challenges that can arise in trying to obtain reliable data.

2.6 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This paper shows how forecasting methods can be applied to predict customers' future purchases to generate delivery time savings. Logistic regression, neural network, random forest, and (one-class) support vector machine are applied. Random forest strongly outperforms all other models in terms of accuracy and speed, indicating that the other models are not suitable in this context. When predicting purchases after an 'add to cart' click using RF, approximately 66% of purchases are correctly predicted. Clustering as input for forecasting does not lead to accuracy improvements.

Interestingly, results show that purchases of men's clothing are easier to predict than purchases of women's clothing. Moreover, customers with a high order frequency seem more predictable than other customers. From the results, we conclude that online purchases are, to a certain extent, predictable, but AS still comes at a high cost. Due to the low number of product site visits that convert into purchases, even a 99% accuracy of predicting those non-purchases correctly results in many products wrongly sent in advance. For the case company, the model would have resulted in 169k products correctly sent in advance throughout the year, and 54k products sent incorrectly.

A simulation of inventory and order fulfillment over two weeks for a set of premium customers shows a reduction in average delivery time from 14.2 h to 13.4 h due to anticipatory shipping. At the same time, almost half of all products sent correctly in advance would not reach the warehouse closest to the customer before the customer places the order, emphasizing the difficulty of implementing AS due to the short time between prediction and purchase. Nevertheless, only a small portion of products wrongly sent in advance are not purchased by another customer in the same region within the two weeks, hence the advanced shipment did eventually benefit a different customer.

AS generally leads to high logistics cost, despite good prediction accuracy. In combination with the fact that for only 15% of all correctly predicted purchases, the time saved would have been more than a day, the model could potentially be better leveraged for alternatives to AS, such as anticipatory picking or packaging.

The results of this study are limited by the data quality provided. Improved data quality would likely result in higher prediction accuracy. 18% of the orders in the data do not have an 'add to cart' click, which is necessary to purchase

a product. Customers not being logged in or not registered yet are most likely the main causes of this. With 'add to cart' clicks playing a vital role in the prediction using RF, such data issues clearly limit model performance. Moreover, only a subset of forecasting methods available for classification is applied in this research. Other methods not tested in this study could potentially result in higher prediction accuracy and lower cost for AS, although the literature suggests limited improvements in accuracy through the application of related methods (Verbeke et al. 2012). Similarly, results from clustering are limited by the fact that this research does not fully explore possibilities for clustering, such as applying other clustering algorithms or using categorical variables as input to clustering.

In terms of future research directions, five areas of interest can be highlighted. First, this research only studies to what extent online purchases can be predicted. Understanding in which quantity and size a customer will buy a product is additional input required to enable AS. Second, being able to predict when a purchase will occur could help determine if delivery time savings are sufficient to send a product in advance. Third, additional data related to pricing, marketing campaigns, and fashion trends, among others, could be further studied to assess their impact on prediction accuracy. Fourth, increasing the forecast horizon could help better capture seasonality trends. Lastly, further research could be conducted to better understand the underlying purchase patterns of customers. This could be used to improve prediction accuracy, for instance through the construction of better customer clusters.

This study has shown that clickstream data can be useful to predict the future orders for individual customers. Whether a similar effect can be found on a more aggregated prediction level will be the focus of the next chapter. Specifically, Chapter 3 investigates whether clickstream data adds value to the product-level demand forecast for the case study partner. Typically, product-level demand forecasts leverage historical sales as input data. Based on the findings of this chapter, we hypothesize that using clickstream data as additional input in the forecasting process helps to improve overall forecast accuracy.

3

THE VALUE OF CLICKSTREAM DATA IN PRODUCT DEMAND FORECASTING

The following chapter is based on Weingarten and Spinler (2020b).

3.1 INTRODUCTION

E-commerce has grown significantly in recent years, with a 47.5% increase in Western Europe from 2015 to 2019 (Centre for Retail Research 2020). The recent Covid-19 pandemic and the resulting lockdown of physical stores in several countries has fueled this trend. Studies show an increase of nearly 10% in e-commerce sales of apparel, department store goods, and beauty products since the beginning of the pandemic (Briedis et al. 2020). The Centre for Retail Research (2020) expects the e-commerce retail market to have grown by as much as 31.1% by the end of 2020, putting pressure on online retailers to deliver rising numbers of orders, often consisting of limited order lines and small quantities, without sacrificing customer experience (Boysen et al. 2019).

In an attempt to attract customers, online retailers are offering wide assortments of products, many with lumpy and volatile demand, which do not drive high sales volumes but increase their revenues (Boyd and Bahn 2009, Morton 2017). As of April 2019, Amazon offered almost 120 million products to its customers (ScrapHero 2019). Combined with ever-increasing customer expectations regarding service and delivery times (e.g., same-day delivery), this makes the physical fulfillment process in e-commerce especially challenging. While the trend toward e-commerce is expected to continue, order fulfillment has been identified as a key bottleneck in online retailers' supply chains, due to order handling inefficiencies in warehouses and distribution networks (Leung et al. 2018). This explains why the last leg of the delivery process (i.e., the last-mile) is often considered to be one of the main challenges in e-commerce supply chains (Leung et al. 2018). Furthermore, for omnichannel companies that sell products through both online and offline channels, efficient warehouse operations are a prominent theme in the literature. Unsurprisingly, as a lever to improve operational performance and optimize delivery speed, demand forecasting has become essential for both purely online and omnichannel retailers.

The topic of demand forecasting has already received extensive academic attention. Typically, demand forecasts leverage information from historical sales to forecast future demand. However, limited historical sales data is available for a large proportion of online retailers' product assortments, owing to small numbers of daily sales and short product lifecycles. On average, online retailers lose half of their product portfolio every year due to stockouts and product end of life, amongst other reasons (Morton 2017). Zalando reports that as many as 95% of its products are new every season (Zalando 2018). Forecasting demand in e-commerce is therefore especially challenging. In the last decade, as a re-

sult of companies' increased ability to store and process large data volumes, the inclusion of additional data in demand forecasting has become a focus of the literature. While variables such as promotion campaigns and static product information have frequently been included in demand forecasting (e.g., Chong et al. 2016, Qi et al. 2019), the emergence of big data provides even more possibilities to include additional information. Compared with brick-and-mortar stores, e-commerce players are in a unique position to collect vast amounts of data on their customers, and more specifically on their online browsing and purchasing behavior. However, whether including such data in the demand forecasting process improves forecast accuracy has received limited attention in the literature.

The forecasting and warehouse operations of a leading online fashion retailer inspired our research. Using a unique dataset containing historical sales and customer clickstream data detailing the product sites visited by customers and their behavior on these sites (e.g., image clicks, view duration, etc.), we investigate the effect of clickstream data on demand forecasting. We add to existing literature on demand forecasting in e-commerce by assessing the extent to which clickstream data can help improve forecast accuracy. Specifically, we analyze which variables derived from clickstream data are best suited to forecasting demand. While other studies have assessed the effect of clickstream data on e-commerce predictions, these have focused mainly on individual customer-level predictions, typically for marketing purposes, rather than on forecasting demand for a company's product assortment. Moreover, building on commonly used variables in this context, we apply feature engineering to define novel variables from clickstream data for our prediction, such as the number of product site visits lasting longer than the average viewing time for a product. To the best of our knowledge, this is the first study to investigate the forecasting power of this set of variables. We also apply clustering to identify particular products for which using clickstream data is especially useful for forecasting. Our results suggest that clickstream data can significantly improve forecast accuracy, especially for medium- and certain intermittent-demand products. The best forecast results are achieved when using a support vector machine model with either the number of 'add to cart' clicks or the number of unique product site visits in combination with the historical conversion rate of these unique visits.

Our research also contributes to existing demand forecasting literature in the fashion industry. The development of advanced forecasting methods to forecast fashion demand is not novel (Ni and Fan 2011, Thomassey 2010), but few studies have examined the effect of clickstream data on forecast accuracy in this context. Our results will be of interest both to purely online sellers and to omnichannel retailers with an online sales channel. For the latter, our approach can be used to improve forecasts and operations for the online channel, or alternatively, as shown by Huang and Van Mieghem (2014), future studies might assess the extent to which using clickstream data in forecasting might also help to improve demand forecasting for offline channels. Specifically, use of location-specific clickstream data (i.e., assessing the geographical regions from which clickstream data arises) may benefit forecasts of in-store demand.

To quantify the impact of improvements to forecast accuracy, we use a case study to test how our forecast models affect picking times in a warehouse, assuming that products may be stored in either a capacity-constrained reserve area with fast picking times, or in a storage-efficient back area with slow picking times. The reserve area resembles a pocket sorter (PS), a novel system for warehouse operations on which limited research has so far been conducted. To estimate savings in picking time, we simulate the application of a greedy heuristic in which decisions on where items are stored are guided by product prioritization resulting from the forecast. The findings suggest that when using a PS for low- to medium-demand products, up to 36% more product orders are picked from the PS when using a machine-learning forecast with clickstream variables, compared with a time-series forecast based solely on historical sales.

The remainder of this paper is structured as follows: Section 3.2 provides an overview of related literature. In Section 3.3 we present our methodology, including the forecasting and clustering methods, and an introduction to the simulation used to measure the impact of the forecast results. Section 3.4 outlines the case study context and dataset, and summarizes the findings from our initial data analysis. Section 3.5 presents both forecasting and simulation results, and Section 3.6 concludes with managerial implications and an outlook on future research.

3.2 RELATED LITERATURE

Forecasting in e-commerce takes place at different levels of aggregation, ranging from market level to product level. Market-level forecasts usually predict total sales, often for strategic purposes such as to identify trends. In contrast, product-level demand forecasts involve forecasting many different individual products, and are typically used to develop short-term forecasts for use in daily operations, such as inventory management or pricing (Fildes et al. 2019). In this study, we focus on product-level forecasting.

The e-commerce industry is generally a very dynamic and complex environment, characterized by:

- Large, fast-changing product assortments, with thousands of different stock-keeping units, varying in color and size (Bandara et al. 2019, Boyd and Bahn 2009, Morton 2017).
- Intermittent, volatile product demand with strong seasonality (Bandara et al. 2019, Thomassey 2010).
- Substantial effects of external factors on sales (e.g., holidays, consumer trends) (Loureiro et al. 2018, Thomassey 2010).
- High levels of substitution between products (Bandara et al. 2019, Qi et al. 2019).

Because of this complexity, sales data in e-commerce is often highly non-stationary, meaning that the process of generating sales data time series is not stable over time (Hyndman and Athanasopoulos 2018). Therefore, the e-commerce domain requires sophisticated forecasting methods that can handle these challenges.

3.2.1 *Use of novel data sources for predictions in e-commerce*

Owing to the complexity of demand forecasting in e-commerce, the use of novel data sources to improve demand forecasts has become a popular research topic. Lau et al. (2018) enhance demand forecasting by using sentiment analysis to extract valuable information from customers' product comments. Steinker et al. (2017) improve aggregated demand forecasts by including weather forecasts in their predictions. Kulkarni et al. (2012) investigate the power of using customer search terms in forecasting. One stream of literature particularly investigates how to leverage customers' browsing behavior for predictions in e-commerce, with demand forecasting as one potential application. Most of this research focuses on making predictions at an individual customer-product level (e.g., Iwanaga et al. 2016, Nishimura et al. 2018, Sismeiro and Bucklin 2004, Weingarten and Spinler 2020a). However, its main focus is typically on applications in marketing, such as personalized marketing services (Kim et al. 2005, Lo et al. 2016), price optimization (Ferreira et al. 2016), or web-page optimization (Moe and Fader 2004). Cirqueira et al. (2020) provide a detailed review of studies using customers' browsing behavior for predictions in e-commerce.

Several studies use customer browsing behavior in a demand forecasting context. For instance, Yeo et al. (2016) predict a customer's intent to purchase a specific product and forecast total product demand through the sum of customers likely to purchase. They conclude that the browsing ratio (i.e., how often a customer visits a particular product site compared with other products' sites) and browsing duration helps to improve model accuracy. Huang and Van Mieghem (2014) combine clickstream and historical purchase data to predict demand for offline orders to improve inventory management, showing that a customer's cumulative number of website visits provides valuable information for demand forecasting. Van den Poel and Buckinx (2005) investigate the effect of various variables derived from clickstream data to predict whether a customer's website visit will result in a purchase, showing a significant effect for variables such as the number of past visits and the total number of historical clicks. Similarly, Guan et al. (2020) use classification methods to predict whether a shopping session will generally result in a purchase, and how early this prediction can accurately be made. However, the relevance of their results to our study is limited because their dataset was collected in the context of a mega sale event. Qi et al. (2019) use another set of variables extracted from clickstream data (e.g., the number of page views, unique customer visits, etc.) for product-level forecasting. Although they do not present results for individual clickstream variables, they do report improvements in forecast accuracy through their inclusion.

Our research builds on many of the results from the aforementioned studies. However, we formulate our demand forecasting endeavor as a regression problem in order to forecast demand at a product level, rather than making predictions for individual customer-product combinations. In this context, we use common variables summarized in previous work, and add to these by proposing a set of new clickstream variables for evaluation.

3.2.2 *Forecasting methods in demand forecasting*

Traditional time series models (e.g., exponential smoothing) are among the most widely used methods in demand forecasting. However, they are often not considered the most suitable in an e-commerce setting because they cannot model the non-linearity and volatility often present in e-commerce sales data (Fildes et al. 2019). Essentially, historical observations of time series are used to develop models that describe the underlying structure of the series. As such, time series models are univariate by nature, meaning that they predict future observations for one time series at a time, and cannot learn across multiple time series (Salinas et al. 2020). However, e-commerce assortments tend to have a hierarchical structure, meaning that products in a subcategory may be similar, and forecasts may benefit from models that can capture shared features across time series (Bandara et al. 2019). Also, time series models are typically unable to accommodate additional variables as inputs into their predictions. With advancements in computing technology, machine learning models have emerged as an alternative to time series models for forecasting. These can capture much more information in the forecasting process, model non-linear data patterns, and are able to learn across sets of related products to make predictions of future demand (Bandara et al. 2019). Compared with time series models, their main drawbacks are typically long computation times, a requirement for sufficient historical observations for model training, and limited interpretability of their results (Petropoulos et al. 2018). Popular machine learning models that have been shown to provide satisfactory forecasting results in comparison with traditional methods include artificial neural networks, decision trees, and support vector machines (Carbonneau et al. 2008, Ferreira et al. 2016, Guan et al. 2020, Qi et al. 2019). Recurrent neural networks, a specific variant of artificial neural networks, have become well-known in time series forecasting, as their recurrent connections make them especially suited to modeling sequenced data such as product sales (Bandara et al. 2020b).

To further improve the accuracy of the above-mentioned models, clustering techniques have been introduced, which are able to reflect product characteristics in the forecasting process. Clustering techniques group products based on a selected set of features in order to derive homogeneous subsets of products (Bandara et al. 2020b). Ideally, forecasting these product clusters separately provides superior forecast accuracy, as shown by Bandara et al. (2020b) and Chang et al. (2009), amongst others.

3.2.3 *E-commerce warehouse operations*

As we examine the impact of our forecast models on e-commerce warehouse operations, existing literature in this domain is also relevant. To ensure rapid delivery of online purchases, logistics activities such as picking, packaging, and distribution must be well managed. Among these logistics activities, order picking is typically considered to be the most labor-intensive (Leung et al. 2018). Several research streams address this topic in the context of e-commerce operations, ranging from solutions for warehouse layout designs (e.g., Hübner et al.

2015) to efficient storage assignment (e.g., Yang et al. 2015) and scheduling (e.g., Zhang et al. 2016). Warehouses typically store inventory in at least two areas: a reserve area for efficient storage and a forward area for efficient order picking (Gu et al. 2010). Determining which products to store in what quantities in the two areas is known as the forward-reserve problem (Gu et al. 2010). In e-commerce, warehouses often lack lightweight solutions suitable for online orders (Leung et al. 2018). To tackle this problem, novel warehousing systems have emerged, one of which are PS systems. Pocket sorters move individual units of inventory in bags, which are moved to and from storage to packing stations on request, automatically sorting units into the right order for packaging (Boysen et al. 2019). It is necessary to determine how many units of which products to store in the PS, so replenishing these systems can be considered to be a specific type of forward-reserve problem. As the PS concept is novel in order fulfillment, limited research has been undertaken on this topic.

3.3 METHODOLOGY

Our research investigates the value of clickstream data in daily demand forecasting using data collected from an online fashion retailer. Specifically, we develop two-day- and seven-day-ahead forecasts, which can be used to make short-term improvements to warehouse operations. Multistep-ahead forecasts also enable us to assess whether the predictive performance of clickstream data diminishes over time. We follow a five-step approach. First, we apply linear regression to gain a first indication of whether clickstream data helps explain variation in future demand. As a baseline model, we perform linear regression using historical sales two and seven days ago, combined with dummy variables. We then add clickstream variables, referred to as the clickstream model. In the second step of our research, we establish a forecast baseline using only historical sales to determine which forecasting methods are most suitable for our dataset. Third, we analyze the effect of including variables based on clickstream data in the forecasting process. As the results of some of the methods applied (e.g., neural networks) are difficult to interpret, we first include additional variables one at a time to determine the effect of each variable on forecast accuracy, and then test combinations of various variables based on our initial findings. In the fourth step, to further improve the effect of including clickstream data in the forecast, we apply clustering, grouping similar products to forecast them separately. Lastly, we assess improvements to forecast accuracy using an order picking simulation. All models applied in this research are implemented in R (version 3.6.2).

3.3.1 *Prediction methods*

While time series methods are typically unable to accommodate external factors in their predictions, dynamic regression (DR) models, an extension of autoregressive integrated moving average (ARIMA) models, are an exception. Specifically, we use a DR model with ARIMA errors to allow for the inclusion of clickstream variables. We focus on a selection of common machine learning

algorithms that are well-known in the context of regression and have shown satisfactory results in previous demand forecasting research. From ensemble learning methods, we apply random forest (RF), which is a supervised learning method that has gained popularity in academia because RFs are simpler to train and easier to interpret than most other machine learning methods (Hastie et al. 2009, Maimon and Rokach 2010). An RF consists of an ensemble of decision trees, where at each node, a randomly selected subset of variables is considered, in order to split the node into two or more daughter nodes in a manner that improves the homogeneity of the daughter nodes compared with the parent node (Breiman 2001). From the class of artificial neural networks (NN), we apply the a multilayer perceptron, a feed-forward neural network which consists of multiple neurons connected by functions and weights. Backpropagation is used to fit the network to the training data (Maimon and Rokach 2010). As previously mentioned, recurrent neural networks (RNN) are particularly well-suited to model time series data. In our research, we apply two well-known forms of RNN, namely the Jordan recurrent neural network (JRNN) and the Elman recurrent neural network (ERNN) (Elman 1990). Support vector machine (SVM) models are well-known in the context of classification. SVM maps data points to a high-dimensional space using a non-linear function to enable the computation of a linear model (James et al. 2013). It then estimates the hyperplane that fits the data in a manner that minimizes the error rate (Vapnik 1999). SVM can also be used for regression, often referred to as support vector regression (SVR). For all machine learning methods, we use grid search to estimate the values of hyperparameters.

3.3.2 *Model assessment*

To evaluate out-of-sample forecast accuracy, we divide the data into training and testing sets. We denote the training period as $\{1, ..., T\}$ and the testing period as $\{T + 1, ..., T + N\}$, where the last 21 days of data are used for out-of-sample testing (i.e., $N = 21$). The whole period includes no major holidays or events. We apply time-series cross-validation, using a series of 14 test sets, each consisting of one observation for the two-day- and seven-day-ahead forecasts (see Figure 3.1). As an illustration, to forecast demand for day T+2, we use all historical observations of sales and clickstream variables up to day T. DR also requires forecasts of additional variables as inputs. We use a naïve forecast to predict future values of the clickstream variables. In this context, a naïve forecast means that the predicted values for T+2 and T+7 equal the actual values at T for the two-day- and seven-day-ahead forecasts, respectively.

To assess forecast accuracy, we use the root-mean-square error (RMSE) and the mean absolute scaled error (MASE). Although the mean absolute percentage error (MAPE) is well-known in the context of demand forecasting, it is infinite or undefined for zero-demand periods, and is therefore unsuitable for datasets with intermittent-demand products (Hyndman and Athanasopoulos 2018). Using both RMSE and MASE helps us understand for which products the inclusion of clickstream data in the forecasting process is especially useful. The RMSE is a scale-dependent measure, as the error term is on the same scale

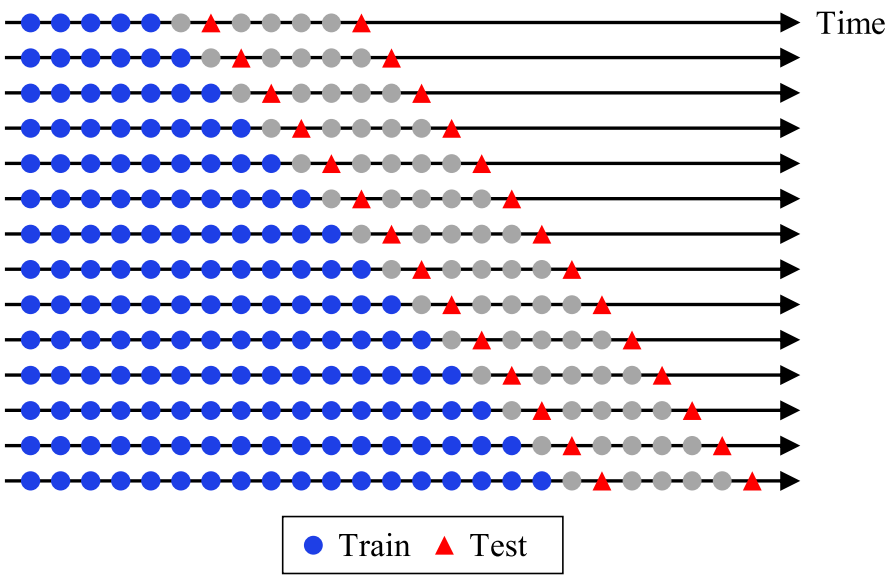

Figure 3.1: Cross-validation

as the data, whereas the MASE is scale-independent, comparing the forecast error of the test set against the forecast error from applying a naïve forecast to the training data (Hyndman and Athanasopoulos 2018). Values below one indicate that the forecast for the test data performs better than a naïve forecast for the training data. As we calculate the RMSE and MASE for all products in our dataset and then take an unweighted average across all products, large improvements in RMSE combined with small improvements in MASE may, for instance, indicate that forecast accuracy has mainly improved for high-demand products. For each product, the RMSE and MASE are computed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{J}\sum_{1}^{J}(y_j - \hat{y}_j)^2} \tag{3.1}$$

$$\text{MASE} = \frac{\frac{1}{J}\sum_{j}^{J}|y_j - \hat{y}_j|}{\frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}|} \tag{3.2}$$

where J, a subset of the test set period, is equal to the 14-day rolling forecast horizon, as we want to compute the forecast accuracy by averaging over the different test sets, while y and $\hat{y}$ denote actual and forecast demand, respectively. For the MASE, the denominator is the mean absolute error of the naïve forecast from the training data, where m equals 2 for the two-day-ahead and 7 for the seven-day-ahead forecast.

