Data-driven Decision Support for perishable Goods

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Jakob Huber
aus Ludwigshafen am Rhein

Mannheim, 2019

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Prof. Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent: Prof. Dr. Nikolaos Kourentzes, Lancaster University, United Kingdom

Tag der mündlichen Prüfung: 27. September 2019

Abstract

Retailers offering perishable consumer goods such as baked goods have to make hundreds of ordering decisions every day because they typically operate numerous stores and offer a wide range of products. Daily decisions or even intraday decisions are necessary as perishable goods deteriorate quickly and can usually only be sold on one day. Obviously, decision making concerning ordering quantities is a challenging but important task for each retailer as it affects its operational performance. Ordering too little leads to unsatisfied customers while ordering too much leads to discarded goods, which is a major cost factor. In practice, store managers are typically responsible for decisions related to perishable goods, which is not optimal for various reasons. Most importantly, the task is time consuming and some store managers may not have the necessary skills, which results in poor decisions. Hence, our goal is to develop and evaluate methods to support the decision-making process, which is made possible by advances in information technology and data analysis. In particular, we investigate how to exploit large datasets to make better decisions.

For daily ordering decisions, we prose data-driven solution approaches for inventory management models that capture the trade-off of ordering too much or ordering too little such that the profits are maximized. First, we optimize the order quantity for each product independently. Second, we consider demand substitution and jointly optimize the order quantities of substitutable products. For intraday decisions, we formulate a scheduling problem for the optimization of baking plans based on hourly forecasts.

Demand forecasts are an essential input for operational decisions. However, retail forecasting research is mainly devoted to weekly data using statistical time series models or linear regression models, whereas large-scale forecasting on daily data is understudied. We phrase the forecasting problem as a supervised Machine Learning task and conduct a comprehensive empirical evaluation to illustrate the suitability of Machine Learning methods.

We empirically evaluate our solution approaches on real-world datasets from the bakery domain that are enriched with explanatory feature data. We find that our approaches perform competitive to state-of-the-art methods. Data-driven approaches substantially outperform traditional methods if the dataset is large enough. We also find that the benefit of improved forecasting dominates other potential benefits of data-driven solution methods for decision optimization. Overall, we conclude that data-driven decision support for perishable goods is feasible and superior to alternatives that are based on unreasonable assumptions or established time series models.

Zusammenfassung

Einzelhändler, die verderbliche Konsumgüter wie Backwaren anbieten, müssen täglich hunderte Bestellentscheidungen treffen, da sie viele Filialen betreiben und ein breites Sortiment anbieten. Tägliche oder gar untertägige Entscheidungen sind notwendig, da verderbliche Waren in der Regel nur an einem Tag verkauft werden können. Natürlich sind Bestellentscheidungen eine herausfordernde, aber wichtige Aufgabe für jeden Einzelhändler, da sie sich auf das Betriebsergebnis auswirken. Eine zu geringe Bestellmenge führt zu unzufriedenen Kunden, während eine zu hohe Bestellmenge zu überschüssigen Waren führt, die ein wesentlicher Kostenfaktor sind. In der Praxis sind die Filialleiter für die Bestellentscheidungen verderblicher Waren verantwortlich, was aus verschiedenen Gründen nicht optimal ist. Hervorzuheben ist, dass die Aufgabe zeitaufwendig ist und einige Filialleiter nicht über die notwendigen Fähigkeiten verfügen, wodurch schlechte Entscheidungen getroffen werden. Daher ist es unser Ziel, Methoden zur Unterstützung des Entscheidungsprozesses zu entwickeln und zu bewerten, die durch Fortschritte in der Informationstechnologie und Datenanalyse ermöglicht werden. Insbesondere untersuchen wir, wie man große Datensätze nutzen kann, um bessere Entscheidungen zu treffen.