3.3.3 *Clustering*

According to Kulkarni et al. (2012), product characteristics may influence both browsing and sales behavior, creating heterogeneity in the data. While machine learning methods are able to learn from various products across sales time series, it can be difficult to identify shared information across these time series

if the data is heterogeneous. Therefore, clustering techniques are introduced into the forecasting process. Two commonly used clustering techniques are hierarchical and K-means clustering. The former structures the data hierarchically, enabling flexible selection of the number of clusters, and consequently analysis of the obtained clusters (James et al. 2013). K-means clustering, on the other hand, requires the number of clusters to be pre-specified, and assigns data points to clusters by minimizing within-cluster variation. For this, each data point is initially assigned randomly to a cluster, and then the data points are re-assigned based on calculating the mean of each cluster. K-means has the disadvantage that the number of clusters must be specified in advance, and the algorithm may lead to local optima due to the initial random allocation of data points. However, K-means is known to be less computationally expensive and to work better than hierarchical clustering for very large datasets (Maimon and Rokach 2010). Given the large product assortments of many online retailers and the computationally complex task of daily forecasting, we apply K-means clustering in our study, and forecast the resulting clusters separately using machine learning methods. Other studies involving demand forecasting have already successfully used K-means to improve forecast accuracy (e.g., Chang et al. 2009, Chen and Lu 2016, Thomassey and Happiette 2007).

To apply K-means, we use two different feature sets to construct clusters:

1. Clustering based on demand characteristics: We use 13 different time series features for clustering (e.g., lumpiness and linearity of sales), calculated based on the last eight weeks of the training data. The selected features are a subset of the time-series features outlined in Hyndman et al. (2015). Further details are given in Appendix B.2.

2. Clustering based on clickstream data and total demand: To assess whether the usefulness of clickstream data for demand forecasting depends on products' demand volume and conversion characteristics, we use each product's total demand in the last eight weeks of the training data and the conversion rate of clickstream data as inputs into clustering. The latter is calculated by taking the training data and dividing the total sum of a clickstream variable, such as unique customer visits to a product site, by the total number of sales of a product.

Before applying K-means clustering, we perform principal component analysis (PCA) to the feature sets to reduce dimensionality. To determine the optimal number of clusters, we use the silhouette method, which measures how close observations in one cluster are to observations in neighboring clusters. The silhouette width indicates how well each observation lies within the assigned cluster. K-means is performed for different numbers of clusters, and the optimal number of clusters is the one that maximizes the average silhouette width (Rousseeuw 1987).

3.3.4 *Impact estimation: application to warehouse operations (order picking)*

To estimate the impact of our forecast models, we simulate the picking of individual product orders, hereafter referred to simply as orders, using a PS for

14 days and the two-day ahead forecast for a dataset containing K products. Throughout the time period, a set of O orders is received and must be picked from the warehouse. Products can be picked from a back area with slow picking times, or a forward area, resembling a PS system, with fast picking times. Products are internally replenished to the PS according to a daily product forecast, F_{kt}. Ideally, picking times can be reduced through improvements to forecast accuracy. We assume that there is always sufficient inventory in the back area for replenishment, while the PS is capacity constrained. We test capacities of 5%, 10%, and 15% of average daily orders. As picking times may vary depending on product characteristics, warehouse design, etc., we refrain from making assumptions concerning picking times for the back and forward areas, but instead calculate the percentage of products picked from the PS throughout the 14 days.

At the beginning of the planning horizon ($t = 1$), the forecast F_{kt} is sorted according to forecast volume, and then the PS is filled according to this prioritization until its maximum capacity is reached. For each product, a maximum number of $m = 3$ units can be stored in the PS to avoid filling the system mainly with a few high-demand products. A list denoted as the remaining forecast R_{kt} is used in the simulation, which resembles the forecast adjusted for product units that have already been replenished to the PS on a given day and those that have already been purchased and forecasted but had been stored in the back area. For every order received on day t at hour h, we check whether there is available inventory in the PS, denoted as I_{kth}. If yes, then the picking time z_o of the order is equal to 1, and 0 otherwise. Every hour, the available capacity of the PS is assessed, and products are replenished from the back area according to the remaining forecast for that day, where the time for internal replenishment r is assumed to be one hour. For the hourly replenishment, the remaining forecast is again sorted according to the (remaining) forecast volume. If a product has had no orders in the last three days, no further units are added to the PS.

At the end of each day, the remaining forecast is replaced with the forecast for the next day and adjusted by the product units currently in the PS, as well as those that are currently being internally replenished. A list of indices and parameters used is given in Table 3.1, and the simulation algorithm is further outlined in Appendix B.4.

3.4 CASE STUDY CONTEXT

The data is obtained from a large European fashion retailer, henceforth referred to as 'the company', that is specifically interested in understanding which clickstream variables could be useful in demand forecasting. The company has a tremendously large product assortment, making product-level forecasting especially challenging. For confidentiality reasons, no detailed information about their current forecast could be obtained. While the company also uses a back and forward area in their warehouses, no detailed information could be given concerning the forward area. We choose to select a pocket sorter for the forward area due to the novelty of the system.

Table 3.1: Notation table

Indices	
k	Product in dataset, $k = 1,...,K$
o	Product order, $o = 1,...,O$
t	Day in planning horizon, $t = 1, ...,T$ ($T = 14$)
h	Hour of the day, $h = 0, ..., H$ ($H = 23$)
Parameter	
F_{kt}	Forecast for product k on day t
R_{kt}	Remaining forecast for product k on day t
I_{kth}	Inventory in PS for product k on day t at hour h
o_{kt}	Order for product k on day t
z_o	Picking time for order o, $z_o = 1$ if product is picked from PS, 0 otherwise
r	Time for internal replenishment, $r = 1$
c	Capacity of PS, $c = \{5\%, 10\%, 15\%\}$
m	Maximum number of units per product allowed in PS, $m = 3$

3.4.1 *Data collection and preprocessing*

The data received spans a period of one year and covers sales and clickstream data for a set of randomly selected stock-keeping unit (SKU), referred to as 'products'. The SKU numbers provided by the company encapsulate information on the brand, model, and color of a product, but not its size. Our forecast could later be disaggregated using size curves and historical data on size selection. The clickstream data covers only aspects relating to product sites, and therefore does not include information on any other pages visited by customers (e.g., product category, brand site, shopping cart, etc.). The clickstream data can be categorized into two types:

1. View data: This data relates to when and for how long a customer visited a product site. A visit to a product site ends when a customer closes or moves to a different site in the browser or app. The variable view duration contains a particularly large number of outliers because customers may leave product sites open without actively viewing the product. We use a boxplot to analyze the view duration, and use the extreme of the upper whisker as the maximum view duration, which is equal to 18.4 seconds.

2. Event data: Events refer to clicks made on a product site, such as clicks on product images or adding a product to a customer's wishlist or shopping cart.

In addition, the data contain information regarding the customers who interact with the products, including their total number of purchases, gender, and date of birth. All products with no sales in the training period are defined as new product introductions and are excluded from the analysis, as new products

typically require a different forecasting approach. Moreover, products with no clickstream data in the training set, for instance relating to customers who were not signed in while browsing, are also excluded as only their historical sales are available for forecasting. This results in a final dataset containing 3,000 different products. All data is scaled to a mean of zero and a standard deviation of one for the application of machine learning models, PCA, and K-means clustering.

3.4.2 *Initial data analysis*

Overall, the sales data shows strong weekly seasonality, with the highest order volumes on Sundays and then decreasing until Saturday. As the company's product lifecycle is rather short, it is unsurprising that most products do not show demand over the whole period of the data: 22% of products had sales on fewer than ten days during the whole year, emphasizing the issue of intermittent demand behavior in the dataset. In fact, in the 3,000-product dataset used, very few products actually drive most of the demand.

As we are investigating the effect of clickstream data on demand forecasting, the time difference between visits/clicks and purchases is of particular interest. For purchased products, the average difference between the first time a customer views a product and the subsequent purchase is 0.64 days. For 'add to wishlist' clicks, an average of 3.18 days elapse until purchase, whereas this is only 0.13 days for 'add to cart' clicks, where customers add products to their shopping cart. These short time differences indicate that using clickstream data for demand forecasts two or seven days ahead may be challenging. However, it should be noted that these numbers do not distinguish between first and repeat purchases (i.e., a customer purchasing the same product again), where the purchase decision process is different.

3.4.3 *Feature selection*

To determine how much data from historical sales to use in the forecast using machine learning models, we look at autocorrelation in the sales time series of various products. While there are no demand lags that show significant autocorrelation across all products, the most recent demand lags generally seem to show the highest autocorrelation. Therefore, we use the last 14 days of the training data as predictors for the machine learning models. With respect to the clickstream data, we explore frequently-used clickstream variables that have been applied in previous studies (Huang and Van Mieghem 2014, Moe and Fader 2004, Van den Poel and Buckinx 2005), such as the number of product site visits. In addition to variables that can easily be observed from the data, we calculate several additional variables:

- We split the variable for product site visits into three separate variables, namely the number of visits in the morning (5 am - 9 am), during the day (9 am - 5 pm), and during the evening (5 pm - 5 am).

- Using the average view time for a product during the training period, we calculate product site visits lasting longer than the average view time and those with a view duration equal to or less than the average view time.
- The average total number of orders of customers visiting a product site on a given day is added to the variable set, as well as the average conversion rate of these customers, referring to the average number of visits resulting in a purchase.
- Lastly, we use the conversion rate for various variables to calculate expected orders for a product. This is done by dividing the number of clicks or visits by the respective conversion rate. As this essentially means dividing the whole clicks or visits time series for a product by its conversion rate, we only use the resulting new time series for the machine learning methods. For DR, forecast results will hardly change if we manipulate the time series with the same value.

In addition, we add weekday as a dummy variable for DR, and two different types of product categories as dummy variables for all machine learning methods. As DR predicts future demand for one product at a time, product category cannot be added as a regressor. The full variable set is listed in Table 3.2, and their pairwise correlations are shown in Table B.3 in the Appendix. As can be expected from a dataset containing clickstream data, most of the variables have a relatively high correlation, limiting the possibility of combining variables in linear regression, forecasting, and clustering.

3.5 RESULTS

In the next subsections, we outline the results of our analysis.

3.5.1 *Linear regression*

Owing to high correlation between the clickstream variables, we only add variables with a correlation below 0.7 to the clickstream model. The results show that for both the two-day- and the seven-day-ahead forecasts, the model containing clickstream data explains more of the variation in sales than the model containing only historical sales, weekday, and product category (Table 3.3).The adjusted R-squared improves from 0.34 to 0.40 for the two-day- and from 0.22 to 0.30 for the seven-day-ahead forecast. These results indicate that clickstream variables do indeed help to explain variation in sales, supporting our hypothesis that they improve forecast accuracy. While most variables are statistically significant at the 1% level, historical sales and the number of 'add to cart' clicks seem to be the best variables to explain sales variation.

3.5.2 *Forecast baseline*

Of the forecasting methods applied, DR and SVR produce the best results, in terms of both RMSE and MASE (Table 3.4), and are therefore used in subse-

Table 3.2: List of variables used

Number	Variable
1	Historical sales
2	Visits (to the product site)
3	Unique visits (i.e., unique customers)
4	Total view time (on product site)
5	Average view time per visit
6	Variance in view time across visits
7	Number of 'add to cart' clicks
8	Number of 'add to wishlist' clicks
9	Total number of clicks
10	Average number of clicks per customer
11	Variance in clicks across customers
12	Visits above (i.e., lasting longer than) average view time
13	Visits below (i.e., shorter than or equal to) average view time
14	Visits in the morning (5 am - 9 am)
15	Visits during the day (9 am - 5 pm)
16	Visits in the evening (5 pm - 5 am)
17	Average total orders of customers (visiting the product site)
18	Average conversion rate of customers (visiting the product site)
19	Orders based on converting visits
20	Orders based on converting unique visits
21	Orders based on converting view time
22	Orders based on converting 'add to cart' clicks
23	Orders based on converting 'add to wishlist' clicks
24	Orders based on converting clicks
25	Weekday
26	Product category (two different variables)

quent analyses. The neural network does not converge for any tested combination of hyperparameters. The fact that the MASE values are all above one indicates that the forecast for the test data performs worse than the naïve forecast for the training data. However, this does not mean that a naïve forecast should be used for the test data rather than the models outlined in 3.4. The training period is much longer than the test period, and is characterized by a large number of consecutive zero-demand days because many products are introduced during the year. For example, if the first half of the training period for a product shows zero demand, then a naïve forecast, where the demand two days ago is used to forecast today's demand, naturally has a very high forecast accuracy. In comparison, with a test dataset spanning only 14 days in which many products do show demand, the forecasting task becomes more difficult. To verify this, we assess the effect of using a naïve forecast on the test data. As expected, the forecast performs worse than most other methods (Table 3.4). The MASE values should therefore not be interpreted in terms of whether they are below one, but should rather be used to compare the forecast models.

Table 3.3: Linear regression results (standardized coefficients)

	2-day-ahead forecast		7-day-ahead forecast	
Variables[1]	Baseline model	Clickstream model	Baseline model	Clickstream model
Historical sales (2 days ago)	0.39***	0.20***	-	-
	(0.0003)	(0.0004)		
Historical sales (7 days ago)	0.28***	0.18***	0.46***	0.21***
	(0.0003)	(0.0004)	(0.0003)	(0.0004)
Number of 'add to cart' clicks	-	0.35***	-	0.36***
		(0.0004)		(0.0004)
Average view time per visit	-	0.01***	-	0.01***
		(0.0005)		(0.0005)
Variance in view time across visits	-	0.02***	-	0.03***
		(0.0004)		(0.0005)
Average number of clicks per customer	-	0.00	-	0.01***
		(0.0004)		(0.0004)
Variance in clicks across customers	-	0.00***	-	0.00***
		(0.0003)		(0.0003)
Average total orders of customers	-	0.00	-	0.00*
		(0.0003)		(0.0003)
Average conversion rate of customers	-	0.00***	-	0.00***
		(0.0003)		(0.0003)
Weekday (Monday)	0.02***	0.03***	0.01***	0.01***
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Weekday (Tuesday)	0.00***	0.01***	0.01***	0.01***
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Weekday (Wednesday)	0.01***	0.01***	0.01***	0.00*
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Weekday (Thursday)	0.00**	0.01***	0.00***	0.00
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Weekday (Saturday)	0.00**	0.00**	0.00***	-0.01***
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Weekday (Sunday)	0.04***	0.05***	0.02***	0.02***
	(0.0001)	(0.0001)	(0.0011)	(0.0011)
Product category 1 (Men's clothing)	-0.01***	0.01***	-0.01***	0.01***
	(0.0007)	(0.0006)	(0.0007)	(0.0007)
Product category 1 (Unisex clothing)	0.02***	0.01***	0.03***	0.02***
	(0.0017)	(0.0016)	(0.0018)	(0.0017)
Product category 1 (Children's clothing)	-0.01***	0.01***	-0.01***	0.01***
	(0.0023)	(0.0016)	(0.0025)	(0.0018)
Product category 1 (Other)	-0.01***	0.01***	-0.01***	0.01***
	(0.0016)	(0.0013)	(0.0017)	(0.0014)
Product category 2 (Footwear)	0.05***	0.02***	0.08***	0.04***
	(0.0012)	(0.0012)	(0.0013)	(0.0013)
Product category 2 (Kids)	0.03***	0.01***	0.05***	0.02***
	(0.0026)	(0.0018)	(0.0029)	(0.0019)
Product category 2 (Sports)	0.02***	0.01***	0.04***	0.02***
	(0.0013)	(0.0013)	(0.0014)	(0.0014)
Product category 2 (Textile)	0.05***	0.04***	0.07***	0.05***
	(0.001)	(0.001)	(0.0011)	(0.0011)
Product category 2 (Beach-/Underwear)	0.05***	0.04***	0.08**	0.06***
	(0.0015)	(0.0015)	(0.0017)	(0.0016)
Product category 2 (Other)	-0.01***	0.00**	-0.02***	0.00
	(0.0011)	(0.0011)	(0.0012)	(0.0011)
Intercept	-0.04***	-0.04***	-0.05***	-0.04***
	(0.0012)	(0.0011)	(0.0013)	(0.0012)
Adjusted R-squared	0.34	0.40	0.22	0.30
Akaike information criterion (AIC)	21,024,096	20,137,818	22,454,316	21,544,349

Standard errors are reported in parentheses. $^{*}p \leqslant 0.1$, $^{**}p \leqslant 0.05$, $^{***}p \leqslant 0.01$.
1 For the 2-day ahead forecast, we use sales two and seven days ago. For the 7-day ahead forecast, only sales seven days ago are used. For the clickstream variables, we use their values two days ago for the 2-day ahead forecast and seven days ago for the 7-day ahead forecast.

Table 3.4: Baseline results (only historical sales)

		2-day-ahead forecast		7-day-ahead forecast	
Method	Best-performing hyperparameters	MASE	RMSE	MASE	RMSE
ARIMA	Automatically selected	1.56	1.40	1.66	1.53
RF	Ntree: 500 Mtry: p/3	1.77	1.49	1.99	1.57
SVR	Kernel: Linear Cost: 0.01	1.51	1.39	1.70	1.44
NN	-	Does not converge			
JRNN	Neurons: 1 & 6[1] Learning rate: 0.001 & 0.0001[1] Max. iterations: 1000	1.56	1.66	2.52	1.65
ERNN	Neurons: (6, 6) Learning rate: 0.0001 Max. iterations: 1000	2.17	1.58	2.85	1.65
Naïve	-	1.81	1.62	1.98	1.76

1 Using 1 neuron and a 0.001 learning rate produces the best results for the 2-day-ahead forecast and 6 neurons with a 0.0001 learning rate the best results for the 7-day-ahead forecast.

3.5.3 *Forecast including clickstream data*

Tables 3.5 and 3.6 show the DR and SVR results when the clickstream variables are added to the forecast (in addition to historical sales). For DR, for both forecasts, statistically significant improvements in forecast accuracy are obtained only when using visits, unique visits, and 'add to wishlist' clicks as additional regressors. The statistical significance is computed using a one-sided paired t-test, with the null-hypothesis that the MASE and RMSE values of the forecast with clickstream variables are lower than those for the forecast containing only historical sales. Interestingly, the results also indicate that visits in the evening are better predictors of future demand than visits at other times of the day. When using SVR, we obtain much more significant results using the clickstream variables. Specifically, the number of 'add to cart' clicks and orders based on converting unique visits show the best forecast accuracy. For the latter, the MASE improves from 1.51 to 1.47 and the RMSE from 1.39 to 1.37 for the two-day-ahead forecast. The dummy variables do not improve the accuracy of either forecast method.

While SVR in combination with the clickstream variables delivers better forecast results than DR, a closer look at the results reveals that SVR is less accurate for the few high-demand products in the dataset (Figure 3.2).
To test whether combinations of clickstream variables further improve forecast accuracy, we combine unique visits and 'add to wishlist' clicks with average total orders, average conversion of customers, and the dummy variables for DR,

Table 3.5: Results of DR (clickstream variables)

	2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historical sales	MASE	RMSE	MASE	RMSE
None	1.56	1.40	1.66	1.53
Visits	1.53	1.39*	1.64	1.49*
Unique visits	1.54	1.38*	1.64	1.48**
Total view time	1.55	1.39	1.67	1.50
Average view time per visit	1.60	1.41	1.72	1.52
Variance in view time across visits	1.58	1.41	1.68	1.52
Visits above average view time	1.56	1.40	1.67	1.51
Visits below average view time	1.54	1.39	1.64	1.49*
Visits in the morning	1.56	1.47	1.65	1.56
Visits during the day	1.56	1.45	1.68	1.54
Visits in the evening	1.52**	1.43	1.64	1.53
Number of 'add to cart' clicks	1.58	1.44	1.66	1.54
Number of 'add to wishlist' clicks	1.52**	1.40	1.61***	1.39*
Total number of clicks	1.55	1.41	1.67	1.52
Average clicks per customer	1.61	1.41	1.73	1.53
Variance in clicks across customers	1.55	1.40	1.66	1.51
Average total orders of customers	1.56	1.40	1.67	1.51*
Average conversion rate of customers	1.58	1.40	1.69	1.50*
Weekday	1.58	1.46	1.68	1.57

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$.

and do the same for 'add to cart' clicks and orders based on converting unique visits for SVR. We also combine the average and variance in view time with the total view time, as total view time alone may not be a good indicator of purchase intention due to the issue of visitors leaving product sites open. For DR, all variable combinations decrease forecast accuracy to some extent (Table 3.7). For SVR, while some variable combinations, such as the number of 'add to cart' clicks combined with the average conversion rate of customers, show statistically significant improvements in forecast accuracy (Table 3.8), better results are achieved when using a single clickstream variable (Table 3.6).

3.5.4 *Clustering*

Using the first feature set for clustering (i.e., clustering based on demand characteristics), the application of K-means clustering results in three product clusters, as suggested by the silhouette method. Several products are excluded from the clustering algorithm because they have zero demand during the time period so the time series features cannot be calculated. These products are added separately to a fourth cluster.

Table 3.6: Results of SVR (clickstream variables)

	2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historic sales	MASE	RMSE	MASE	RMSE
None	1.51	1.39	1.70	1.44
Visits	1.47^{***}	1.38^{***}	1.68	1.43^{**}
Unique visits	1.48^{***}	1.38^{***}	1.68	1.42^{***}
Total view time	1.47^{***}	1.38^{***}	1.66^{***}	1.43^{**}
Average view time per visit	3.01	1.59	3.92	1.68
Variance in view time across visits	2.99	1.59	3.26	1.64
Visits above average view time	1.48^{**}	1.38^{**}	1.66^{***}	1.43^{**}
Visits below average view time	1.47^{***}	1.38^{***}	1.68^{*}	1.43^{**}
Visits in the morning	1.51	1.39^{*}	1.68	1.44
Visits during the day	1.48^{***}	1.38^{**}	1.67^{***}	1.43^{**}
Visits in the evening	1.49^{**}	1.39^{**}	1.69	1.43^{**}
Number of 'add to cart' clicks	1.48^{**}	1.37^{***}	1.67^{**}	1.42^{***}
Number of 'add to wishlist' clicks	1.49^{*}	1.39	1.67^{**}	1.43^{**}
Total number of clicks	1.48^{*}	1.38^{***}	1.65^{***}	1.43^{***}
Average clicks per customer	2.28	1.53	3.99	1.69
Variance in clicks across customers	1.54	1.40	2.51	1.59
Average total orders of customers	2.20	1.52	3.18	1.64
Average conversion rate of customers	2.21	1.53	3.20	1.64
Orders based on converting visits	1.48^{***}	1.37^{***}	1.65^{***}	1.44
Orders based on converting unique visits	1.47^{***}	1.37^{***}	1.64^{***}	1.43
Orders based on converting view time	1.48^{***}	1.38^{***}	1.67^{***}	1.44
Orders based on converting 'add to cart' clicks	1.48^{***}	1.37^{***}	1.67^{*}	1.42^{***}
Orders based on converting 'add to wishlist' clicks	1.46^{***}	1.39	1.66^{***}	1.44
Orders based on converting clicks	1.47^{***}	1.38^{***}	1.64^{***}	1.43
Product categories	1.86	1.44	2.46	1.54

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$.

For clustering based on the clickstream data and total demand, we use each product's conversion rate of unique visits and total demand during the last eight weeks of the training data. The unique visits variable is selected because it is a clickstream variable with one of the best forecast results in the previous analysis. The conversion rates generally show extremely high correlation ($>$ 0.9); therefore we refrain from using all of them and performing PCA, and instead only select the conversion rate of unique visits. To arrive at meaningful results, we replace all outlier values with a maximum unique visits conver-

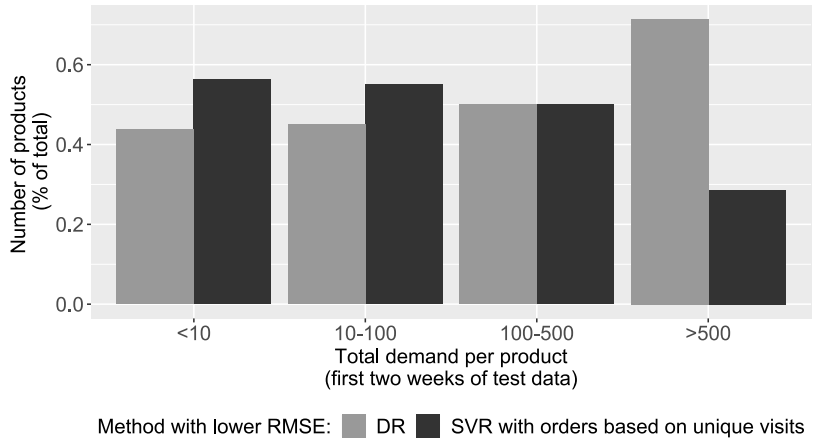

Figure 3.2: Comparison of DR and SVR results (2-day ahead forecast)

sion rate of 52 and maximum total demand of 77 units. Furthermore, for this feature set, the silhouette method suggests three clusters, which can easily be analyzed because the feature set is small. The first cluster consists mainly of products with low (i.e., intermittent) to medium demand but high conversion rates, meaning that these products have a high number of unique visits without subsequent purchases. The second cluster contains products with medium to high demand and low conversion rates. Cluster three contains low- to medium-demand products with low conversion rates. For confidentiality reasons, product names are anonymized, so we cannot investigate what distinguishes the low-demand products with either low or high conversion rates from each other. The latter may potentially be products that were sold out so that unique visits could not actually result in purchases.

To forecast demand, we again use the clickstream variables with the best results in the previous SVR forecast (i.e., number of 'add to cart' clicks and orders based on converting unique visits). The results show that the clusters based on demand characteristics worsen both RMSE and MASE (Table 3.9). Although including the clickstream variables improves forecast accuracy, the results are worse than running SVR on the full product dataset. For the clusters based on the clickstream data and total demand (Table 3.9), there is a large improvement in the MASE, even without including the clickstream data in the forecast. Adding the clickstream variables further improves accuracy. To extend our analysis, we examine the results by cluster (Table 3.10). For comparison reasons, we also include the forecast accuracy for each cluster when SVR is run across the entire 3,000-product dataset.

Looking at the low- to medium-demand clusters (1 and 3), we see that clustering itself produces large improvements in the MASE, while adding the clickstream variables has a small, but partially significant effect (e.g., adding 'add to cart' clicks for cluster 3). However, for the two-day-ahead forecast, adding the clickstream variables for cluster 1 does not significantly improve forecast accuracy. For the second cluster, clustering itself worsens MASE, but the clickstream data does substantially improve accuracy. Generally, the results suggest that including the clickstream data is most useful for forecasting medium- to

Table 3.7: Results of DR (combined variables)

	2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historic sales	MASE	RMSE	MASE	RMSE
None	1.56	1.40	1.66	1.53
Unique visits + Average total orders of customers	1.59	1.40	1.66	1.51
Unique visits + Average conversion of customers	1.58	1.42	1.70	1.53
Unique visits + Weekday	1.54	1.42	1.66	1.53
'Add to wishlist' clicks + Average total orders of customers	1.56	1.42	1.65	1.52
'Add to wishlist' clicks + Average conversion rate of customers	1.57	1.44	1.67	1.54
'Add to wishlist' clicks + Weekday	1.58	1.46	1.67	1.57
Total view time + Average view time per visit	1.61	1.42	1.70	1.53
Total view time + Variance view time across visits	1.62	1.43	1.70	1.53

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$.

high-demand products, and low- to medium-demand products with low conversion rates.

Compared with our previous analysis, a forecast per cluster with the clickstream variables delivers the best MASE values (1.38 and 1.46 for the two- and seven-day-ahead forecasts, respectively), while forecasting with the clickstream variables but without clustering results in a lower RMSE value (1.37) for the two-day-ahead forecast. The RMSE for the seven-day-ahead forecast is the same with or without clustering. This may indicate that clustering works especially well for improving the forecast accuracy of intermittent-demand products, while running machine learning methods across all products is more suitable for medium- to high-demand products.

3.5.5 *How forecasts including clickstream data improve order picking time*

To further investigate products for which including clickstream data in the forecast is especially useful, we perform an ABC analysis before simulating order picking. Using the last two weeks of the training data, products making up 80% of the demand volume are denoted as A products, and the remaining ones as BC products. In the order picking simulation, we use two different datasets resulting from the ABC analysis: one using all products (K = 3,000) and the other containing only B and C products (K = 2,752).

The capacity constraints of 5%, 10%, and 15% of the PS translate into 175, 350, and 525 available bags when using all products, and 50, 100, and 150

Table 3.8: Results of SVR (combined variables)

	2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historic sales	MASE	RMSE	MASE	RMSE
None	1.51	1.39	1.70	1.44
Number 'add to cart' clicks + product categories	1.55	1.38*	2.07	1.49
Number 'add to cart' clicks + Average total orders of customers	1.53	1.38*	3.14	1.62
Number 'add to cart' clicks + Average conversion rate of customers	1.50	1.37***	1.69	1.42**
Orders based on converting unique visits + product categories	1.63	1.40	2.12	1.51
Orders based on converting unique visits + average total orders of customers	2.16	1.50	3.10	1.63
Orders based on converting unique visits + average conversion rate of customers	1.48***	1.38***	2.41	1.57
View time + Average view time per visit	2.29	1.53	3.84	1.67
View time + Variance in view time across visits	2.97	1.58	3.21	1.63

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$.

bags when using only B and C products, respectively. As DR shows superior forecast results for high-demand products, we use both DR and SVR forecasts, with and without clickstream variables, in the simulation. In addition, we use the forecast resulting from the second feature set for clustering as an input into our simulation. The results are shown in Table 3.11. For the dataset containing all products, the highest number of orders is picked from the PS when using DR, independent of the capacity level. This is in line with our forecast results suggesting that DR outperforms SVR for high-demand products. As the PS is filled according to the highest forecast volume, A products are generally prioritized. While the clickstream data shows a small positive effect on forecast accuracy when using DR, this is only reflected in the 5% capacity simulation. For the other two capacity levels, the number of orders picked from the PS decreases when adding the clickstream data.

When using BC products, SVR in combination with the clickstream data results in the largest proportion of products being picked from the PS. At a 5% capacity level, the share of orders increases from 3.40% to 4.38% when the number of 'add to cart' clicks is added to the SVR forecast using only historical sales. Compared with the DR forecast without clickstream data (3.21%), the SVR forecast with 'add to cart' clicks shows a relative increase of 36% in

Table 3.9: Overall cluster results (SVR)

		2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historical sales	Clustering	MASE	RMSE	MASE	RMSE
None	None	1.51	1.39	1.70	1.44
None	Demand characteristics	1.84	1.48	2.00	1.53
Number of 'add to cart' clicks	Demand characteristics	1.63	1.41	1.80	1.47
Orders based on converting unique visits	Demand characteristics	1.65	1.42	1.78	1.48
None	Clickstream data[1]	1.39^{***}	1.40	1.48^{***}	1.43
Number of 'add to cart' clicks	Clickstream data[1]	1.39^{***}	1.38	1.46^{***}	1.42^{***}
Orders based on converting unique visits	Clickstream data[1]	1.38^{***}	1.38^{***}	1.46^{***}	1.42^{*}

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$.
1 Clusters 1 and 2 are forecasted using a linear kernel (cost = 0.01) and cluster 3 a radial kernel (cost = 0.1, gamma = 0.01).

products picked from the PS. At 10% and 15% capacity, the increases are 28% and 25%, respectively. Using forecasts from clustering produces very low picking performance, supporting our hypothesis that clustering helps to improve the forecast for intermittent-demand products in our dataset, while using SVR with clickstream data on the full dataset is better able to forecast medium- to high-demand products. It should be noted that 72% of B products are allocated to the medium- to high-demand cluster when using K-means, meaning that this cluster does not contain only A products.

Combining our findings from the forecasting and the order picking simulation, we conclude that using clickstream data for demand forecasting is especially useful for B products. For the few products driving most of the demand, DR without clickstream data produces better results. We might very well imagine that for these products, different warehousing systems will be applied, for instance in the form of a separate forward area, where the inventory of these products is frequently replenished. Our analysis shows that using a PS system for BC products in combination with a forecast model that includes clickstream data might substantially improve picking times.

3.6 DISCUSSION AND CONCLUSION

Despite extensive research on demand forecasting, the inclusion of clickstream data in product-level demand forecasting has received limited attention. This paper investigates the value of using clickstream data in short-term sales forecasting for the product assortment of a leading European fashion retailer. In a

case study of warehouse operations, we assess how our forecast models impact the order picking process.

Table 3.10: Results per cluster (SVR)

			2-day-ahead forecast		7-day-ahead forecast	
Variables in addition to historical sales	Cluster[1]	Forecast per cluster?	MASE	RMSE	MASE	RMSE
None	1	No[2]	1.45	0.44	1.84	0.50
None	1	Yes	1.31	0.44	1.59	0.50
Number of 'add to cart' clicks	1	Yes	1.30	0.44	1.60	0.49**
Orders based on converting unique visits	1	Yes	1.31	0.44	1.55***	0.49**
None	2	No[2]	2.39	4.94	2.49	5.05
None	2	Yes	2.45	4.94	2.57	5.05
Number of 'add to cart' clicks	2	Yes	2.40***	4.87**	2.49***	4.98***
Orders based on converting unique visits	2	Yes	2.41***	4.86***	2.52***	5.02
None	3	No[2]	1.20	0.41	1.35	0.43
None	3	Yes	1.03	0.42	1.03	0.41
Number of 'add to cart' clicks	3	Yes	1.02*	0.42***	1.03*	0.41***
Orders based on converting unique visits	3	Yes	1.03	0.42*	1.03	0.41

* $p \leqslant 0.1$, ** $p \leqslant 0.05$, *** $p \leqslant 0.01$; significance tested in comparison to results when running SVR per cluster, with no variables in addition to historic demand.
1 Clusters 1 and 2 are forecasted using a linear kernel (cost = 0.01) and cluster 3 a radial kernel (cost = 0.1, gamma = 0.01).
2 The forecast results are shown for products in the cluster but from running SVR on the whole product dataset.

Comparing a traditional time-series forecasting method (DR) and various machine learning methods, our forecast results show that, particularly when using machine learning methods, clickstream data can help improve forecast accuracy. Using several different clickstream variables in the prediction, the number of 'add to cart' clicks and the number of converted unique visits, estimated using the conversion rate of unique visits for a product, show the largest improvements in forecast accuracy when using a support vector machine model. These results are significant at the 1% level. This is similar to findings from Weingarten and Spinler (2020a) who show that 'add to cart' clicks are a strong predictor of short-term sales. However, for very high-demand products, traditional time series methods using only historical sales outperform machine learning methods with clickstream variables.

Clustering products based on demand volume and clickstream data further reveals that using clickstream data in forecasting is most useful for medium- to high-demand products, as well as for low- to medium-demand products

Table 3.11: Simulation results (order picking)

Data-set	Forecast method	Variables in addition to historic sales	Forecast per cluster?	Orders picked from PS (%) c = 5%	c = 10%	c = 15%
ABC	DR	None	No	14.62	21.42	25.88
ABC	DR	Unique visits	No	14.72	21.03	25.08
ABC	DR	Number of 'add to wishlist' clicks	No	14.12	21.13	24.92
ABC	SVR	None	No	12.98	18.58	22.09
ABC	SVR	Number of 'add to cart' clicks	No	13.59	19.33	23.58
ABC	SVR	Orders based on converting unique visits	No	13.60	19.56	23.11
ABC	SVR	Number of 'add to cart' clicks	Yes[1]	13.36	19.20	23.14
ABC	SVR	Orders based on converting unique visits	Yes[1]	13.25	19.12	22.94
BC	DR	None	No	3.21	5.81	7.85
BC	DR	Unique visits	No	4.19	6.37	8.27
BC	DR	Number of 'add to wishlist' clicks	No	3.73	5.93	8.18
BC	SVR	None	No	3.40	5.12	7.43
BC	SVR	Number of 'add to cart' clicks	No	4.38	7.45	9.81
BC	SVR	Orders based on converting unique visits	No	4.30	6.69	8.78
BC	SVR	Number of 'add to cart' clicks	Yes[1]	2.80	5.56	7.97
BC	SVR	Orders based on converting unique visits	Yes[1]	2.98	5.42	7.90

1 Using total demand and clickstream data as input to clustering. Cluster 1 and 2 are forecasted using a linear kernel (cost = 0.01) and cluster 3 a radial kernel (cost = 0.1, gamma = 0.01).

with low conversion rates (i.e., a small number of clicks and visits resulting in purchases). The latter suggests that clickstream data might be useful for new product demand forecasting, where limited sales data is available for forecasting immediately after the product introduction.

The application to order picking is largely in line with our forecast results. Using forecasts to steer internal replenishments between a back area in the warehouse with slow picking times and a forward area with fast picking times, resembling a PS system, DR results in the largest proportion of product orders being picked from the forward area. After categorizing products into A, B, and C according to their demand volume and only using B and C products for internal replenishments to the forward area, SVR in combination with clickstream variables produces better results than DR. In fact, adding clickstream variables to an SVR model enables 25-36% more product orders to be picked from the forward area compared with the DR model using only historical sales. This indicates that SVR in combination with clickstream variables is especially suited to forecasting demand for B products.

Our results show the usefulness of including clickstream data in short-term sales forecasts. In addition to order picking, our approach might be applied to several other warehouse operations, including the relocation of inventory

across warehouses to shorten delivery times. Other areas, such as marketing and customer relationship management, might also benefit from improved short-term demand forecasts. We believe that our results are generally applicable to other online fashion retailers. Whether this is the case for other industries cannot be confirmed from our study, as the relationship between clickstream data and purchases may be very different for non-fashion items. It would also be interesting to assess the effect of clickstream variables in forecasting for fashion retailers with both offline and online sales channels.

Naturally, forecasting using clickstream data is limited by data quality. In particular, issues with tracking click types create challenges to using clickstream data in forecasting. Our results are also limited by the fact that only the last three weeks of the dataset serve as test data. To account for differences across seasons and specific events (e.g., holidays), future research should investigate the effect of clickstream data over a longer time period, as well as whether clickstream data is similarly beneficial in long-term sales forecasts.

We use DR and five different machine learning methods for our analysis. Replicating our approach using other methods, such as more complex recurrent neural networks, might result in further improvements to forecast accuracy. Moreover, additional variables, such as price or promotion campaigns, which could not be obtained for our dataset, should be investigated in this context. In particular, price may affect the relationship between clickstream data and future sales.

It is difficult to predict true future demand because, as in this study, forecasts typically use sales to measure accuracy. In the case of temporary stockouts, clickstream data might add even more value in combination with historical sales in demand forecasting. Such forecasts might be used, for instance, to prioritize returns handling so that forecasted products are made available again more quickly. However, this can only be assessed if retailers are able to estimate true demand, and not just observed demand in the form of sales.

Online business-to-consumer (B2C) retailers have tremendous opportunities to collect a wealth of data that can be leveraged in the forecasting process. Whether similar opportunities exist in the business-to-business (B2B) context will be the focus of the next chapter. Using a dataset from a large supplier in the construction industry, we investigate whether data mining and machine learning methods can be used to automatize mechanisms in the forecasting process that would typically require human judgement, for instance in the form of expert input. While the dataset might not qualify as big data, the methods applied go beyond the use traditional software tools.

4

USING SEQUENTIAL PATTERN MINING TO IMPROVE DEMAND FORECAST ACCURACY

The following chapter is based on Weingarten and Spinler (2020c).

4.1 INTRODUCTION

Forecasting is increasingly important to organizations, owing to rising competitive pressure, shorter product lifecycles, and changing consumer needs such as faster and more reliable delivery times (Sanders and Manrodt 2003). Accurate demand forecasts are relevant across all processes of the supply chain (e.g., production, inventory management, materials requirement planning) and also affect companies' suppliers and retailers. Inventory management, in particular, is a crucial part of the supply chain as it drives inventory costs and service levels. Inaccurate forecasts may result in excessive inventory or stock-outs, and ultimately lost sales if customer demand cannot be fulfilled. Research suggests that even small improvements in forecast accuracy can have a substantial effect on inventory and service levels (Petropoulos et al. 2018). According to Kremer et al. (2016), every percentage improvement in forecast accuracy results in a similar percentage improvement in terms of reduced safety stock, without affecting customer service levels.

Given its relevance, there is a large body of research on demand forecasting. Many contributions focus on developing forecasting techniques to better predict future demand, with varying degrees of complexity and ease of use (Zotteri and Kalchschmidt 2007). Much of this research addresses the selection of forecasting methods (e.g., Makridakis et al. 2018, Petropoulos et al. 2018), from naïve and averaging methods, to exponential smoothing (ETS) and autoregressive integrated moving average (ARIMA) models. In addition to these traditional forecasting methods, more complex models using machine learning machine learning (ML) algorithms, such as neural networks, are also applied to demand forecasting (e.g., Carbonneau et al. 2008, Gutierrez et al. 2008). However, these are typically limited in their interpretability, providing few insights into how forecasts are developed and the effect of input variables (Petropoulos et al. 2018).

Despite the volume of research on demand forecasting, recent studies suggest that practical applications of forecasting techniques still lag behind academic developments (Sanders and Manrodt 2003). A common approach that companies apply to forecasting demand involves developing an initial statistical demand forecast using simple forecasting methods, such as exponential smoothing, usually based on historical sales data. This initial forecast is then reviewed and adjusted for some key products, taking into account additional information available to the company, such as promotional and advertising activities, price changes, holidays or insufficient inventories (Fildes and Goodwin

2007, Fildes et al. 2009). The consensus in the literature is that these judgemental adjustments may have both negative and positive effects on forecast accuracy (Fildes et al. 2009, Perera et al. 2019). Judgemental adjustments often introduce bias into forecasts (Fahimnia et al. 2019), potentially owing to the human tendency to see patterns in noise (O'Connor et al. 1993); yet they are often the only means to incorporate additional information into the forecast, as many of the previously mentioned forecasting methods struggle to incorporate such 'soft data' (Goodwin and Fildes 1999, Petropoulos et al. 2018). Nevertheless, the complexity of the task often makes it impossible to review and adjust the forecast for all products in a company's portfolio. This raises the question of how human judgement can optimally be combined with forecasting methods.

One rarely mentioned reason for judgemental adjustments in forecasting is additional information about the product portfolio. Typically, demand for each product is forecasted separately, assuming that each is statistically independent. However, in some situations demand for products is correlated, and a forecast may thus benefit from taking account of demand for these related products. Reasons for demand correlation between products include new product introductions (i.e., a new product replaces a mature product), substitution or complimentary products. In such cases, higher demand for one product may increase or decrease demand for related products (Bandara et al. 2019). Another reason for demand correlation is hierarchy in a product portfolio, meaning that certain products can only be used in combination with others. For example, in the construction industry, certain drill bits can only be used with specific hammer drills. In such cases, an increase in demand for one product may result in increased demand for the correlated product. A few studies have investigated how to incorporate correlated demand into forecasting (e.g., Bandara et al. 2020a, 2019, Garnier and Belletoile 2019), but none appear to have examined how to identify products with correlated demand automatically, without using human judgement, and consequently to use this information in demand forecasting to estimate the impact on sales and inventory.

Our research is motivated by the demand forecasting of a leading construction industry supplier, ConstructX (anonymized to maintain confidentiality). Its product portfolio is characterized by thousands of products with very heterogeneous demand patterns, ranging from constant weekly demand to very intermittent demand. The company's existing forecast is initially constructed using simple statistical methods and later adjusted based on judgement. ConstructX is particularly interested in exploring how ML might improve this initial forecast. Previous research on demand forecasting in the construction industry has focused mainly on developments in the housing market and demand for construction overall, with limited attention to levers that might improve ConstructX's forecast. Our study therefore addresses three aspects. First, we investigate how ML might improve the accuracy of the company's initial forecast, including the application of traditional time series forecasting methods, such as ARIMA, to provide benchmarks on the performance of ML methods. The context of our study is especially interesting as we use a dataset containing thousands of products with very heterogenous demand patterns. Moreover, very limited research exists regarding product demand forecasting in the construction industry. In our application, machine learning improves the mean

absolute scaled error (MASE) of ConstructX's existing forecast by ~11-13%. Results from initial data analyses of the company's historical sales are used as additional input to refine the forecast, improving the MASE by an additional ~4-8%. Second, the main contribution of our research is to explore the extent to which sequential pattern mining, a rule-based ML method, might improve demand forecasting. We apply sequential pattern mining to automatically identify products with correlated demand, and leverage this information as an input into demand forecasting. This approach helps to avoid bias, which may occur when company experts hypothesize demand correlation between products. The results of our analyses indicate that sequential pattern mining may be beneficial when used in combination with traditional forecasting methods. In combination with a dynamic regression model, sequential pattern mining improves the MASE by up to 6.8%. Lastly, the impact of changes to the forecast accuracy is quantified in relation to inventory and lost sales. While several studies have used ML methods to forecast product demand (e.g., Bandara et al. 2019, Carbonneau et al. 2008, Gutierrez et al. 2008), most have merely compared their performance in terms of forecast accuracy. Our results show that our forecasting model would reduce the company's costs of inventory holding and lost sales by up to 6.9%.

The remainder of the paper is structured as follows: Section 4.2 reviews literature relating to this study, Section 4.3 introduces the dataset used in our research, and Section 4.4 explains the research approach and methodology. Section 4.5 presents the results and Section 4.6 assesses their effect on inventory planning and control. We conclude in Section 4.7 with a short summary and suggestions for future research.

4.2 LITERATURE REVIEW

Our study relates to three streams of research: applications of machine learning to demand forecasting, forecasting of correlated time series, and sequential pattern mining.

4.2.1 *Machine learning in demand forecasting*

Practical applications of demand forecasting are often challenged by datasets exhibiting erratic, intermittent and highly fluctuating demand, which often violates the assumptions of traditional forecasting methods, such as stationarity (Salinas et al. 2020). A stationary time series is one with properties (e.g., mean and variance) that do not change over time (Hyndman and Athanasopoulos 2018). Traditional forecasting methods are essentially linear methods and often cannot capture non-linear data patterns, such as non-linear trends or seasonal fluctuations (Gutierrez et al. 2008) which are often present in sales data (Chu and Zhang 2003). ML models may help overcome these limitations, which could explain their recent popularity as an alternative to traditional forecasting methods. Artificial neural networks (NN) are some of the most popular ML methods for forecasting and have been shown to provide satisfactory forecast results compared with traditional methods (Gutierrez et al. 2008, Hill et al.