Für tägliche Bestellentscheidungen entwickeln wir datengetriebene Lösungsansätze für Bestandsführungsmodelle, die den Kompromiss zwischen zu viel oder zu wenig zu bestellen erfassen, so dass die Gewinne maximiert werden. Zunächst optimieren wir die Bestellmenge für jedes Produkt unabhängig voneinander. Danach berücksichtigen wir Substitution und optimieren die Bestellmengen von substituierbaren Produkten gemeinsam. Für untertägige Entscheidungen formulieren wir ein Planungsproblem zur Optimierung von Backplänen auf Basis von stündlichen Prognosen.

Bedarfsprognosen sind ein wesentlicher Einflussfaktor für operative Entscheidungen. Die Forschung widmet sich jedoch überwiegend wöchentlichen Daten unter Verwendung statistischer Zeitreihenmodelle oder linearer Regressionsmodelle, während Prognosemethoden auf großen Mengen täglicher Daten nicht untersucht werden. Wir formulieren das Prognoseproblem als überwachte Maschinelle Lernaufgabe und bieten eine Bewertung verschiedener datengetriebener Methoden an.

Wir evaluieren unsere Lösungsansätze empirisch mit echten Datensätzen aus der Bäckereindustrie. Wir stellen fest, dass unsere Ansätze wettbewerbsfähig mit etablierten Methoden sind. Maschinelles Lernen ist wesentlich leistungsfähiger als herkömmliche Methoden, wenn der Datensatz groß genug ist. Wir stellen auch fest, dass der Nutzen einer verbesserten Prognose andere potenzielle Vorteile datengetriebener Lösungsmethoden dominiert. Insgesamt kommen wir zu dem Schluss, dass eine datengesteuerte Entscheidungshilfe für verderbliche Waren möglich und Alternativen überlegen ist, die auf unangemessenen Annahmen oder etablierten Zeitreihenmodellen basieren.

Acknowledgments

When I started my studies in Mannheim, I never dreamed that I would write my dissertation about ten years later. Now, I would like to thank some people who played an important role in making this happen.

The first person I am very grateful and deeply indebted to is Heiner Stuckenschmidt, who is the supervisor of this thesis and the best superior anyone can imagine. During the last years, he gave me lots of freedom, supported my decisions, and also gave me the necessary guidance.

I joined Heiner's group about nine years ago and had the chance to work with many great people. In particular, during my time as a student assistant, I worked for a long time for Christian Meilicke, Jan Noessner, and Mathias Niepert who always gave me interesting tasks that made me repeatedly renewing my contract. Ultimately, I was convinced by Christian to do a PhD and join the group as a research assistant. Additionally, I am deeply grateful to Christian because he helped me to refine my research skills and taught me how to approach research projects.

I am extremely grateful to Alexander Gossmann and Marc Huber, the founders of OPAL - Operational Analytics GmbH, who gave me the chance to do my PhD on such an interesting topic as well as funding my research. It was also a very valuable experience to work on several innovative industry projects and discussing various topics with my colleagues at OPAL, especially Andrej Krakau, Marta Castela and Marius Wirths.

I am also very grateful to Sebastian Müller and Moritz Fleischmann, who played a very important role in connecting machine learning and operations research. Working with them was invaluable as they offered a slightly different view on the research problems that we collaborated on.

Special thanks to Sebastian Kotthoff and Markus Oestringer for enabling the extensive evaluation in this thesis by setting up the necessary infrastructure and immediately resolving any kind of technical issues. Thanks also to Stephanie Keil for handling many administrative barriers and organizational tasks.

I would like to thank my friend Timo Sztyler with whom I studied together and who was always there to discuss anything. While we did not collaborate during our time as PhD students, we participated together as a team in the Hypo University Challenge where we beat all competitors. It was a remarkable experience as well as a remembrance of past times.

Finally, I would like to thank my family for their constant encouragement and support that enabled me to achieve my goals.