1996), although discussion continues over when and under what conditions NNs outperform traditional methods (De Gooijer and Hyndman 2006). Even though NNs provide the benefit of enabling simulation of complex underlying non-linear relationships in the data, their computation is also quite time-consuming and requires large amounts of training data (Borovykh et al. 2018, Chu and Zhang 2003, Gutierrez et al. 2008). Recurrent neural networks (RNN) and long short-term memory networks (LSTM), a variant of RNN, have become well-known in the context of time series forecasting. Their recurrent connections make them especially suited to model sequenced data such as time series (Bandara et al. 2020b). Research on demand forecasting also extends to other ML models, such as support vector regression (SVR) (e.g., Carbonneau et al. 2008) and decision trees (e.g., Thomassey and Fiordaliso 2006), which are also able to model non-linear data patterns (James et al. 2013). The performance of various ML methods has been explored in several forecasting competitions (e.g., NN3, NN5). Ahmed et al. (2010) present a detailed review and comparison of ML methods in time series forecasting. While applying ML models to product demand forecasting and comparing their performance to traditional methods are not novel topics in the literature, we apply several of these methods to a new setting using the dataset of a construction industry supplier. The nature of the dataset, such as the presence of both products with high and intermittent demand and products with correlated demand, makes the application of ML models especially interesting.

4.2.2 *Demand forecasting of correlated time series*

The aforementioned traditional approaches to demand forecasting (e.g., ETS and ARIMA) not only have difficulty modelling non-linearities, but are also restricted to modelling univariate time series. This means that they are only capable of forecasting one time series at a time, without accounting for any potential correlations or shared features across time series. Although much research has focused on forecasting single or small groups of time series, general practical applications often require the forecasting of thousands of (potentially related) time series, such as large retailers' product portfolios (Salinas et al. 2020). To tackle this problem, recent studies have investigated the development of global models that can be run across multiple time series simultaneously. These often make use of ML methods such as RNN and LSTM (e.g., Bandara et al. 2020a,b, 2019, Salinas et al. 2020). Identifying shared information across time series is often difficult owing to the heterogeneous nature of the data (Salinas et al. 2020). Therefore, much of this research has focused on identifying subgroups of time series that share similar features, and then building a global model for each subgroup. Techniques applied range from manual grouping of products, for instance based on an overarching product category, on the assumption that they share similar demand features (Bandara et al. 2019, Borovykh et al. 2018), to clustering techniques that group time series based on a selected set of features (Bandara et al. 2020b, Thomassey and Fiordaliso 2006). However, few studies have investigated how information on historical demand for products can be used as an input into demand forecasts for related products. For example,

Trapero et al. (2015) leverage the effect of promotions on demand for a set of products as an input into forecasting demand for products with limited historical information on promotional effects. In addition to exploring the possibility of building global ML models across the time series in our dataset, we also investigate how to use the historical demand for products to forecast the demand for related products. We focus on identifying which products affect the future demand for other products using sequential associate rule mining, which is a novel approach that does appear to have been previously researched.

4.2.3 *Sequential pattern mining*

Technological developments have enabled companies to collect large amounts of sales data on products bought by specific customers and the dates of those transactions. One data mining method often used to derive knowledge from such datasets is associate rule mining, also referred to as market basket analysis, which was first introduced by Agrawal et al. (1993). Associate rule mining helps to determine 'which items are often bought together in the same transaction', and is typically applied in marketing, for instance for product shelf placements and promotions (Zaki 2001). Agrawal and Srikant (1995) later introduced the problem of sequentially mining transactions, essentially extending associate rule mining by including a temporal aspect. This approach, known as sequential pattern mining, can be used to model the evolution of customers' purchases, for instance to determine whether buying peanut butter in the past is associated with a higher likelihood of purchasing bread in the future (Agrawal and Srikant 1995). This helps to identify product bundles that are likely to be purchased sequentially. Agrawal and Srikant's (1995) proposed algorithm generates rules representing purchase sequences. An example of such rules in the construction industry might be '70% of customers who buy a hammer drill also buy drill bits a month later'. With this information, it might be hypothesized that using the historical demand and forecast for hammer drills as inputs into forecasting future demand for drill bits might improve forecast accuracy. Discovering such rules in a large dataset may be challenging. Therefore, Zaki (2001) developed an algorithm, referred to as sequential pattern discovery using equivalence classes (SPADE), for finding sequential patterns in large datasets. We apply this algorithm to identify products with potentially related demand (for details, see Section 4.4.3). The numerous applications of sequential pattern mining include product recommendations (e.g., Liu et al. 2009), customer behavior analysis (e.g., Chen et al. 2009), and product manufacturing (e.g., Deriyenko et al. 2017). However, our study appears to be the first to apply sequential pattern mining to the area of product demand forecasting.

4.3 DEMAND DATA

Since actual demand data were unavailable, historical sales data were collected from ConstructX to approximate demand. The dataset received contains sales for ten Central European countries over a time span of almost three years (January 2017 to November 2019), amounting to 150 weeks of data and consisting of

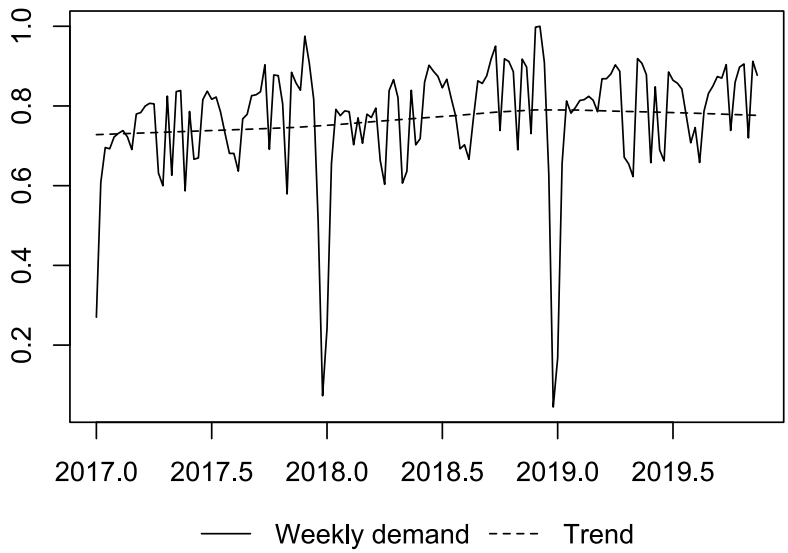

Figure 4.1: Weekly orders (Jan. 2017 - Nov. 2019)

approximately 22,000 different products. New product introductions, defined as products with no demand prior to 2019, were excluded, as new products typically require a different forecasting approach. The company's customer base is purely business-to-business (B2B), comprising companies from several different industries (e.g., dry wall and electrical installation).

For the purpose of inventory planning, the company's data are aggregated to a weekly level to enable weekly forecasting of demand. Figure 4.1 shows the development of and overall trend in weekly orders over the time period. For confidentiality reasons, the data are anonymized by indexing the values so that the largest weekly demand equals 1. The trend line shows that the number of orders had not grown much over those three years. However, orders seem to exhibit a yearly seasonality pattern, with increasing orders throughout the year, and a large peak close to the end of the year caused by an annual promotional campaign. As expected, owing to public holidays around Christmas and New Year's Eve, orders dropped at the end of the year. The autocorrelation function (ACF), which measures the linear relationship between current and past values of the time series (Hyndman and Athanasopoulos 2018), indicates yearly seasonality, with significant peaks at lags 51-54. The ACF for the order volume (in pieces) per week for several individual products also shows significant lags in those weeks, confirming yearly seasonality. No other consistently significant lags can be found across products. Moreover, the weekly order volume at a product level shows large fluctuations, making the company's inventory planning especially difficult. As high-frequency time series, such as weekly or daily data, often exhibit complicated seasonal patterns (Hyndman and Athanasopoulos 2018), we use multiple seasonal decomposition to decompose our time series into seasonal, trend and remainder components to further investigate seasonal patterns. However, we find no additional seasonality in the data other than yearly seasonality.

4.4 MODEL SPECIFICATION

In this section, we describe the process we use to develop the forecast model for ConstructX. First, we specify a baseline model containing only variables based on historical sales data. In the baseline model, we test both traditional and ML forecasting methods to determine which are most suited to our data. The best-performing methods are applied in subsequent steps of our analysis. Second, we adjust the baseline model based on hypotheses derived from interviews with ConstructX members and initial analyses of the sales data. Third, we apply the SPADE algorithm for sequential pattern mining to identify products with correlated demand, and include these products' historical sales and forecasts into our forecast. The baseline model serves as a benchmark to assess whether the hypotheses derived in step two and sequential pattern mining in step three improve forecast accuracy. As the company is producing one-month and three-month forecasts which it wishes to adapt to weekly forecasts, we translate this multi-step forecast into 5-week and 13-week forecasts. Owing to the long training times required by some of the forecasting methods used (e.g., neural networks), all models are first trained and tested on a random subset of 3,000 products. All models are implemented in R (version 3.6.2).

4.4.1 *Baseline model*

ConstructX has been using a combination of simple moving average (SMA), simple exponential smoothing (SES), and Croston's method (Croston) to forecast demand. For confidentiality reasons the exact forecast methodology was not shared, and as the company's forecasts are at a monthly level, overall forecast accuracy cannot be compared to our results. Therefore, we apply all three methods to approximate the company's current forecast accuracy. SMA, a classic method in time series forecasting, uses the average demand of several sequential demand values to forecast future values. SMA treats observations equally, whereas SES makes predictions based on the weighted sum of past observations, where a smoothing factor controls the exponentially decreasing weight assigned to past observations (Hyndman and Athanasopoulos 2018). The third method used by the company, Croston's method, is known to work especially well for intermittent demand, as it separately smoothes demand size and the inter-arrival time of demand, which refers to the average time between non-zero demand periods (Croston 1972). In addition to these three forecasting methods, we also use ARIMA, dynamic harmonic regression (DHR), and TBATS. In addition to exponential smoothing models, ARIMA is another widely used approach to time series forecasting that aims to capture autocorrelation in a time series by using time series lagged values and lagged forecast errors. An underlying assumption when using ARIMA is that the time series is stationary. This can be achieved through differencing, where the differences between consecutive observations are computed (Box et al. 2016). In order to include additional explanatory variables, we use a regression model with ARIMA errors as outlined by Hyndman (2010), referred to as dynamic regression (DR). When the seasonal period is large, as is the case for our data (seasonal period

of approximately 52 weeks), DHR is often a more suitable forecasting method. DHR uses Fourier terms to handle seasonality, although, unlike ARIMA, it assumes that the seasonal pattern does not change over time (Hyndman and Athanasopoulos 2018). Lastly, we use TBATS, which is an extension of exponential smoothing methods and adds trigonometric terms (De Livera et al. 2011). Like DHR, TBATS is able to handle large seasonal periods, but allows seasonality to change slowly over time (Hyndman and Athanasopoulos 2018).

As we are running the forecasting methods on thousands of products, we choose the parameters that perform best for all products. For SES and TBATS, the best parameters are automatically estimated through the application of the R forecast package, while for ARIMA, the auto.arima function in the R package is selected as it automatically returns the best ARIMA model according to criteria such as the akaike information criterion (AIC), which compares the quality of forecast models (Hyndman and Athanasopoulos 2018). To enable reproducibility, a summary of all parameters tested for each forecasting method is given in Table C.1 in the Appendix.

With regard to machine learning, we specifically focus on methods that are well-known in the context of regression, as demand is a continuous variable. Specifically, random forest (RF), feed-forward neural networks (NN), recurrent neural networks (RNN), and SVR are applied. RF is a supervised learning method, meaning that it learns relationships between input variables and a response variable, and constructs an ensemble of decision trees. At each split in a decision tree, a random subset of variables is considered and the data are split so as to improve the homogeneity of the daughter nodes compared with the parent node (Breiman 2001). RF is a popular learning method as it is simple to train while yielding high accuracy (Hastie et al. 2009). In the class of NN, the multilayer perceptron is applied, which is the most commonly used form of artificial neural networks. A multilayer perceptron consists of multiple neurons (nodes) arranged in several layers. It learns relationships between the variables and the response variable through backpropagation. RNN are specific types of neural networks that are able to model sequenced data as they contain feedback connections that preserve sequential information. In our study, we apply two well-known RNN: the Elman recurrent neural network (ERNN) and the Jordan recurrent neural network (JRNN) (Elman 1990). Lastly, SVR is another supervised learning method that uses a non-linear function to map data points into a high-dimensional space, enabling a linear model to be computed (James et al. 2013). SVR then finds a line (or hyperplane) to fit the data that best minimizes the error rate (Vapnik 1999). These five ML methods are selected because they are well-known in the context of regression, but differ in terms of accuracy and training times depending on the input data (Hastie et al. 2009). For all ML methods, the variables are standardized to give sample means of 0 and a standard deviation of 1.

Common approaches to multi-step ahead forecasts rely on either a recursive or direct prediction strategy. Using a recursive approach, an H-step ahead forecast is achieved by iterating a one-step ahead forecast H times. In each iteration, the one-step ahead forecast is used as an input into the following forecast, which has the disadvantage that errors are accumulated. Using a direct strategy, an H-step ahead forecast is directly computed (Sorjamaa et al.

2007). As the first results of RF predictions show greater forecast accuracy using a direct strategy than using a recursive technique, the former approach is adopted.

4.4.1.1 *Evaluating forecast accuracy*

To assess the models' predictive performance, time series cross-validation is applied (Figure 4.2), in which we use 14 different test sets, each consisting of two different weeks to be predicted, as we are calculating a multi-step ahead forecast (five weeks and 13 weeks ahead). The training data consists only of observations prior to those in the test data. To give the models sufficient training data from which to learn, our test set consists of weeks toward the end of the time period covered by our dataset.

Measures of forecast accuracy have frequently been discussed in the literature, often with recommendations on which measures are best suited to particular types of data. Some of the most well-known forecast accuracy measures are the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root-mean-square error (RMSE). Some measures, such as MAE and RMSE, depend on the scale of the data and should not be applied to data with different scales (Kolkova 2020). As our dataset consists of both products with high weekly demand and products with very intermittent demand, MAE and RMSE are not used to measure forecast accuracy. Although MAPE is not scale-dependent, it has the disadvantage of being infinite or undefined if the demand is zero (Hyndman and Athanasopoulos 2018). Therefore, as our dataset contains intermittent demand products, we also refrain from using MAPE. Hyndman and Koehler (2006) propose a different measure, referred to as the mean absolute scaled error (MASE), to compare forecast accuracy across data with different units. MASE uses the MAE of the training data from a naïve forecast method to scale the MAE of the values forecasted from the test data. It is not only independent of the scale of the data, but is also easy to interpret. Values larger than one indicate that the forecast is worse than the average naïve forecast of the training data, whereas values smaller than one indicate better performance (Hyndman and Koehler 2006). As our data shows yearly seasonality, we apply a seasonal naïve forecast method to the training data. For a seasonal time series, the MASE can be calculated as:

$$\text{MASE} = \frac{\frac{1}{J}\sum_j |e_j|}{\frac{1}{T-m}\sum_{t=m+1}^{T} |y_t - y_{t-m}|} \tag{4.1}$$

where e_j is the forecast error for a given week and J is equal to our 14-week rolling forecast horizon, as we seek to compute the forecast accuracy by averaging over the different test datasets. The denominator is the MAE of the seasonal naïve forecast of the training data, where m is equal to 52 weeks. A seasonal naïve forecast means that actual demand in the previous season is used as the forecast. As we use an expanding window for our training data, where for each subsequent week of the forecast horizon, the training data also expand by one week (see Figure 4.2), we apply the seasonal naïve forecast method to

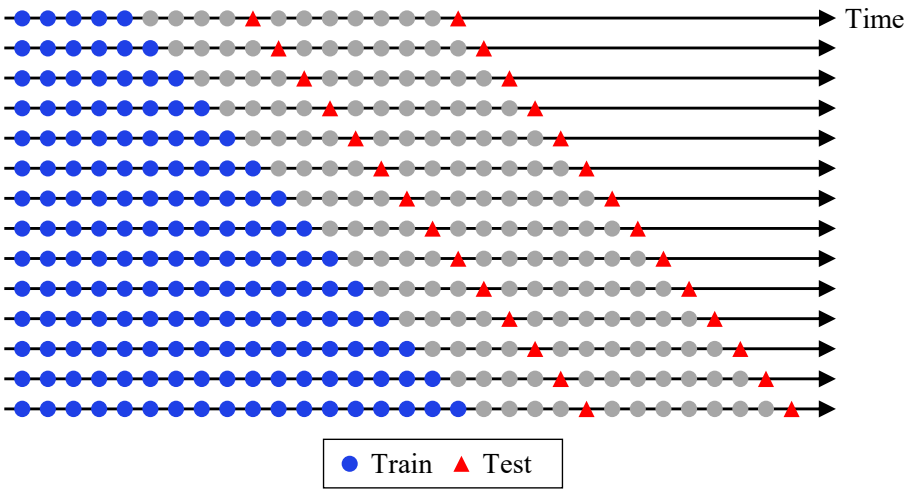

Figure 4.2: Time series cross-validation (14-week rolling horizon)

the longest training dataset. In order to compare the MASE across forecasting methods, we compute the average MASE across all products for each method.

4.4.1.2 *Variable selection for machine learning*

ML methods were run both across all products (AP) simultaneously, to make use of their ability to learn across time series, and per product (PP). This is hereafter referred to as the 'ML set-up'. When training an ML method on a single product, the product's time series must be split into sequences so that the algorithm has several batches of observations on which to be trained. The length of each sequence and the number of sequences per product depend on the variables selected to train the model. In time series forecasting, the inputs for ML methods typically consist of a selection of lagged observations, but determining this selection can be difficult (Zhang et al. 1998). To assess which demand lags are most suitable to predict demand five and 13 weeks ahead, the RF importance measure is calculated, whereby an RF is constructed to assess variables' importance. The importance measure reflects the mean decrease in prediction accuracy if a variable is excluded. When calculating the RF importance measure for different weeks, no set of demand lags consistently shows a high mean decrease in prediction accuracy, making it difficult to select variables based on significance. Therefore, we test four different sets of variables: the last six demand lags (Set 1), the last six demand lags and the 52^{nd} lag (Set 2), all last demand lags up to the 52^{nd} lag (Set 3), and all demand lags, which was only used when running ML methods across all products (Set 4). Figures C.1 and C.2 in the Appendix show an example of using the last six demand lags for a five-week ahead forecast when applying ML methods per product and across products.

4.4.1.3 *Time series clustering*

As outlined in Section 4.2.2, achieving good forecast results when applying ML methods across several time series often requires the identification of sub-

groups of time series sharing similar features, so that an ML model can be trained for each subgroup. In order to identify these subgroups, hierarchical clustering is applied. This technique constructs a hierarchy of clusters based on a set of features, typically by considering each observation as an individual cluster and then merging similar clusters at each iteration based on their proximity (bottom-up approach). Several approaches, also referred to as linkage methods, compute the proximity of two clusters (Hastie et al. 2009). We use Ward's (1963) method, which merges clusters at each iteration by minimizing the increase in the total within-cluster sum of the squared error, as this is the only linkage method resulting in a reasonable number of products per cluster for our dataset. All other linkage methods result in one large cluster containing almost all products alongside several very small clusters.

Several features can be computed to describe the characteristics of a time series. The features used for hierarchical clustering in this study are given in Table C.2 in the Appendix. To determine the optimal number of clusters, we use the elbow method, which estimates the sum of squared errors (SSE) for several numbers of clusters. By plotting the number of clusters against the SSE, we establish the optimal number of clusters when the SSE no longer decreases much with an additional cluster, referred to as the 'elbow' of the graph (Ketchen and Shook 1996).

4.4.1.4 *Generalizability*

To determine whether our findings from the baseline model are also applicable to other data, we simulate an additional dataset. For this, we use the simulator for intermittent demand from the tsintermittent package in R, which simulates time series based on three metrics: the average intermittent demand interval, the squared coefficient of variation of non-zero demand periods, and the mean demand of non-zero demand periods. This simulation assumes that arrivals of non-zero demand weeks follow a Bernoulli distribution and that non-zero demand weeks follow a negative binomial distribution. We calculate the average of the three metrics for all clusters identified with hierarchical clustering, and use this to simulate an additional dataset with 3,000 products. All forecasting methods are then applied to the simulated data, using the parameters that performed best in the baseline model.

4.4.2 *Formulation of hypotheses to adjust the baseline model*

Following interviews with members of the logistics department at ConstructX and initial analyses of the data, hypotheses are developed to refine the baseline forecast. With regard to external factors influencing demand, the company strongly believes that public holidays and working days per week affect demand owing to the B2B nature of the business. For the same reason, industry-specific variables relating to the growth of the construction industry and gross domestic product (GDP) development might similarly affect demand. The construction industry is also heavily influenced by weather conditions. However, we are constructing 5- and 13-week ahead forecasts, and weather forecasts are relatively unreliable so far ahead (Steinker et al. 2017). Instead, we use the

month of the week being forecasted as a proxy for weather, for instance with winter months representing cold and potentially snowy weather conditions. This leads to our first hypothesis regarding external variables:

> *H1: Using external variables (public holidays as a binary variable, number of working days per week, construction industry growth, GDP, and month as a proxy for weather) as inputs for demand forecasting improves forecast accuracy.*

We obtained data on public holidays from Public Holidays Global[1]. Growth in the construction industry was obtained from the index of production in construction (in Germany) produced by the Organisation for Economic Co-operation and Development (OECD)[2]. The same source was used for GDP data.

ConstructX sells its products through several different sales channels. When investigating the order volume per week at a product level, it is noticeable that two sales channels, namely retail (physical stores) and direct sales (by phone), show different time series patterns from other sales channels (for an example of one specific product, see Figure 4.3). Values are again indexed so that the largest weekly demand across sales channels equals 1. This leads to the formulation of our second hypothesis:

> *H2: Separately forecasting the 'retail' and 'direct sales' sales channels provides better forecast accuracy than a single forecast across all sales channels.*

Estimating the effects of input variables is generally difficult with ML models owing to their limited interpretability. Therefore, we test both hypotheses, and specifically all external variables in hypothesis 1, separately to assess their effect on forecast accuracy.

4.4.3 *Sequential pattern mining as an input into forecasting*

The SPADE algorithm for sequential pattern mining uses a dataset in vertical ID-list format. In our application, each customer is given an identifier (ID), and the database associates purchased products with this customer ID, along with a time-stamp. A collection of products purchased by a customer is referred to as an 'itemset', which may consist of single or groups of items purchased together at a specific time or in sequence. Itemsets with temporal sequences may take the form $(BC \rightarrow AD)$, referred to as a temporal rule, meaning that a purchase of products A and D is preceded by a purchase of products B and C (Zaki 2001). Hereafter, we refer to the left-hand side of a rule as 'preceding products' and the right-hand side as 'predicted products'.

At a daily level, a customer's purchases of products a few days apart may be caused by product unavailability or deliveries from different warehouses, not by the customer purchasing a product in relation to previously purchased products. To truly determine which products show sequentially correlated demand, we aggregate purchases to a monthly level and measure sequential purchase

[1] https://publicholidays.de/
[2] http://www.oecd.org/

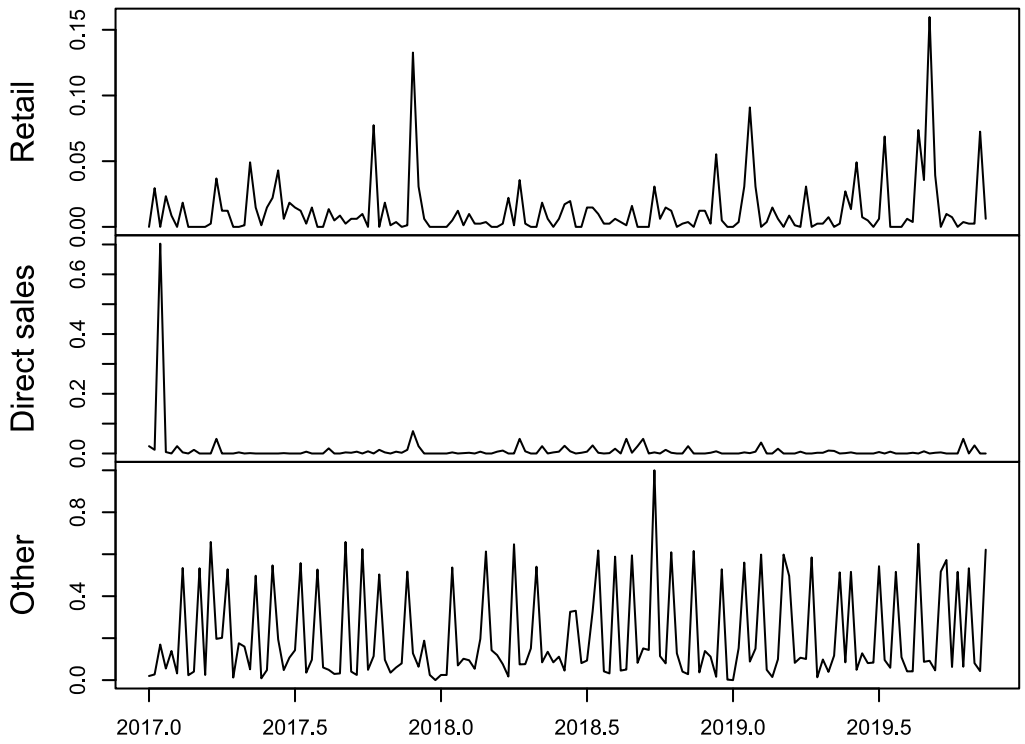

Figure 4.3: Order volume (in pieces) per sales channel for one product

patterns in two consecutive months across the last seven months of our training data.

The aim of the SPADE algorithm is to find frequent itemsets and temporal rules. The two main terms used to calculate frequency are 'support' and 'confidence'. Support refers to the number of customers who purchased a certain itemset. Support of 50% for itemset (AB) means that 50% of all customers purchased A and B in a month, while support of 50% for $(B \rightarrow A)$ means that half of the customers purchased B and then A a month later. Confidence refers to the likelihood of a temporal rule. Support of 50% and confidence of 10% for rule $(B \rightarrow A)$ means that while half of the customers purchased B and then A a month later, there is only a 10% likelihood that customers purchase product A in the month after they purchased product B.

Running SPADE for two consecutive months across seven months results in six different sets of temporal rules as outputs. As we want to use products' historical demand and forecasts as inputs into forecasts for a related product, we remove all rules where the right-hand side of a rule consists of more than one product. As the six sets of temporal rules contains different rules for the same product, we select the rule with the highest support. Before running the SPADE algorithm, thresholds for support and confidence must be chosen. Owing to the large number of products and often limited demand per product, we decide that support for an itemset should be at least 0.1%. Higher values result in almost no rules being found. For confidence, we test thresholds of 5%, 10% and 20%. The SPADE algorithm can be implemented in R using the arulesSequences package. For a detailed description of the algorithm, see Zaki (2001).

4.5 RESULTS

In this section, we present the results from the baseline model, and from the adjusted baseline based on the developed hypotheses in Section 4.4.2 and the inputs of sequential pattern mining.

4.5.1 *Baseline model*

Of the three methods used by the company, SES performs best in terms of forecast accuracy, while ARIMA outperforms all traditional forecasting methods, with MASE values of 0.64 and 0.65 for the 5- and 13-week ahead forecasts. Table 4.1 summarizes the model's performance for all applied forecasting methods using the ConstructX dataset, including the best-performing parameter for each model.

After fitting the ARIMA models, a plot of the ACF of the residuals for two example products (Product A with high demand, Product B with intermittent demand) shows some remaining autocorrelation (Figure 4.4). However, the Ljung-Box statistic (Ljung and Box 1978) shows a large p-value for both products, indicating white noise and hence good model fit. However, some of the residuals of other products do not exhibit a white noise process, suggesting that the selected ARIMA model does not provide a good fit. This is unfortunately a disadvantage of using auto.arima, as it may not always produce a model with better fit in comparison to manually selecting ARIMA model parameters, which is infeasible to do across thousands of products. In the ARIMA models chosen by auto.arima, only 2% of the 3,000 products use a seasonal ARIMA model, indicating that the 52-week seasonality seen in the data might be driven by just a handful of products. Moreover, for half of the products, an ARIMA (0,0,0) model was fitted, indicating that the time series of these products resembles white noise, suggesting that these are time series with no autocorrelation (Hyndman and Athanasopoulos 2018).

Of the ML methods, SVR provides the highest forecast accuracy for ConstructX, with a MASE of 0.57 for the 5-week ahead forecast and 0.58 for the 13-week ahead forecast. Out of all ML methods, the variable set containing the last six demand lags and the 52^{nd} lag delivers the best MASE results. Moreover, all ML methods apart from RF result in extremely large MASE values when run across all products. Closer examination of the data reveals that the ML methods are heavily influenced by products with high demand, making it difficult to predict the many zero-demand periods of other products correctly. NN delivers no results: when run across all products, the algorithm does not converge, and when run per product, it runs endlessly without providing any results. When running the ML models per product, several products have to be excluded. Owing to the large number of zero-demand periods for several products, the response variable in the training data sometimes consisted of only zero-demand weeks, resulting in an error when applying any ML method. RF requires the response variable to have more than five unique values, further excluding products from the forecast. Products that have to be excluded from the ML forecast are replaced by results from ARIMA in order to compare all

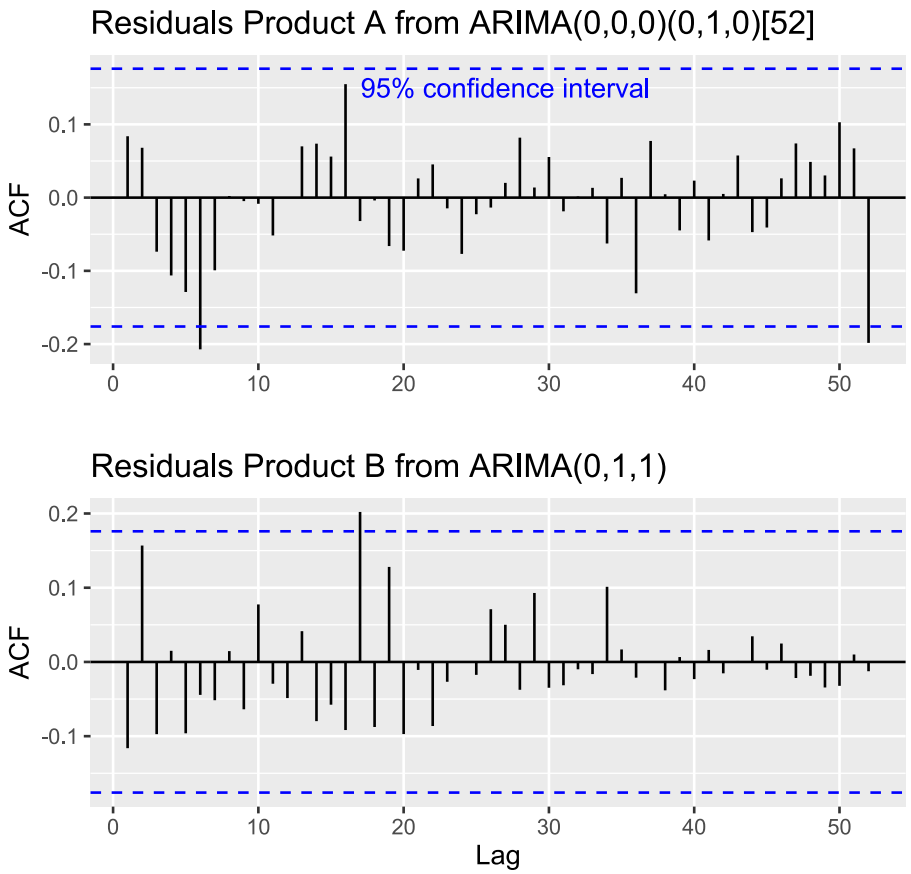

Figure 4.4: Baseline model: autocorrelation function of residuals (two example products)

3,000 products across methods. Overall, ARIMA, RF, SVR and JRNN produce better forecast accuracy than SES. To test whether the forecast accuracy of those four methods is significantly better compared to SES, a paired samples Wilcoxon Test is used, as the difference in MASE is not normally distributed. Results show that the improvement in MASE for ARIMA, SVR and JRNN are statistically significant at $p \leqslant 0.01$.

To evaluate the SVR model's ability to generalize, we apply SVR to both the test and training sets. As we do not replace any products with ARIMA results in this case, we receive a forecast for only around 2,000 products. The test data shows a MASE of 0.67 and 0.68 for the 5- and 13-week ahead forecasts, while the training data produces values of 0.59 and 0.59. As there is no large increase in MASE between the training and test data, overfitting does not appear to be a concern.

For the simulated data, SVR also shows the best performance in terms of MASE (Table 4.1). The simulated data does not take account of autocorrelation or seasonality in ConstructX's original dataset as the simulation is purely based on metrics related to the nature of intermittent demand of the company's products. Our results highlight the power of SVR in forecasting intermittent demand, irrespective of other characteristics of the time series.

Table 4.1: Results: baseline model

Method	Best-performing parameters	ML set-up	Variable Selection	MASE (ConstructX) 5-week ahead forecast	MASE (ConstructX) 13-week ahead forecast	MASE (Simulated data) 5-week ahead forecast	MASE (Simulated data) 13-week ahead forecast
SMA	Order: 7	-	-	0.87	0.91	1.15	1.21
SES[1]	-	-	-	0.68	0.68	0.99	1.05
Croston	Alpha: 0.1	-	-	1.28	1.30	1.04	1.09
ARIMA[1]	-	-	-	0.64	0.65	0.98	1.03
DHR	K: 1	-	-	0.70	0.73	0.99	1.05
TBATS[1]	-	-	-	0.84	0.90	1.05	1.12
RF	Ntree: 500	PP	Set 2	0.65	0.66	1.01	1.07
SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.57	0.58	0.90	0.95
NN	Neurons: c(2,1)	PP	Set 2	Does not converge		1.08	1.15
JRNN	Neurons: 1 Learning rate: 0.01 Max. iterations: 1000	PP	Set 2	0.63	0.64	0.99	1.04
ERNN	Neurons: c(6,4) Learning rate: 0.01 Max. iterations: 1000	PP	Set 2	0.79	0.81	1.17	1.21

1 Best parameters are automatically selected for each product.

4.5.2 *Baseline model with clustering*

Applying hierarchical clustering and using the elbow method to determine the optimal number of clusters results in a set of four different clusters. The first consists of products with relatively stable demand, meaning little demand variability and no large demand peaks. Cluster two contains products with several zero-demand periods and average demand variability. Cluster three contains solely intermittent-demand products, meaning a very large number of zero-demand periods, combined with very high demand variability. Lastly, the fourth cluster contains products with almost no zero-demand periods.

To improve forecast accuracy, we test all forecasting methods on each cluster. The results are presented in Table 4.2. For clusters 1, 2 and 4, SVR with the same parameters as in the baseline model performs best. For cluster 3, ARIMA provides better results. However, calculating the MASE across all clusters produces the same forecast accuracy as using SVR for all products, indicating no improvement in accuracy through clustering. Furthermore, after clustering, most ML methods still deliver extremely large MASE values when run across all products in a cluster. Therefore, we apply hierarchical clustering a second time, now using the average demand in non-zero demand periods in the last 52 weeks of the training data, as this seems to greatly affect the ML models' ability to deliver lower MASE values when run across products. Again testing all forecasting methods per cluster, ML methods run across all products in a cluster now deliver MASE values below 1. However, SVR run per product with variable set 2 is again the best-performing forecasting method for each cluster, resulting in no improvement in MASE compared with the baseline model without clustering.

Table 4.2: Results: baseline model (with clustering)

Cluster	Method	Parameters	ML set-up	Variable selection	MASE (5-week ahead forecast)	MASE (13-week ahead forecast)
1	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.67	0.69
2	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.62	0.61
3	ARIMA[1]	-	-	-	0.21	0.22
4	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.59	0.58
All		Best method per cluster			0.57	0.58

1 Best parameters are automatically selected for each product.

Table 4.3: Results: adjusted baseline model (external variables)

		5-week ahead forecast			13-week ahead forecast		
Method	External variables included	Baseline MASE[1]	MASE	Δ	Baseline MASE[1]	MASE	Δ
DR	Public holidays	0.72[2]	0.74	0.02	0.85	0.86	0.01
DR	Working days per week	0.72[2]	0.78	0.06	0.85	0.91	0.06
DR	Construction growth	0.72[2]	0.76	0.04	0.85	0.88	0.03
DR	GDP	0.72[2]	0.72	0.00	0.85	0.85	0.00
DR	Weather approximation	0.64[3]	0.82	0.18	0.65	0.84	0.19
SVR	Public holidays	0.63[2]	0.65	0.02	0.76	0.77	0.01
SVR	Working days per week	0.63[2]	0.68	0.05	0.76	0.81	0.05
SVR	Construction growth	0.63[2]	0.68	0.05	0.76	0.80	0.04
SVR	GDP	0.63[2]	0.65	0.02	0.76	0.78	0.02
SVR	Weather approximation	0.57[3]	0.62	0.05	0.58	0.64	0.06

1 Represents the MASE using ARIMA or SVR without including external variables.
2 Based on demand data from Germany only.
3 Based on all demand data.

4.5.3 *Adjusted baseline model*

To investigate the effect of the external variables outlined in Section 4.4.2, we use both ARIMA and SVR (with the best-performing parameters, variable set and ML set-up from the baseline model) as these were the traditional and ML forecast methods that resulted in the lowest MASE values in the baseline model. In order to include external variables, we replace ARIMA with a regression model with ARIMA errors, namely DR. Approximately 60% of the company's demand originates in Germany. To establish whether the external variables generally have a positive effect on forecast accuracy, public holidays, the number of working days per week, construction market growth and GDP for Germany are used to forecast demand from Germany for the 3,000 products. For construction market growth and GDP, the forecasts are computed using ARIMA, as actual values would not be known at the time of forecasting. Table 4.3 shows the forecast accuracy of DR and SVR without external variables, referred to as the baseline MASE, compared with the forecast accuracy with external variables included. None of the external variables improves forecast accuracy when using DR or SVR. Interestingly, the RF importance measure indicates a higher mean decrease in accuracy for most external variables compared with the lagged demand values. When using RF, the inclusion of external variables actually improves forecast accuracy. Nevertheless, the MASE using RF is still larger than for DR and SVR. As a result, no further analyses involving the external variables are conducted, and H1 is rejected.

When forecasting sales channels separately, again SVR performs best, but with slightly different parameters for each sales channel (Table 4.4). Overall, the results show a MASE of 0.57 for both multi-step ahead forecasts, meaning that the 13-week ahead forecast improves by 0.01. This supports H2, as the difference in MASE is statistically significant at the at $p \leqslant 0.01$.

Table 4.4: Results: adjusted baseline model (sales channel)

Sales channel	Method	Parameters	ML set-up	Variable selection	MASE (5-week ahead forecast)	MASE (13-week ahead forecast)
Retail	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.64	0.74
Direct sales	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.75	0.82
Other	SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	PP	Set 2	0.66	0.68
All		Best method per sales channel			0.57	0.57

4.5.4 *Sequential pattern mining*

Sequential pattern mining, as outlined in Section 4.4.3, results in the identification of only a few temporal rules, despite the low support value chosen (0.1%). This can be explained by the large number of intermittent-demand products in the company's product portfolio. To derive more rules to allow estimation of the effect on forecast accuracy, SPADE was run on the full 22,000-product dataset, which results in temporal rules for a set of 73, 118 and 157 products for three different confidence levels (5%, 10%, 15%). Although this is a small fraction of the whole dataset, the products for which temporal rules were identified are those with particularly high demand, where significant savings in inventory cost, backorders and lost sales might be achieved through greater forecast accuracy.

We forecast these products separately in order to assess whether the forecast accuracy might be improved by using the preceding items as inputs into the forecast. Once again, we use DR and SVR with the best-performing parameters, variable selection and the ML set-up from the baseline model. Prior to this, we calculate forecasts for the preceding products using SVR with a linear kernel, as this produces the best results in terms of MASE for these products. Table 4.5 shows the results for the three confidence levels. In order to estimate improvements in forecast accuracy, the table also shows the MASE results without including the preceding products in the forecast, referred to as the baseline MASE. Using DR, the MASE shows a statistically significant improvement for all three confidence levels when including the preceding products. The largest improvement in the MASE (6.8%) is achieved with a confidence level of 10%. For SVR, on the other hand, there is no improvement in forecast accuracy for any of the three confidence levels. Thus, traditional time series forecasting methods may be better able than ML methods to make use of the demand of preceding products as inputs. When the former provides better forecast results than the latter, sequential pattern mining may be beneficial. However, for ConstructX, we conclude that sequential pattern mining should not be included in the forecast.

4.5.5 *Final model*

In a last step, the findings from the (adjusted) baseline model are applied to the full 22,000-product portfolio to estimate the overall improvement in forecast accuracy compared with ConstructX's existing forecast, which is approximated using SES. Table 4.6 shows that the company might improve its forecast accuracy by 11-13% by switching to an SVR forecasting method. Forecasting demand per sales channel improves the MASE by an additional 4-8%. In the 3,000-product dataset, forecasting by sales channel only produced a marginal improvement in the MASE. However, its application to the full dataset verifies that forecasting by sales channel has a positive impact on forecast accuracy, confirming H2. All improvements in MASE are statistically significant at the 1% confidence level.

Table 4.5: Results: sequential pattern mining

Confidence level	Number of predicted products	Method	5-week ahead forecast			13-week ahead forecast		
			Baseline MASE[1]	MASE	Δ	Baseline MASE[1]	MASE	Δ
5%	157	DR	0.73	0.71	-0.02**	0.76	0.74	-0.02***
		SVR	0.68	0.71	0.03	0.69	0.73	0.04
10%	118	DR	0.74	0.69	-0.05***	0.76	0.72	-0.04***
		SVR	0.67	0.69	0.02	0.68	0.71	0.03
20%	73	DR	0.77	0.75	-0.02*	0.84	0.81	-0.03**
		SVR	0.69	0.71	0.02	0.74	0.78	0.04

1 Represents the MASE without the inclusion of preceding items.
*$p \leqslant 0.1$, **$p \leqslant 0.05$, ***$p \leqslant 0.01$; statistical significance of difference in MASE is only tested for improvements in MASE when using temporal rules.

Table 4.6: Results: final model (using 22,000-product dataset)

Forecasting step	Method	5-week ahead forecast			13-week ahead forecast		
		Baseline MASE[1]	MASE	Δ	Baseline MASE[1]	MASE	Δ
Baseline model (best performing method)	SVR	0.68	0.59	-0.09***	0.76	0.65	-0.08***
Adjusted baseline model (forecast by sales channel)	SVR	0.68	0.56	-0.12***	0.69	0.60	-0.13***

1 Represents the MASE when using SES.
*$p \leqslant 0.1$, **$p \leqslant 0.05$, ***$p \leqslant 0.01$; statistical significance of difference in MASE is tested.

The rationale for SVR performing substantially better than other methods for this dataset is determined by investigating the particular products for which SVR delivers better results. Comparing the average MASE to the number of zero-demand periods per product across the 14-week forecast horizon reveals that the difference in MASE between SVR and ARIMA becomes much larger when the number of zero-demand periods increases (Figure 4.5). It seems that the more zero-demand periods for a product, the better SVR performs compared to ARIMA, indicating that SVR may be especially suited to intermittent-demand products.

4.6 HOW FORECAST ACCURACY IMPACTS INVENTORY PLANNING

This section presents our assessment of the impact of our forecasting models compared with the company's existing forecast. This is done through a simulation of inventory across a 14-week planning horizon using the 5-week ahead forecast. Since high forecast accuracy reduces excess inventory and inventory shortages, the impact of the forecasting models are assessed in terms of the cost of holding excess inventory and lost sales arising from inventory shortages.

4.6.1 *Inventory planning model*

We assume that ConstructX follows an (r, S) inventory policy, meaning that for every r time period, the company places sufficient replenishment orders to restore the on-hand inventory to target level S (Silver et al. 1998). In our inventory planning model, r is equal to 1, meaning that replenishment orders are placed weekly. The order up-to level $S_{p,t}$ for each product p in week t is determined by the forecasted demand plus the safety stock that the company keeps to mitigate the risk of lost sales due to errors in the forecast. For simplicity reasons, we assume that the lead time LT, meaning the time it takes for order replenishments to arrive, is the same for each product. The lead time is set to four weeks to truly estimate the effect of the 5-week ahead forecasts. With a lead time of four weeks, the forecast can be used to place replenishment orders to ensure sufficient inventory on-hand for the forecasted week.

An inventory shortage $I^f_{t,p}$ occurs when there is insufficient inventory on hand to fulfill demand and potential backorders from previous weeks. This results in either lost sales or backorders requiring fulfillment in subsequent weeks. Owing to the nature of business in the construction industry, customers seldomly refrain from purchasing if products are unavailable. However, the company did not provide data on the percentage of lost sales c, meaning the fraction of purchases lost due to inventory shortage. Therefore, we test different lost sales percentages, ranging from 10% to 50%. The fraction of the inventory shortage not resulting in lost sales is equivalent to backorders. The cost of lost sales L across the 14-week planning horizon can therefore be calculated as:

$$L = \sum_{t=1}^{T} \sum_{p=1}^{P} I^f_{t,p} \times s_p \times m \times c \tag{4.2}$$

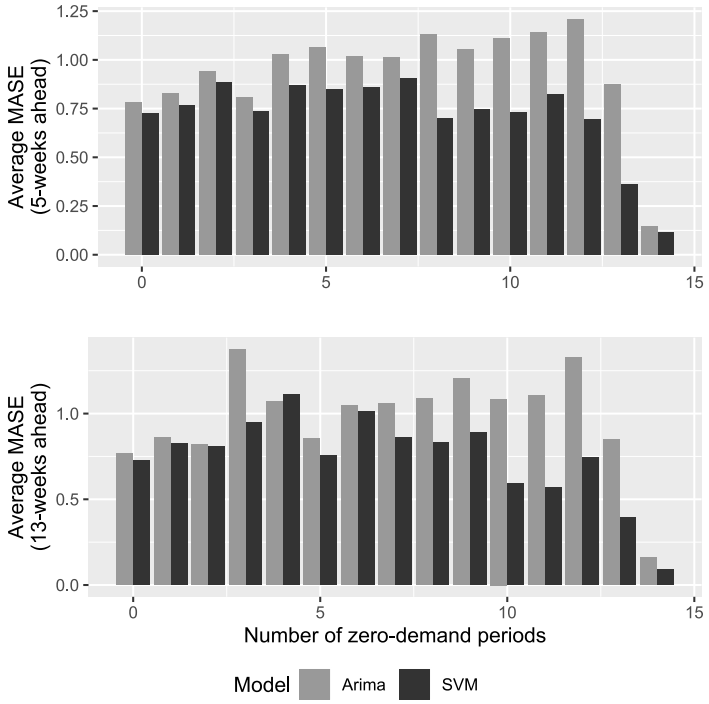

Figure 4.5: Comparison of SVR and ARIMA

For confidentiality reasons, ConstructX only provided an indexed sales price s_p, with the index known only to the company. To estimate the cost of lost sales, the indexed sales price is multiplied by an assumed profit margin m of 15%.

Inventory holding costs are associated with the costs of inventory storage, such as facility, labor and insurance costs, and depend on the average inventory on hand. Using the inventory at the beginning of week $I^b_{t,p}$ and at the end of the week $I^e_{t,p}$, the average inventory on hand over the 14-week planning horizon can be calculated, resulting in the following formula for total inventory holding cost C:

$$C = h \times \sum_{p=1}^{P} \frac{s_p \times (1-m)}{T} \sum_{t=1}^{T} \frac{I^b_{t,p} + I^e_{t,p}}{2} \tag{4.3}$$

The inventory holding cost h is assumed to be 15% of the cost of goods sold.

Table 4.7 presents an overview of the model notation and input parameter values. For a full explanation of the model, including all relevant equations, see Appendix C.4.

4.6.2 *Results of inventory planning model*

A first simulation of the inventory reveals that SVR tends to under-forecast demand, which may be very costly for a high lost sales percentage. See Appendix C.5 for the analysis relating to forecast bias. As different values of the SVR parameters result in similar forecast accuracy, we choose a combination that reduces the issue of under-forecasting while only marginally impacting on forecast accuracy (cost = 1, gamma = 0.1). With the updated parameters, the

Table 4.7: Notation table

Indices	
t	Time period (week), t,...,T (T = 14)
p	Product in product portfolio, p,...,P (P = 22,000)
Parameter and variables	
L	Total cost of lost sales
C	Total inventory holding cost
r	Review period, r = 1 week
$S_{p,t}$	Order up-to level for each product p in week t
LT	Lead time to replenish inventory, LT = 4 weeks
$I^b_{p,t}$	Inventory of product p at the beginning of week t
$I^f_{p,t}$	Inventory shortage of product p at the end of week t
$I^e_{p,t}$	Inventory of product p at the end of week t
s_p	Indexed sales price of product p
c	Percentage of lost sales, c = {10%, 20%, 30%, 40%; 50%}
m	Profit margin, m = 15%
h	Inventory holding cost, h = 15%

MASE changes to 0.61 and 0.66 for the 5-week and 13-week ahead forecasts using SVR for the 22,000-product dataset. For SVR forecasts by sales channel, the MASE changes to 0.58 and 0.62. All results are still statistically significant at the 1% confidence level. The simulation results with the updated parameters for SVR are presented in Table 4.8, using SES to approximate the company's current forecast. The costs of lost sales are extrapolated for one year in order to compare them with yearly savings on inventory holding costs. By simply switching to an SVR forecasting method, the company might decrease its yearly net cost, meaning the sum of the costs of inventory holding and lost sales, by up to 5.2%. Using an SVR forecast by sales channel, the net cost might be reduced by up to 6.9%. However, with a high lost sales percentage, the net cost savings reduce to zero, and are even negative for the forecast by sales channel. This is attributable to the fact that despite the change in parameter values, the SVR forecast still results in more products with an under-forecast than the SES forecast (see Appendix C.5). In an actual application of our forecast model, this forecast bias could be removed, for instance by adding the average forecast error to the forecast. However, to do this, the forecast error should first be observed over a longer period of time, which is outside the scope of our analysis as our planning horizon is set to 14 weeks.

Overall, an increased lost sales percentage also increases the inventory holding cost. This is because the former lowers the number of backorders, which must be fulfilled from inventory on hand in subsequent weeks, causing a larger average inventory on hand. However, the effect of different lost sales percentages on inventory holding costs is very minimal, as shown in Table 4.8.

Table 4.8: Effect of forecast models on costs of lost sales and inventory holding

	ConstructX's existing forecast (SES)			Baseline model (best-performing method: SVR)				Adjusted baseline model (forecast by sales channel: SVR)			
Lost sales percentage	Cost of lost sales	Inventory holding cost	Net cost	Cost of lost sales	Inventory holding cost	Net cost	Δ Net cost (%)	Cost of lost sales	Inventory holding cost	Net cost	Δ Net cost (%)
10%	1.7	9.9	11.6	2.0	9.0	11.0	-5.2	2.2	8.6	10.8	-6.9
20%	2.9	10.0	12.9	3.5	9.0	12.5	-3.1	3.8	8.6	12.4	-3.9
30%	3.8	10.0	13.8	4.6	9.0	13.6	-1.5	5.0	8.6	13.6	-1.5
40%	4.4	10.0	14.4	5.3	9.0	14.3	-0.7	5.8	8.6	14.4	0
50%	5.7	10.0	15.7	6.7	9.0	15.7	0	7.2	8.6	15.8	0.6

4.7 CONCLUSION

The increasing importance of accurate demand forecasts has given rise to a large body of research. The development of advanced forecasting methods and method selection are frequently discussed in the literature, while another stream of research focuses on the inclusion of human judgement in forecasts. Our study lies at the intersection of human judgement and method selection, as we investigate how to automatically detect information typically supplied by knowledge experts within a company. We also leverage this information to forecast demand using several traditional and machine learning methods. Using real-world data from a leading supplier in the construction industry, we study the impact of our forecasting models in terms of forecast accuracy and cost.

We find that of all the tested forecasting methods, SVR performs best in terms of forecast accuracy. It improves the company's existing forecast from a MASE of 0.68 for a 5-week ahead forecast and 0.73 for a 13-week ahead forecast to 0.59 and 0.65 respectively. Further analysis also suggests that SVR works especially well for intermittent-demand products. These findings are corroborated by applying the forecasting method to an additional simulated dataset based on the intermittent-demand characteristics of the original dataset.

To further improve forecast accuracy, external variables (e.g., public holidays and GDP) are included in the forecast, but these result in no improvement. As our initial data analysis suggest that demand behavior differs across sales channels, we use SVR to forecast demand per sales channel. This improves the MASE to 0.56 (5-week ahead forecast) and 0.60 (13-week ahead forecast).

Products are often used in combination, particularly for suppliers in the construction industry, and therefore show correlated demand. To leverage this information for our forecast, we apply sequential pattern mining, which identifies products with correlated demand by finding sequential patterns. Our results show that forecast accuracy improves when using this additional information with a dynamic regression model. However, it does not improve the results of the SVR model, which still delivers superior forecast accuracy. Consequently, we conclude that sequential pattern mining is inappropriate for use by ConstructX. However, for companies and datasets where traditional forecasting methods, such as dynamic regression, deliver superior results to machine learning methods, sequential pattern mining might be used to improve forecast accuracy.

Most previous research has only assessed the impact of ML methods on forecast accuracy. Therefore, in the last step of our research, we investigate the impact of the 5-week ahead forecast on the costs of inventory holding and lost sales at ConstructX. The results show that the company might decrease its net costs by up to 6.9% with an SVR forecast by sales channel. However, this decrease in net cost diminishes with high percentages of lost sales, relating to the fraction of inventory shortage resulting in loss of sales. This can be explained by the fact that our SVR forecast tends to under-forecast slightly more often than the company's existing forecast, approximated by an SES model. In an actual application of our forecasting model, the under-forecast could be re-

moved, enabling companies to decrease their net costs even with high lost sales percentages.

While the results presented in this study depend largely on characteristics of the dataset used, our analysis points to the potential benefits of using ML in product demand forecasting, and especially SVR in the context of intermittent-demand forecasting. We argue that ML methods may significantly outperform traditional forecasting methods, as shown with both the original and simulated datasets. However, when traditional methods provide better results, accounting for correlated demand in the forecast by using sequential pattern mining may be a lever to improve forecast accuracy.

Our research has some limitations. First, as actual historical demand data were unavailable, historical sales data is used to approximate demand. This may lead to inaccuracies in the analysis, as sales data do not accurately reflect unfulfilled demand (i.e., lost sales) or the time of demand, for instance because demand was fulfilled at a later point in time. Moreover, the impact of our forecasting model on the costs of inventory holding and lost sales is only assessed over a short period of time. In order to remove potential forecast bias and truly assess the impact of different forecast models, the application should be tested over a longer time period.

Future research might further investigate the use of sequential pattern mining in demand forecasting. Owing to the large number of intermittent-demand products in the company's dataset, our application of sequential pattern mining identifies only a very small set of products with correlated demand. It would be very interesting to see the application of sequential pattern mining, and its subsequent use in demand forecasting, in a dataset with products with much higher demand volumes, potentially resulting in a larger set of related products and larger improvements in forecast accuracy, possibly also when using ML methods.

5

CONCLUSION AND OUTLOOK

5.1 SUMMARY

The tremendous growth of data volumes from various sources has provided companies with new opportunities to shape and optimize their supply chain. Especially data from unstructured sources (e.g., social media, companies' websites, sensors) has the potential to provide companies with new insights to be used in their decision-making. While other fields, such as marketing, have already started to leverage insights drawn from data related to customer behavior, the use of such data in supply chain management (SCM) is novel, and empirical insights on the topic remain scarce.

This dissertation aims to contribute to both academia and practice by providing insights into the value of customer behavior in demand forecasting in three separate research papers. For this purpose, empirical data from two very different industry research partners are leveraged. The first two research papers focus on demand forecasting at an online retailer in a business-to-consumer (B2C) context, where customer behavior is observed through data collected from the company's online webshop. Specifically, customers' interaction with product sites is used as input to demand forecasting. The third research paper takes place in a B2B setting in the construction industry, where customers resemble other businesses whose behavior is investigated in the form of related product purchases. The insights from our research are three-fold.

First, as one of the main contributions of this dissertation, all three research papers highlight the value of customer behavior in demand forecasting. Chapters 2 and 3 show that clickstream data can be useful to enhance short-term demand forecasts. Specifically, Chapter 2 reveals that it is possible to very accurately predict customers' purchases right after they added a product to their shopping cart. This enables companies to implement anticipatory actions before customers place their orders, for instance, to optimize delivery times. Chapter 3 shows that customer clickstream data can be powerful in improving product-level demand forecasts, especially for medium-demand products as well as certain intermittent demand products. Similar to Chapter 2, 'add to cart' clicks are found to be a strong predictor for future demand. Chapter 4 highlights the value of using customers' historical purchase patterns in demand forecasting. Sequential pattern mining is applied to determine products with related demand — a task for which the involvement of company experts is usually required. Using the demand of these related products in the forecasting process shows improvements in forecast accuracy when using time-series models.

Second, this research highlights both the potential and limitations of machine learning in demand forecasting. Chapters 2 and 3 reveal that machine learning methods are able to leverage underlying behavioral patterns from customer behavior data to predict future demand. Especially Chapter 3 shows that machine

learning methods are better able to capture information from customer clickstream data compared to traditional time-series methods. Furthermore, Chapters 3 and 4 show that support vector regression (SVR) is especially well-suited to predict intermittent demand. However, these two chapters also outline that traditional time-series models often provide very similar, in some instances even better, forecast accuracy. In Chapter 3, for instance, dynamic regression, a time-series forecasting method, outperforms machine learning methods when forecasting the demand for high-demand products. As machine learning models are black-boxes that are difficult to interpret, time series methods, which are much easier to understand, might be preferred in practice, especially if their performance in terms of forecast accuracy is comparable.

Third, this dissertation highlights the importance of moving beyond metrics that solely measure forecast accuracy. Translating improvements in demand forecasts into the impact in an actual business application enables companies to reveal both the true benefits but also limitations of demand predictions. Chapter 2 shows that while it is possible to accurately predict customers' future orders, using those predictions to ship products in advance comes at a high cost. This is due to the large share of product site visits not resulting in a purchase, where small forecast errors cause a large number of products wrongly shipped in advance. Order predictions could therefore better be leveraged for other anticipatory processes, such as anticipatory picking and packaging. In Chapter 3, measures of forecast accuracy indicate the potential of using clickstream data in demand forecasting. However, the application of forecasts to order picking in the warehouse reveals that clickstream data should, in fact, not be used to forecast high-demand products. Lastly, while SVR outperforms all other methods in terms of forecast accuracy in Chapter 4, the application to inventory planning shows that time-series methods sometimes result in lower inventory holding and lost sales cost, despite lower forecast accuracy. This is because, for the dataset used, the SVR model tends to under-forecast, which can be expensive if the cost of lost sales from stockouts is high. Most research investigating machine learning models in forecasting do not typically assess forecast bias.

It should be noted that the results of this dissertation are largely dependent on the data quality provided. Especially for clickstream data, inaccurate tracking systems, or customers not being logged in can make it difficult for any analytical model to truly understand customer behavior.

In terms of generalizability, we believe that the results from Chapters 2 and 3 are also applicable to other online fashion retailers. Whether clickstream data in demand forecasting is also useful in other industries cannot be answered with this research, as the purchase decision for non-fashion items might be very different. The approach developed in Chapter 4 could equally be applied at any company with assortments containing products with related demand, such as the electronics industry, although this research does not answer whether the use of related products could help improve forecast accuracy at other companies.

5.2 OUTLOOK

This dissertation investigates different applications of advanced analytics to draw insights from customer behavior to enhance demand forecasts. While the insights discovered contribute to the literature on demand forecasting as well as (big) data analytics in SCM, several areas for future research are suggested.

The insights from Chapters 2 and 3 mainly relate to the fashion industry. While some studies have assessed the use of clickstream data for predictions in other industries (e.g. Qi et al. 2019), future research should assess to what extent the value of clickstream data in predictions differs across industries. Also, developing a deeper understanding of the purchase decision process for different types of products (e.g., fashion, electronics, beauty) could provide further insights to enhance prediction models.

As already outlined previously, sequential pattern mining could be a useful tool in the demand forecasting process. While our research shows that using related products as input to demand forecasting has the potential to improve forecast accuracy, it would be very interesting to validate the use of the developed approach on a dataset with much more related products.

Across all three research papers, various prediction methods are tested to determine the best-suited method for each dataset. Although machine learning models are seen as black boxes, approaches have been developed to make them more interpretable. Future research assessing why and when certain (machine learning) methods outperform others would be extremely powerful to reduce the effort needed to develop dataset- and application-specific prediction models.

In conclusion, the expected growth and future relevance of big data and advanced analytics stress the importance of developing efficient prediction models that can be used in practice. While this dissertation provides new insights into specific topics in this field, there are many aspects that remain for future research, of which just a few are presented above. We believe that the intersection of customer behavior, predictive analytics, and SCM provides a very promising agenda for future research.

A

APPENDIX TO CHAPTER 2

A.1 FORECASTING METHODS

To enable reproducibility of the results, Table A.1 presents all hyperparameters tested for each forecasting method, and the R package and function used.

Table A.1: Overview of forecasting methods and hyperparameters used

Method	Hyperparameters tested	R package	R function
RF	Ntree: 50, 500* Mtry: default, 4, 10*, 12	randomForest	randomForest()
NN	Neurons: 3, 5*, 7, c(2,1) c(2,2), c(3,2)	neuralnet	neuralnet()
SVM	Kernel: linear, radial* Cost: 0.1, 1*, 10 Gamma (only for radial kernel): 0.001, 0.01*, 0.1	e1071	svm()
One-class SVM	Kernel: radial Gamma: 0.01*, 0.1 Nu: 0.1, 0.5, 0.6, 0.8*	e1071	svm()
LG	-	glmnet	glm(family = binomial)
K-means	Nstart: 10 k: 1-10	stats	kmeans()

*Hyperparameters resulting in the highest AUPR value for the respective forecasting method.

A.2 SIMULATION ALGORITHM FOR ANTICIPATORY SHIPPING

Our simulation seeks to estimate the average delivery time, the number of orders fulfilled from the warehouse closest to the customer, and the number of orders wrongly shipped in advance without subsequent orders for orders from a set of premium customers in a two-week planning horizon. Those three metrics are compared with and without the application of anticipatory shipping. The latter is referred to as the baseline.

In the baseline, I_o^p denotes the inventory of product p after order o has been fulfilled. The starting inventory per product $I_{o=0}^p$ is equal to the total demand in the two weeks, which ensures that all orders can be fulfilled. To represent a supply chain with specialized warehouses, the starting inventory is allocated across warehouses so that each warehouse stocks an equal number of products. I_o^{pw} denotes the inventory of product p in warehouse w after order o. If an or-

der can be fulfilled from a warehouse that is closest to the customer, we assume an average delivery time d_o of $l_1 = 8$ h and $l_2 = 16$ h otherwise. To compare the baseline results to the results obtained from anticipatory shipping, we only simulate those orders in the baseline which contain products that occur on the list for anticipatory shipping, resulting in a total number of O orders simulated. For confidentiality reasons, this number cannot be stated. Moreover, one order is equal to the demand for one unit of a specific product. Orders are fulfilled from four different warehouses so that for each order and corresponding delivery address, warehouses are ranked according to their distance. Ideally, as many orders as possible are fulfilled from the closest warehouse. Table A.2 outlines the indices and parameters used while Figure A.1 demonstrates the algorithm to simulate inventory and order fulfillment in the baseline.

begin
  Set $I^p_{o=0}$:= demand for product p during the whole planning horizon T;
  Set $I^{pw}_{o=0} := I^p_{o=0}$ if product p was allocated to warehouse w, set to 0 otherwise;
  Set o := 1;
  Sort all orders o in O by their purchase time in ascending order
  for o *in* O **do**
    /* Determine delivery time for order o. */

$$d_o = \begin{cases} l_1, & \text{if } I^{pw}_{o-1} > 0 \text{ and } w = \text{closest warehouse to delivery address of o} \\ l_2, & \text{otherwise} \end{cases}$$

    /* For the warehouse the order was supplied from, update inventory. */

$$I^{pw}_o = I^{pw}_{o-1} - 1$$

  end
end

Figure A.1: Algorithm 1 (baseline simulation)

To simulate the application of anticipatory shipping, the parameter t is introduced, resembling the time of purchase or prediction, which is equal to the time of the first add to cart click for the latter. The time horizon of two weeks is therefore represented as t = 1, ..., 336 h. The additional parameter is needed so that products from a prediction can be reserved for customers. Additionally, a binary variable k is introduced, distinguishing whether an order is an actual order (k = 0) or predicted order (k = 1). The starting inventory $I^p_{t=0}$ of each product is now equal to the total demand in the two weeks plus the demand needed for anticipatory shipping. This starting inventory is allocated to the four warehouses in the same manner as in the baseline simulation. If a purchase is predicted, the product is shipped to the warehouse closest to the delivery address of the prediction, which takes $l_3 = 8$ h on average, and reserved for a period of r = 48 h, unless the closest warehouse already stocks the product. To simulate this, the notation of a *reservation* list is introduced, which stores information on the customer, product, and closest warehouse for

Table A.2: Notation table (baseline simulation)

Indices	
w	Inventory storage location (warehouse), w = {1,2,3,4}
p	Product, $p = 1, ..., P$ ($P = 579$)
o	Customer product order, $o = 1, ..., O$
Parameter	
I_o^p	Inventory of product p after order o
I_o^{pw}	Inventory of product p in warehouse w after order o
d_o	Delivery time of order o
l_1	Delivery time if an order can be fulfilled from the warehouse closest to the customer, $l_1 = 8$ h
l_2	Delivery time if an order can be fulfilled from the warehouse closest to the customer, $l_2 = 16$ h

which a purchase was predicted, as well as the time m until which the product is reserved. After the reservation period, the product is available again for all customers. If a customer, for which a product has been shipped in advance, purchases a product that has not arrived at the closest warehouse yet, then the customer's delivery time is equal to $l_1 = 8$ h plus the time the order remains in transit to the warehouse. As the actual delivery address for an order might be different from the prediction, for instance, because a customer ordered products to a different delivery address in the past, a check is made whether the warehouse from the predicted order on the reservation list is the same as the warehouse closest to the delivery address of the actual order. Table A.3 outlines the indices and parameters used while Figure Figure A.2 demonstrates the algorithm to simulate inventory and order fulfillment with an application of anticipatory shipping.

begin

Set $I^{p}_{t=0}$:= demand for product p during the planning horizon T + inventory for anticipatory shipping;

Set $I^{pw}_{t=0}$:= $I^{p}_{t=0}$ if product p was allocated to warehouse w, set to 0 otherwise;

Set o := 1;

Set t := 1;

Sort all orders o in O by their purchase time (or time of prediction) in ascending order

for o_k *in* O **do**

/* Check for each order if t needs to be updated and update starting inventory accordingly. */

if $t_o \neq t$ **then**

$$t = t + 1$$

$$I^{pw}_{t} = I^{pw}_{t} + I^{pw}_{t-1}$$

end

if $k = 1$ *and* $I^{pw}_{t} = 0$ *in the warehouse closest to the delivery address* **then**

/* Calculate m_o and add order to reservation list. */

$$m_o = t_o + r$$

$$reservation = reservation + o_k$$

/* For the warehouse the prediction was supplied from, update inventory. Add starting inventory in the warehouse prediction was shipped to. */

$$I^{pw}_{t} = I^{pw}_{t} - 1$$

$$I^{pw}_{t+r} = I^{pw}_{t+r} + 1$$

else

if *customer, product, and closest warehouse of the order are on the reservation list and* $t_o < m_o$ **then**

/* Determine delivery time for order, incl. potential waiting time for products in transit and remove order from reservation list. */

$$d_o = l_1 + \max\{0, m - t - (r - l_3)\}$$

$$reservation = reservation - o_k$$

else

/* Determine delivery time for the order. For warehouse the order was supplied from, update inventory. */

$$d_o = \begin{cases} l_1, & \text{if } I^{pw}_{t} > 0 \text{ and } w = \text{closest warehouse to delivery address of o} \\ l_2, & \text{otherwise} \end{cases}$$

$$I^{pw}_{t} = I^{pw}_{t} - 1$$

end

end

end

end

Figure A.2: Algorithm 2 (anticipatory shipping simulation)

Table A.3: Notation table (anticipatory shipping simulation)

Indices	
w	Inventory storage location (warehouse), w = {1,2,3,4}
p	Product, $p = 1, ..., P$ ($P = 579$)
o	Customer product order, $o = 1, ..., O$
k	Binary variable indicating whether o is an actual ($k = 0$) or predicted order ($k = 1$)
t	Time stamp (hour), $t = 1, ..., T$ ($T = 336$)
Parameter	
I_t^p	Inventory of product p at time stamp t
I_t^{pw}	Inventory of product p in warehouse w at time stamp t
o_k	Order of type k
d_o	Delivery time of order o
t_o	Time stamp of order o
l_1	Delivery time if an order can be fulfilled from the warehouse closest to the customer, $l_o = 8$ h
l_2	Delivery time if an order can be fulfilled from the warehouse closest to the customer, $l_2 = 16$ h
l_3	Time to ship an order from one warehouse to another, $l_3 = 8$ h
r	Reservation period, $r = 48$ h
m_o	Time until which the product of a predicted order o is reserved for a customer
Lists	
reservation	List containing information on product, customer and closest warehouse for predicted orders

B

APPENDIX TO CHAPTER 3

B.1 FORECAST METHODS

To enable reproducibility of the results, Table B.1 presents all hyperparameters tested for each forecasting method, and the R package and function used.

Table B.1: Overview of forecasting methods and hyperparameters used

Method	Hyperparameters tested	R package	R function
DR	Automatically selected	forecast	auto.arima()
RF	Ntree: 500, 5000	randomForest	randomForest()
SVR	Kernel: linear, radial Cost: $10^{(-3:2)}$ Gamma: $10^{(-3:1)}$	e1071	svm()
FFNN	Neurons: 1, 2, 3, (1,1)	neuralnet	neuralnet()
JRNN	Neurons: 1, 3, 4, 6 Learning rate: $10^{(-5:-1)}$ Max. iterations: 1000, 2000	RSNNS	jordan()
ERNN	Neurons: 3, 6, (2,2), (6,6), (12,12) Learning rate: $10^{(-5:-1)}$ Max. iterations: 1000, 2000	RSNNS	elman()
Naïve	-	forecast	naive()

B.2 CLUSTERING

The time series features used for clustering based on demand characteristics are given in Table B.2. These are a subset of the features outlined in Hyndman et al. (2015). They aim to capture global information in time series, and can be computed using the tsmeasures package in R. As the function identifies no seasonal component for many of the intermittent-demand products, the features concerning season are not computed. For further explanation of the features used, see Hyndman et al. (2015).

Table B.2: Features for time-series clustering using demand characteristics

Feature	Description
Lumpiness	Variance of annual variances of remainder
Entropy	Spectral entropy
ACF1	Autocorrelation (first order)
Lshift	Level shift using a rolling window
Vchange	Change in variance
Cpoints	Number of crossing points
Fspots	Flat spots (using discretization)
Trend	Strength of trend
Linearity	Strength of linearity
Curvature	Strength of curvature
Spikiness	Strength of spikiness
KLscore	Kullback-Leibler score
Change.idx	Index of the maximum KL score

B.3 INITIAL DATA ANALYSES

In the 3,000-product dataset, very few products drive most of the demand (Figure B.1). Moreover, analysis of orders per weekday shows that the highest order volumes are placed on Sundays, and volumes decrease continuously over the following weekdays (Figure B.2).

To assess possibilities for combining clickstream variables in forecasting, we assess pairwise correlations (Table B.3). As most variables have a high correlation (pearson correlation (r)> 0.7), as can be expected from such a dataset, there are limited possibilities for variable combinations.

B.4 SIMULATION ALGORITHM FOR ORDER PICKING

In Figure B.3, we outline the algorithm used to simulate order picking. The list of indices and parameters can be found in Table 3.1 in chapter 3.

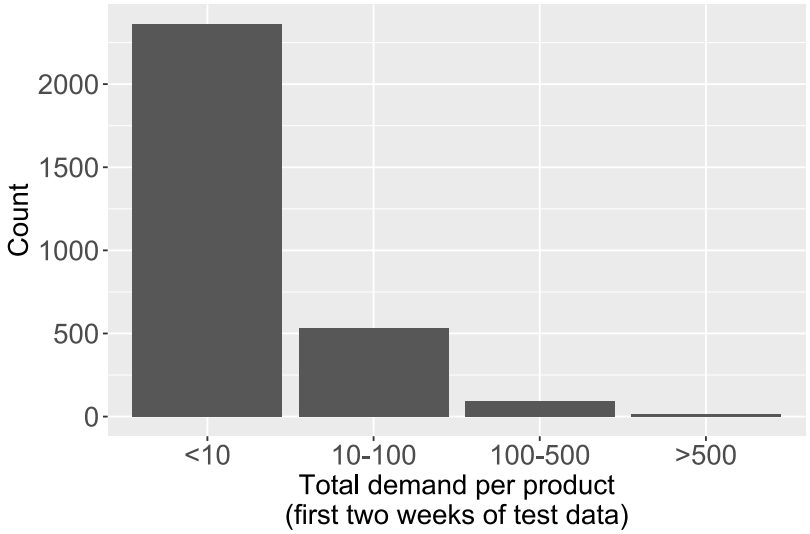

Figure B.1: Demand distribution across products

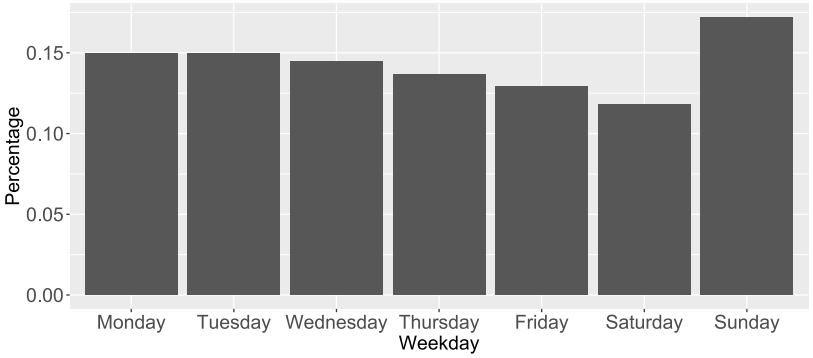

Figure B.2: Order volume across weekdays

Table B.3: Pairwise correlations (numerical variables)

Variable	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.00																							
2	0.55	1.00																						
3	0.55	0.99	1.00																					
4	0.57	0.99	0.98	1.00																				
5	0.17	0.23	0.24	0.26	1.00																			
6	0.18	0.26	0.28	0.29	0.78	1.00																		
7	0.67	0.82	0.80	0.84	0.21	0.23	1.00																	
8	0.54	0.91	0.91	0.91	0.22	0.25	0.79	1.00																
9	0.57	0.96	0.95	0.97	0.26	0.29	0.85	0.89	1.00															
10	0.18	0.25	0.26	0.27	0.67	0.66	0.23	0.24	0.32	1.00														
11	0.03	0.04	0.04	0.05	0.10	0.11	0.04	0.04	0.09	0.35	1.00													
12	0.56	0.99	0.98	1.00	0.25	0.28	0.83	0.91	0.96	0.26	0.05	1.00												
13	0.54	1.00	0.99	0.97	0.21	0.25	0.81	0.91	0.95	0.24	0.04	0.97	1.00											
14	0.53	0.91	0.91	0.91	0.22	0.25	0.78	0.84	0.89	0.24	0.04	0.91	0.90	1.00										
15	0.53	0.98	0.97	0.96	0.22	0.24	0.80	0.89	0.94	0.24	0.04	0.97	0.98	0.89	1.00									
16	0.54	0.98	0.98	0.97	0.23	0.26	0.80	0.9	0.95	0.25	0.04	0.97	0.98	0.87	0.93	1.00								
17	0.06	0.08	0.09	0.09	0.29	0.26	0.07	0.08	0.09	0.25	0.03	0.09	0.08	0.08	0.08	0.08	1.00							
18	0.06	0.06	0.06	0.06	0.32	0.23	0.06	0.05	0.07	0.21	0.03	0.06	0.05	0.06	0.05	0.05	0.13	1.00						
19	0.71	0.77	0.77	0.78	0.23	0.26	0.84	0.74	0.78	0.24	0.04	0.77	0.76	0.73	0.74	0.75	0.08	0.07	1.00					
20	0.71	0.76	0.77	0.78	0.24	0.26	0.84	0.74	0.77	0.24	0.04	0.77	0.75	0.73	0.74	0.75	0.08	0.07	0.99	1.00				
21	0.72	0.76	0.76	0.78	0.25	0.27	0.85	0.74	0.78	0.25	0.04	0.78	0.74	0.73	0.74	0.75	0.08	0.08	0.99	0.99	1.00			
22	0.73	0.72	0.71	0.74	0.22	0.24	0.92	0.7	0.75	0.24	0.04	0.73	0.70	0.69	0.69	0.70	0.07	0.07	0.93	0.92	0.93	1.00		
23	0.66	0.71	0.71	0.72	0.22	0.24	0.79	0.78	0.73	0.24	0.04	0.72	0.70	0.67	0.69	0.70	0.07	0.07	0.92	0.92	0.92	0.86	1.00	
24	0.72	0.75	0.75	0.77	0.24	0.26	0.86	0.74	0.79	0.28	0.07	0.77	0.74	0.72	0.73	0.74	0.08	0.08	0.98	0.98	0.98	0.94	0.91	1.00

begin
  Sort o_{kth} by their purchase time in ascending order;
  Set $r := 1$; $m := 1$; $t := 1$, $h := 0$;
  Set $c :=$ number of available bags resulting from dataset and capacity level used;
  Set $R_{kt} := F_{kt}$ and sort products k in R_{kt} by forecasted volume in descending order;
  Set $k := 1$ and sort products k in I_{kth} so that they have the same order as in R_{kt}
  /* Initialize starting inventory in PS. */
  while $\sum_k^K I_{kth} < c$ **do**
    if $I_{kth} < m$ *and* $R_{kt} > 0$ **then**
      $I_{kth} = I_{kth} + 1$
      $R_{kt} = R_{kt} - 1$
    else
      $k = k + 1$
    end
  end
  /* Check picking time for product order. */
  for o_{kt} *in* O **do**
    if $I_{kth} > 0$ **then**
      $z_o = 1$
      $I_{kth} = I_{kth} - 1$
    else
      $z_o = 0$
      /* Remove from F_{kt} if product was forecasted. */
      if $F_{kt} > 0$ **then**
        $R_{kt} = R_{kt} - 1$
    end
    /* Decide on internal replenishment. */
    if h *reaches the next hour without any further order arriving* **then**
      $I_{kth} = I_{kth} + I_{kt,h+1}$
      $h = h + 1$
      Sort products k in R_{kt} by forecasted volume in descending order
      Sort products k in I_{kth} so that they have the same order as in R_{kt}
      $k = 1$
      while $\sum_k^K I_{kth} + \sum_k^K I_{kt,h+1} < c$ **do**
        if $I_{kth} > 0$ *and* $\sum_{t-2}^{t} o_{kt} = 0$ **then**
          $k = k + 1$
        else
          /* If there are less than 3 units in PS and the product has a forecasted volume. */
          if $\sum_h^{h+1} I_{kth} < m$ *and* $R_{kt} > 0$ **then**
            $I_{kth} = I_{kth} + I_{kt,h+1}$
            $R_{kt} = R_{kt} - 1$
          else
            $k = k + 1$
          end
        end
      end
    end
    /* End of day, update remaining forecast with forecast for next day. */
    if t *reaches the next day without any further order arriving* **then**
      $t = t + 1$
      $R_{kt} = F_{kt} - I_{kth} - I_{kt,h+1}$
    end
  end
end

Figure B.3: Simulation algorithm (order picking)

C

APPENDIX TO CHAPTER 4

C.1 FORECASTING METHODS

To enable reproducibility of the results, Table C.1 presents all (hyper-)parameters tested for each forecasting method, and the R package and function used. To identify the best-performing parameters, cross-validation across 14 weeks is performed.

ML methods are run both across all products (AP) at once, to make use of their ability to learn across time series, as well as per product (PP). Figures C.1 and C.2 show examples of using the last six demand lags for a 5-week ahead forecast when applying ML methods per product and across products. Here, X_t is equal to the lagged demand in week t, representing the input variables for the ML methods, and Y_t is the demand in week t, representing the response variable.

Table C.1: Overview of forecasting methods and (hyper-)parameters used

Method	Parameters tested	R package	R function
SMA	Order: 6, 7, 8	forecast	ma()
SES	Automatically selected	forecast	ses()
Croston	Alpha: 0.01, 0.1, 0.2	forecast	croston()
ARIMA	Automatically selected	forecast	auto.arima()
DR	Automatically selected	forecast	auto.arima()
DHR	Fourier term K: 1, 2	forecast	auto.arima()
TBATS	Automatically selected	forecast	tbats()
RF	Ntree: 500, 5000	randomForest	randomForest()
SVR	Kernel: linear, radial Cost: 0.001, 0.01, 0.1, 1 Gamma: 0.001, 0.01, 0.1	e1071	svm()
NN	Neurons: c(2,1), c(2,2)	neuralnet	neuralnet()
JRNN	Neurons: 1, 3, 5 Learning rate: 0.01, 0.1, 0.2 Max. iterations: 1000, 2000	RSNNS	jordan()
ERNN	Neurons: c(3,2), c(6,4) Learning rate: 0.01, 0.1, 0.2 Max. iterations: 1000, 2000	RSNNS	elman()

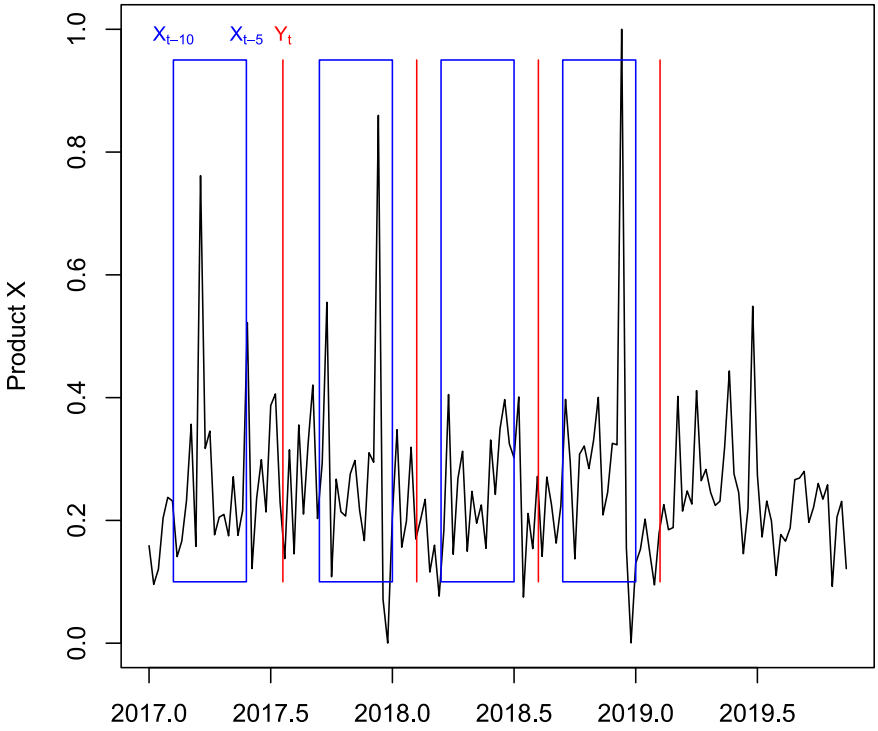

Figure C.1: Training machine learning methods per product (5-week ahead forecast with variable set 1)

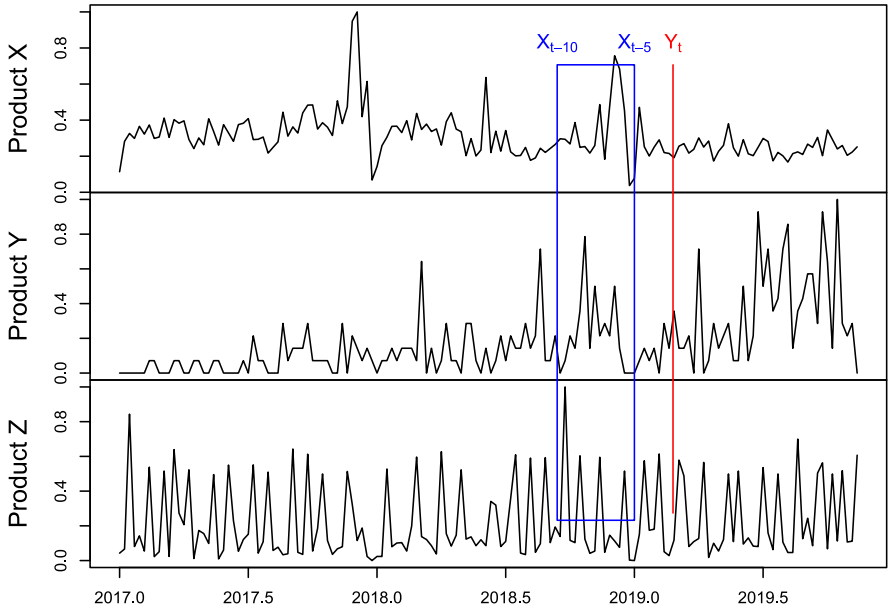

Figure C.2: Training machine learning methods across products (5-week ahead forecast with variable set 1)

C.2 CLUSTERING

The time-series features used for clustering are given in Table C.2.

Table C.2: Features for time series clustering

Feature	Description
CV	Coefficient of variation, measuring the variance in demand height
ADI	Average demand interval, measuring the average interval between two periods of non-zero demand
Nonzero	Number of non-zero demand periods
Skewness	Measure describing the symmetry of the data
Kurtosis	Peak measure, characterizing the tail extremity of a distribution

C.3 DATA SIMULATION

To test the generalizability of the forecasting methods used, we apply them to a simulated dataset. The new data are simulated using the average demand interval (ADI), the squared coefficient of variation (CV2), and the mean level of demand in non-zero demand periods (Level) of the original data from ConstructX. For each cluster constructed with hierarchical clustering, the average of these three metrics across all products is calculated (Table C.3) and used to simulate exactly the same number (N) of time series, resulting in a dataset containing 3,000 products.

Table C.3: Assessment of clusters for data simulation

Cluster	N	ADI	CV2	Level
1	1,433	1.88	0.77	6,027
2	935	8.29	0.92	2,772
3	550	17.61	2.12	1,331
4	81	1.22	3.08	1,669

C.4 INVENTORY PLANNING MODEL

We assume that ConstructX follows an (r, S) inventory policy, meaning that every r time periods, the company places sufficient replenishment orders to restore the on-hand inventory to target level S. An overview of the model notation can be found in Table C.4.

In our inventory planning model, r is equal to 1, meaning that replenishment orders are placed weekly. The order up-to level $S_{p,t}$ for each product p in week t is determined by the forecast demand $\hat{y}_{p,t}$ plus the safety stock SS_p:

$$S_{p,t} = \hat{y}_{p,t} + SS_p \tag{C.1}$$

$$SS_p = z\sqrt{\hat{\sigma}_p^2 LT} \tag{C.2}$$

Parameter z is the standard score, which is based on the company's service level. For a 98% alpha-service level, z is equal to 2.05. $\hat{\sigma}_p^2$ refers to the variance of the forecast error. Equations (C.3)-(C.6) outline how inventory develops over time:

$$I_{p,t}^b = I_{p,t-1}^e + x_{p,t-LT} \tag{C.3}$$

$$I_{p,t}^f = |min\left\{0, I_{p,t}^b - y_{p,t} - I_{p,t-1}^f\right\}| \tag{C.4}$$

$$I_{p,t}^e = max\left\{0, I_{p,t}^b - y_{p,t} - I_{p,t-1}^f \times (1-c)\right\} \tag{C.5}$$

$$I_{p,t}^d = I_{p,t}^e + \sum_{t=t-1}^{t=t-LT} x_{p,t} \tag{C.6}$$

Equation (C.3) ensures that the inventory at the beginning of the week $I_{p,t}^b$ is equal to the inventory at the end of the previous week $I_{p,t-1}^e$ plus the replenishment order x_{t-LT} placed earlier with respect to lead time LT. We assume that replenishment orders are always available at the beginning of the week. An inventory shortage $I_{p,t}^f$ occurs when the inventory at the beginning of the week is insufficient to fulfill the weekly demand and any inventory shortage from the previous week (C.4). An inventory shortage results in either lost sales or backorders to be fulfilled in subsequent weeks, and is determined by the lost sales percentage c, which is assessed for a range from 10% to 50%. The inventory at the end of the week $I_{p,t}^e$ is calculated by subtracting the weekly demand $y_{p,t}$ and backorders from the previous week from the inventory at the beginning of the week (C.5). Lastly, the inventory position is defined as the amount of inventory on hand at the end of the week plus any inventory from outstanding replenishment orders (C.6).

If the inventory position at the end of the week is insufficient to fulfill the forecasted demand $\hat{y}_{p,t}$ during lead time LT and review period r, the required safety stock and the current inventory shortage, then a replenishment order $x_{p,t}$ is placed (C.7):

$$x_{p,t} = max\left\{0, \sum_{t=t+1}^{t=t+r+LT} \hat{y}_{p,t} + SS_p + I_{p,t}^f \times c - I_{p,t}^d\right\} \tag{C.7}$$

At the beginning of our 14-week planning horizon, the forecast error is not yet known. Therefore, we simulate an additional 14 weeks of inventory prior to our planning horizon, referred to as the initialization period. This is necessary to estimate the safety stock, inventory shortage, and inventory at the end of week $t = 0$. The forecast error used to estimate the safety stock is therefore calculated based on the demand and forecast during the initialization period. As no data on inventory or backorders were provided by ConstructX, we assume that there was no inventory shortage at the start of the initialization period (C.8), and set inventory at the end of week before the initialization period equal to the safety stock (C.9).

$$I_{p,t=-T}^f = 0 \tag{C.8}$$

$$I_{p,t=-T}^e = SS_p \tag{C.9}$$

To assess the impact of different forecasting methods, the total cost of lost sales L and total inventory holding cost are calculated across the 14-week planning horizon with:

$$L = \sum_{t=1}^{T} \sum_{p=1}^{P} I_{t,p}^{f} \times s_p \times m \times c \tag{C.10}$$

$$C = h \times \sum_{p=1}^{P} \frac{s_p \times (1-m)}{T} \sum_{t=1}^{T} \frac{I_{t,p}^{b} + I_{t,p}^{e}}{2} \tag{C.11}$$

Parameter s_p reflects the indexed sales price, the index being known only to ConstructX. The profit margin m is assumed to be 15%. Lastly, the inventory holding cost h is assumed to be 15% of the cost of goods sold. In a last step, the total cost of lost sales is extrapolated to one year in order to compare it with the total inventory holding cost.

Table C.4: Notation table

Indices	
t	Time period (week), $t = -T,...,T$ ($T = 14$)
p	Product in product portfolio, $p,...,P$ ($P = 22{,}000$)
IParameter and variables	
L	Total cost of lost sales
C	Total inventory holding cost
r	Review period, $r = 1$ week
$S_{p,t}$	Order up-to level for each product p in week t
$\hat{\sigma}_p^2$	Variance of the forecast error for product p during the initialization period
LT	Lead time to replenish inventory, $LT = 4$ weeks
α	alpha-service level, $\alpha = 98\%$
z	Standard score associated with alpha-service level of 98%, $z = 2.05$
$I_{p,t}^{b}$	Inventory of product p at the beginning of week t
$I_{p,t}^{f}$	Inventory shortage of product p at the end of week t
$I_{p,t}^{e}$	Inventory of product p at the end of week t
$I_{p,t}^{d}$	Inventory position of product p at the end of week t
$\hat{y}_{p,t}$	Forecast for product p in week t
$y_{p,t}$	Demand for product p in week t
$x_{p,t}$	Replenishment order placed for product p at the end of week t
s_p	Indexed sales price of product p
c	Percentage of lost sales, $c = \{10\%, 20\%, 30\%, 40\%; 50\%\}$
m	Profit margin, $m = 15\%$
h	Inventory holding cost, $h = 15\%$

C.5 FORECAST BIAS

Forecast bias is the tendency of a forecast to either under- or over-forecast. In the case of under-forecasting, the forecast values are on average lower than the actual values. A common measure to quantify forecast bias is the tracking signal. From the various formulae available to measure a form of tracking signal, we choose to calculate the ratio of the cumulative sum of forecast errors to their mean absolute deviation, as outlined by (Venkataraman and Pinto 2016). According to Ravi (2014), a typical rule of thumb is that for tracking signals in the range of ± 4, a forecast is assumed to work correctly, with no bias. To be more conservative, we used a range of ± 3 to flag forecast bias. According to Valentini and Dietterich (2004), the cost and gamma parameters of SVR may have a large impact on forecast bias. Specifically, low values may cause forecast bias. Increasing the two parameters to the next higher number that we test (Table C.1), i.e., increasing cost from 0.1 to 1 and gamma from 0.01 to 0.1, improves the forecast bias tremendously with no great impact on forecast accuracy. Table C.5 shows the impact of the change in SVR parameters on forecast bias, as well as the forecast bias when using SES.

Table C.5: Evaluation of forecast bias

Forecast method	Parameters	Products with an under-forecast	Products with an over-forecast	Products without forecast bias
SES	Automatically selected	21%	31%	48%
SVR	Kernel: radial Cost: 0.1 Gamma: 0.01	46%	20%	34%
SVR	Kernel: radial Cost: 1 Gamma: 0.1	32%	22%	46%

BIBLIOGRAPHY

Agrawal, R., Imieliński, T., Swami, A., 1993. Mining association rules between sets of items in large databases, in: Buneman, P., Jajodia, S. (Eds.), Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY. pp. 207–216.

Agrawal, R., Srikant, R., 1995. Mining sequential patterns, in: Agrawal, R., Srikant, R. (Eds.), Proceedings of the Eleventh International Conference on Data Engineering, IEEE Comput. Soc. Press. pp. 3–14.

Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H., 2010. An empirical comparison of machine learning models for time series forecasting. Econometric Reviews 29, 594–621.

Akter, S., Wamba, S.F., 2016. Big data analytics in e-commerce: A systematic review and agenda for future research. Electronic Markets 26, 173–194.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J., 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54, 627–635.

Bandara, K., Bergmeir, C., Hewamalage, H., 2020a. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. IEEE Transactions on Neural Networks and Learning Systems.

Bandara, K., Bergmeir, C., Smyl, S., 2020b. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. Expert Systems with Applications 140, 112896.

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., Seaman, B., 2019. Sales demand forecast in e-commerce using a long short-term memory neural network methodology, in: Gedeon, T., Wong, K.W., Lee, M. (Eds.), Neural Information Processing: 26th international conference, ICONIP 2019, proceedings part III. Springer, Cham, Switzerland, pp. 462–474.

Bergmann, H., 2018. Versand- und Retourenmanagement im E-Commerce 2018. URL: `https://www.ehi-shop.de/image/data/PDF_Leseproben/EHI-Studie_Versand-Retourenmanagement_im_E-Commerce_2018_LP.pdf`.

Borovykh, A., Sander, B., Oosterlee, C., 2018. Dilated convolutional neural networks for time series forecasting. Journal of Computational Finance 22, 73–101.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2016. Time series analysis: Forecasting and control (Wiley series in probabilit and statistics), 5th Edition. Wiley, Hoboken, NJ.

Boyd, D.E., Bahn, K.D., 2009. When do large product assortments benefit consumers? An information-processing perspective. Journal of Retailing 85, 288–297.

Boysen, N., de Koster, R., Weidinger, F., 2019. Warehousing in the e-commerce era: A survey. European Journal of Operational Research 277, 396–411.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Briedis, H., Kronschnabl, A., Rodriguez, A., Ungerman, K., 2020. Adapting to the next normal in retail: The customer experience imperative. URL: www.mckinsey.com/industries/retail/our-insights/adapting-to-the-next-normal-in-retail-the-customer-experience-imperative.

Brinch, M., Stentoft, J., Jensen, J.K., Rajkumar, C., 2018. Practitioners understanding of big data and its applications in supply chain management. The International Journal of Logistics Management 29, 555–574.

Brown, B., Chui, M., Manyika, J., 2011. Are you ready for the era of big data? McKinsey Quarterly 4, 24–35.

Carbonneau, R., Laframboise, K., Vahidov, R., 2008. Application of machine learning techniques for supply chain demand forecasting. European Journal of Operational Research 184, 1140–1154.

Centre for Retail Research, 2020. Online: UK, Europe & N. America 2020 estimates. URL: www.retailresearch.org/online-retail.html.

Chang, P.C., Liu, C.H., Fan, C.Y., 2009. Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. Knowledge-Based Systems 22, 344–355.

Chen, I., Lu, C., 2017. Sales forecasting by combining clustering and machine-learning techniques for computer retailing. Neural Computing and Applications 28, 2633–2647.

Chen, I.F., Lu, C.J., 2016. Sales forecasting by combining clustering and machine-learning techniques for computer retailing. Neural Computing and Applications 28, 2633–2647.

Chen, Y.L., Kuo, M.H., Wu, S.Y., Tang, K., 2009. Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data. Electronic Commerce Research and Applications 8, 241–251.

Chen, Z.Y., Fan, Z.P., Sun, M., 2012. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. European Journal of Operational Research 223, 461–472.

Choi, Y., Lee, H., Irani, Z., 2018. Big data-driven fuzzy cognitive map for prioritising IT service procurement in the public sector. Annals of Operations Research 270, 75–104.

Chong, A.Y.L., Li, B., Ngai, E.W., Ch'ng, E., Lee, F., 2016. Predicting online product sales via online reviews, sentiments, and promotion strategies. International Journal of Operations & Production Management 36, 358–383.

Chu, C.W., Zhang, G.P., 2003. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of Production Economics 86, 217–231.

Cirqueira, D., Hofer, M., Nedbal, D., Helfert, M., Bezbradica, M., 2020. Customer purchase behavior prediction in e-commerce: A conceptual framework and research agenda, in: Ceci, M., Loglisci, C., Manco, G., Masciari, E., Ras, Z. (Eds.), New Frontiers in Mining Complex Patterns. Springer, Cham, Switzerland, pp. 119–136.

Croston, J.D., 1972. Forecasting and stock control for intermittent demands. Journal of the Operational Research Society 23, 289–303.

Cui, R., Gallino, S., Moreno, A., Zhang, D.J., 2018. The operational value of social media information. Production and Operations Management 27, 1749–1769.

Davis, J., Goadrich, M., 2006. The relationship between precision-recall and ROC curves, in: W. Cohen, A. Moore (Eds.), Proceedings of the 23rd International Conference on Machine Learning, ACM, New York, NY. pp. 233–240.

De Caigny, A., Coussement, K., de Bock, K.W., 2018. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research 269, 760–772.

De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International Journal of Forecasting 22, 443–473.

De Livera, A.M., Hyndman, R.J., Snyder, R.D., 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association 106, 1513–1527.

Deriyenko, T., Hartkopp, O., Mattfeld, D.C., 2017. Supporting product optimization by customer data analysis, in: Dörner, K.F., Ljubic, I., Pflug, G., Tragler, G. (Eds.), Operations Research Proceedings 2015. Springer, Cham, Switzerland, pp. 491–496.

Elman, J.L., 1990. Finding structure in time. Cognitive Science 14, 179–211.

Fahimnia, B., Pournader, M., Siemsen, E., Bendoly, E., Wang, C., 2019. Behavioral operations and supply chain management–A review and literature mapping. Decision Sciences 50, 1127–1183.

Feng, Q., Shanthikumar, J.G., 2018. How research in production and operations management may evolve in the era of big data. Production and Operations Management 27, 1670–1684.

Ferreira, K.J., Lee, B.H.A., Simchi-Levi, D., 2016. Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing & Service Operations Management 18, 69–88.

Fildes, R., Goodwin, P., 2007. Against your better judgment? How organizations can improve their use of management judgment in forecasting. Interfaces 37, 570–576.

Fildes, R., Goodwin, P., Lawrence, M., Nikolopoulos, K., 2009. Effective forecasting and judgmental adjustments: An empirical evaluation and strategies for improvement in supply-chain planning. International Journal of Forecasting 25, 3–23.

Fildes, R., Ma, S., Kolassa, S., 2019. Retail forecasting: Research and practice. International Journal of Forecasting.

Garnier, R., Belletoile, A., 2019. A multi-series framework for demand forecasts in e-commerce. arXiv preprint arXiv:1905.13614.

Gessner, G.H., Volonino, L., 2005. Quick response improves returns on business intelligence investments. Information Systems Management 22, 66–74.

Goodwin, P., Fildes, R., 1999. Judgmental forecasts of time series affected by special events: Does providing a statistical forecast improve accuracy? Journal of Behavioral Decision Making 12, 37–53.

Gu, J., Goetschalckx, M., McGinnis, L.F., 2010. Solving the forward-reserve allocation problem in warehouse order picking systems. Journal of the Operational Research Society 61, 1013–1021.

Guan, M., Cha, M., Li, Y., Wang, Y., Sun, J., 2020. From anticipation to action: Data reveal mobile shopping patterns during a yearly mega sale event in China. IEEE Transactions on Knowledge and Data Engineering.

Gutierrez, R.S., Solis, A.O., Mukhopadhyay, S., 2008. Lumpy demand forecasting using neural networks. International Journal of Production Economics 111, 409–420.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction, 2nd Edition. Springer, New York, NY.

Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. International Journal of Production Economics 154, 72–80.

Hill, T., O'Connor, M., Remus, W., 1996. Neural network models for time series forecasts. Management Science 42, 1082–1092.

Hofmann, E., Rutschmann, E., 2018. Big data analytics and demand forecasting in supply chains: A conceptual analysis. The International Journal of Logistics Management 29, 739–766.

Hu, M., Monahan, S.T., 2016. US e-commerce trends and the impact on logistics. URL: `https://www.kearney.com/article/?/a/us-e-commerce-trends-and-the-impact-on-logistics`.

Huang, T., Van Mieghem, J.A., 2014. Clickstream data and inventory management: Model and empirical analysis. Production and Operations Management 23, 333–347.

Hübner, A., Holzapfel, A., Kuhn, H., 2015. Operations management in multi-channel retailing: An exploratory study. Operations Management Research 8, 84–100.

Hyndman, R.J., 2010. The ARIMAX model muddle. URL: `https://robjhyndman.com/hyndsight/arimax`.

Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and practice, 2nd Edition. OTexts [Online]. https://www.otexts.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679–688.

Hyndman, R.J., Wang, E., Laptev, N., 2015. Large-scale unusual time series detection, in: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE. pp. 1616–1619.

Iwanaga, J., Nishimura, N., Sukegawa, N., Takano, Y., 2016. Estimating product-choice probabilities from recency and frequency of page views. Knowledge-Based Systems 99, 157–167.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An introduction to statistical learning. Springer, New York, NY.

Kache, F., Seuring, S., 2017. Challenges and opportunities of digital information at the intersection of big data analytics and supply chain management. International Journal of Operations & Production Management 37, 10–36.

Ketchen, D.J., Shook, C.L., 1996. The application of cluster analysis in strategic management research: An analysis and critique. Strategic Management Journal 17, 441–458.

Khan, S.S., Madden, M.G., 2014. One-class classification: Taxonomy of study and review of techniques. The Knowledge Engineering Review 29, 345–374.

Kim, Y., Street, W.N., Russell, G.J., Menczer, F., 2005. Customer targeting: A neural network approach guided by genetic algorithms. Management Science 51, 264–276.

Kolkova, A., 2020. The application of forecasting sales of services to increase business competitiveness. Journal of Competitiveness 12, 90–105.

Kremer, M., Siemsen, E., Thomas, D.J., 2016. The sum and its parts: Judgmental hierarchical forecasting. Management Science 62, 2745–2764.

Kulkarni, G., Kannan, P.K., Moe, W., 2012. Using online search data to forecast new product sales. Decision Support Systems 52, 604–611.

Lau, R.Y.K., Zhang, W., Xu, W., 2018. Parallel aspect-oriented sentiment analysis for sales forecasting with big data. Production and Operations Management 27, 1775–1794.

Lee, C.K.H., 2017. A GA-based optimisation model for big data analytics supporting anticipatory shipping in retail 4.0. International Journal of Production Research 55, 593–605.

Lessmann, S., Baesens, B., Mues, C., Pietsch, S., 2008. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. IEEE Transactions on Software Engineering 34, 485–496.

Lessmann, S., Voß, S., 2009. A reference model for customer-centric data mining with support vector machines. European Journal of Operational Research 199, 520–530.

Leung, K.H., Choy, K.L., Siu, P.K., Ho, G., Lam, H.Y., Lee, C.K., 2018. A B2C e-commerce intelligent system for re-engineering the e-order fulfilment process. Expert Systems with Applications 91, 386–401.

Liu, D.R., Lai, C.H., Lee, W.J., 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. Information Sciences 179, 3505–3519.

Ljung, G.M., Box, G.E.P., 1978. On a measure of lack of fit in time series models. Biometrika 65, 297–303.

Lo, C., Frankowski, D., Leskovec, J., 2016. Understanding behaviors that lead to purchasing, in: Krishnapuram, B., Shah, M., Smola, A., Aggarwal, C., Shen, D., Rastogi, R. (Eds.), Proceedings of the 22nd ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY. pp. 531–540.

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

Maimon, O., Rokach, L., 2010. Data mining and knowledge discovery handbook, 2nd Edition. Springer, Boston, MA.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. The M4 competition: Results, findings, conclusion and way forward. International Journal of Forecasting 34, 802–808.

Matthias, O., Fouweather, I., Gregory, I., Vernon, A., 2017. Making sense of big data – can it transform operations management? International Journal of Operations & Production Management 37, 37–55.

McAfee, A., Brynjolfsson, E., Davenport, T.H., Patil, D.J., Barton, D., 2012. Big data: The management revolution. Harvard Business Review 90, 60–68.

Mizuno, M., Saji, A., Sumita, U., Suzuki, H., 2008. Optimal threshold analysis of segmentation methods for identifying target customers. European Journal of Operational Research 186, 358–379.

Moe, W.W., Fader, P.S., 2004. Dynamic conversion behavior at e-commerce sites. Management Science 50, 326–335.

Montgomery, A.L., Li, S., Srinivasan, K., Liechty, J.C., 2004. Modeling online browsing and path analysis using clickstream data. Marketing Science 23, 579–595.

Morton, E., 2017. More products lead to more growth in online retailing. URL: `https://www.digitalcommerce360.com/2017/08/29/products-lead-growth-online-retailing/`.

Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., Lin, Y., 2018. Big data analytics in supply chain management: A state-of-the-art literature review. Computers & Operations Research 98, 254–264.

Ni, Y., Fan, F., 2011. A two-stage dynamic sales forecasting model for the fashion retail. Expert Systems with Applications 38, 1529–1536.

Nishimura, N., Sukegawa, N., Takano, Y., Iwanaga, J., 2018. A latent-class model for estimating product-choice probabilities from clickstream data. Information Sciences 429, 406–420.

O'Connor, M., Remus, W., Griggs, K., 1993. Judgemental forecasting in times of change. International Journal of Forecasting 9, 163–172.

O'Donovan, P., Leahy, K., Bruton, K., O'Sullivan, D.T.J., 2015. An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. Journal of Big Data 2, 117.

Perera, H.N., Hurley, J., Fahimnia, B., Reisi, M., 2019. The human factor in supply chain forecasting: A systematic review. European Journal of Operational Research 274, 574–600.

Petropoulos, F., Kourentzes, N., Nikolopoulos, K., Siemsen, E., 2018. Judgmental selection of forecasting models. Journal of Operations Management 60, 34–46.

Qi, Y., Li, C., Deng, H., Cai, M., Qi, Y., Deng, Y., 2019. A deep neural framework for sales forecasting in e-commerce, in: Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E., Carmel, D., He, Q., Xu Yu, J. (Eds.), Proceedings of the 28th ACM International Conference on Information and Knowledge Management, ACM, Beijing, China. pp. 299–308.

Ravi, P.S., 2014. An analysis of a widely used version of the CUSUM tracking signal. Journal of the Operational Research Society 65, 1189–1192.

Reinsel, D., Gantz, J., Rydning, J., 2018. Data age 2025: The digitization of the world from edge to core. URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65.

Rozados, I.V., Tjahjono, B., 2014. Big data analytics in supply chain management: Trends and related research, in: 6th International Conference on Operations and Supply Chain Management, Bali.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting 36, 1181–1191.

Sanders, N.R., 2016. How to use big data to drive your supply chain. California Management Review 58, 26–48.

Sanders, N.R., Ganeshan, R., 2018. Big bata in supply chain management. Production and Operations Management 27, 1745–1748.

Sanders, N.R., Manrodt, K.B., 2003. The efficacy of using judgmental versus quantitative forecasting methods in practice. Omega 31, 511–522.

Schoenherr, T., Speier-Pero, C., 2015. Data science, predictive analytics, and big data in supply chain management: Current state and future potential. Journal of Business Logistics 36, 120–132.

ScrapHero, 2019. How many products does Amazon sell? URL: https://www.scrapehero.com/number-of-products-on-amazon-april-2019/.

Silver, E.A., Pyke, D.F., Peterson, R., 1998. Inventory management and production planning and scheduling, 3rd Edition. Wiley, New York, NY.

Sismeiro, C., Bucklin, R.E., 2004. Modeling purchase behavior at an e-commerce web site: A task-completion approach. Journal of Marketing Research 41, 306–323.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A., 2007. Methodology for long-term prediction of time series. Neurocomputing 70, 2861–2869.

Spiegel, J.R., McKenna, M.T., Lakshman, G.S., Nordstrom, P.G., 2013. Method and system for anticipatory package shipping. URL: https://patents.google.com/patent/US8086546B2/en.

Srinivasan, R., Swink, M., 2018. An investigation of visibility and flexibility as complements to supply chain analytics: An organizational information processing theory perspective. Production and Operations Management 27, 1849–1867.

Steinker, S., Hoberg, K., Thonemann, U.W., 2017. The value of weather information for e-commerce operations. Production and Operations Management 26, 1854–1874.

Syntetos, A.A., Babai, Z., Boylan, J.E., Kolassa, S., Nikolopoulos, K., 2016. Supply chain forecasting: Theory, practice, their gap and the future. European Journal of Operational Research 252, 1–26.

Tan, K.H., Zhan, Y., Ji, G., Ye, F., Chang, C., 2015. Harvesting big data to enhance supply chain innovation capabilities: An analytic infrastructure based on deduction graph. International Journal of Production Economics 165, 223–233.

Tax, D.M., Duin, R.P., 2004. Support vector data description. Machine Learning 54, 45–66.

Thomassey, S., 2010. Sales forecasts in clothing industry: The key success factor of the supply chain management. International Journal of Production Economics 128, 470–483.

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

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

Tiwari, S., Wee, H.M., Daryanto, Y., 2018. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. Computers & Industrial Engineering 115, 319–330.

Trapero, J.R., Kourentzes, N., Fildes, R., 2015. On the identification of sales forecasting models in the presence of promotions. Journal of the Operational Research Society 66, 299–307.

Valentini, G., Dietterich, T.G., 2004. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. Journal of Machine Learning Research 5, 725–775.

Van den Poel, D., Buckinx, W., 2005. Predicting online-purchasing behaviour. European Journal of Operational Research 166, 557–575.

Vapnik, V.N., 1999. The nature of statistical learning theory, 2nd Edition. Springer, New York, NY.

Vassakis, K., Petrakis, E., Kopanakis, I., 2018. Big data analytics: Applications, prospects and challenges, in: Skourletopoulos, G., Mastorakis, G., Mavromoustakis, C.X., Dobre, C., Pallis, E. (Eds.), Mobile Big Data. Springer, Cham, Switzerland, pp. 3–20.

Venkataraman, R.R., Pinto, J.K., 2016. Operations management: Managing global supply chains, 2nd Edition. Sage Publications, Thousand Oaks, CA.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., Baesens, B., 2012. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. European Journal of Operational Research 218, 211–229.

Viet, N.Q., Behdani, B., Bloemhof, J., 2020. Data-driven process redesign: Anticipatory shipping in agro-food supply chains. International Journal of Production Research 58, 1302–1318.

Voccia, S.A., Campbell, A.M., Thomas, B.W., 2019. The same-day delivery problem for online purchases. Transportation Science 53, 167–184.

Waller, M.A., Fawcett, S.E., 2013. Data science, predictive analytics, and big data: A revolution that will transform supply chain design and Management. Journal of Business Logistics 34, 77–84.

Wamba, S.F., Akter, S., Edwards, A., Chopin, G., Gnanzou, D., 2015. How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study. International Journal of Production Economics 165, 234–246.

Wang, G., Gunasekaran, A., Ngai, E.W., Papadopoulos, T., 2016. Big data analytics in logistics and supply chain management: Certain investigations for research and applications. International Journal of Production Economics 176, 98–110.

Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58, 236–244.

Weingarten, J., Spinler, S., 2020a. Shortening delivery times by predicting customers' online purchases: A case study in the fashion industry, in: Bui, T. (Ed.), Proceedings of the 53rd Hawaii International Conference on System Sciences, IEEE, Maui, HI. pp. 1288–1297.

Weingarten, J., Spinler, S., 2020b. The value of clickstream data in product demand forecasting. Unpublished working paper.

Weingarten, J., Spinler, S., 2020c. Using sequential pattern mining to improve demand forecast accuracy. Unpublished working paper.

Weingarten, J., Spinler, S., 2021. Shortening delivery times by predicting customers' online purchases: A case study in the fashion industry. Information Systems Management 38, 287–308.

Xu, L., Duan, J.A., Whinston, A., 2014. Path to purchase: A mutually exciting point process model for online advertising and conversion. Management Science 60, 1392–1412.

Yang, P., Miao, L., Xue, Z., Ye, B., 2015. Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems. Transportation Research Part E: Logistics and Transportation Review 79, 164–177.

Yang, Y., Pan, B., Song, H., 2014. Predicting hotel demand using destination marketing organization's web traffic data. Journal of Travel Research 53, 433–447.

Yeo, J., Kim, S., Koh, E., Hwang, S.W., Lipka, N., 2016. Browsing2purchase: Online customer model for sales forecasting in an e-commerce site, in: Bourdeau, J., Hendler, J.A., Nkambou, R.N., Horrocks, I., Zhao, B.Y. (Eds.), Proceedings of the 25th International Conference Companion on World Wide Web, ACM, Montréal. pp. 133–134.

Zaki, M.J., 2001. SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42, 31–60.

Zalando, 2018. Spoiled for Choice. URL: `https://corporate.zalando.com/en/newsroom/en/stories/spoiled-choice`.

Zhang, G., Eddy Patuwo, B., Y. Hu, M., 1998. Forecasting with artificial neural networks. International Journal of Forecasting 14, 35–62.

Zhang, J., Wang, X., Huang, K., 2016. Integrated on-line scheduling of order batching and delivery under B2C e-commerce. Computers & Industrial Engineering 94, 280–289.

Zhao, Q., Zhang, Y., Friedman, D., Tan, F., 2015. E-commerce recommendation with personalized promotion, in: Werthner, H., Zanker, M., Golbeck, J., Semeraro, G. (Eds.), Proceedings of the 9th ACM Conference on Recommender Systems, ACM, New York, NY. pp. 219–226.

Zotteri, G., Kalchschmidt, M., 2007. Forecasting practices: Empirical evidence and a framework for research. International Journal of Production Economics 108, 84–99.