Contents

List of Tables

List of Figures

Part I

Motivation & Foundation

1 Introduction

1.1 Research Motivation

Retailers offering perishable fast-moving consumer goods are required to make hundreds of ordering decisions on a daily basis as they typically run numerous stores and offer a broad assortment. Obviously, determining ordering decisions is a challenging but important task for each retailer as it affects its operational performance. In this section, we provide a brief overview of the specific challenges and highlight opportunities for improvement that motivate our research.

Characteristics of perishable Goods. Fast moving consumer goods (FMCG) comprise articles that are sold at a high frequency as they are mostly required to fulfill the daily needs (e.g. food, drinks) (Kaiser, 2011). The group of FMCG also includes goods that have a short shelf-life due to the perishable character of the products, which makes regular ordering necessary. van Donselaar et al. (2006) classify items as perishable goods if they have a high rate of deterioration at ambient storage conditions (e.g. vegetables) or an obsolescence date that makes reordering impractical (e.g. newspapers). They report that perishable items have a 50% higher number of average sales per week and a 40% smaller median case pack size compared to non-perishable items. Thus, they conclude that the time between two orders is 2.5 times smaller for perishable goods, which indicates that they are rather fast moving goods. Perishable goods are typically delivered several times per week (van Donselaar et al., 2006) and can only be sold for at most a few days as the freshness of such products decreases rapidly. Hence, items that are not sold in time are waste and have to be discarded, which is a major cost factor. On the other hand, running out-of-stock (OOS) leads to loss of revenue as the customers cannot buy the item they are looking for. A retailer can increase its revenue and profit by increasing the availability of the articles while limiting the waste. A common problem for retailers related to the ordering of fresh food is that the order quantities are often determined by store managers based on their experience (van Donselaar et al., 2006, 2010).

Effects and Causes of Stock-outs. The effect of ordering too much can be quantified as the unsold items of perishable goods are waste and cannot be sold after the shelf-life expires. From a financial point of view, the retailer loses the costs related to the production and delivery of the unsold items and has to pay waste collection fees or donate them to charity. For articles having a small profit margin, it is important to limit avoidable costs. On the other

hand, ordering too little leads to OOS, which is much harder to quantify as the customer reaction is uncertain. Ehrenthal and Stölzle (2013) consider that an article is OOS if it cannot be bought by a customer at a given point in time. Studies suggest that the global average of OOS is 8.3% (Corsten and Gruen, 2003). OOS leads to an immediate revenue loss of 4% (Gruen et al., 2002) but also affects customer loyalty and jeopardizes future sales (Zinn and Liu, 2008).

The effects of OOS have been widely investigated (Campo et al., 2000; Gruen et al., 2002; Gruen and Corsten, 2007; Helm et al., 2013). Campo et al. (2000) state that customers switch stores, substitute items, postpone the purchase or do not buy anything if the required item is not available. However, the actual response depends on factors like a pre-shopping agenda, urgency of the purchase, brand loyalty and store prices (Zinn and Liu, 2001). Therefore, OOS leads to lost sales, dissatisfied shoppers and diminishes store loyalty. It also obstructs sales planning as the historic sales data is distorted and does not reflect the actual demand. This affects the forecast accuracy and consequently decisions because of demand underestimation of items that were occasionally sold out in the past as well as to demand overestimation due to substitution effects. These effects are not limited to the directly affected article category. Ehrenthal and Stölzle (2013) report that OOS of fresh goods leads to the highest turnover loss compared to other categories. Hence, decreasing OOS is a possibility to increase revenue.

The described effects of OOS underline that a retailer gains a competitive advantage by avoiding OOS. Thus, understanding the causes of OOS is required as it points to issues that need to be improved in order to achieve a better service level. Ehrenthal and Stölzle (2013) report that the causes for OOS in the retail industry are specific to retailer, store, category and item. However, many researchers identified inefficient store operations (Gruen et al., 2002; Gruen and Corsten, 2007; Ehrenthal and Stölzle, 2013) and not issues in the upstream supply chain (e.g. shortage) as the primary cause for OOS (Aastrup and Kotzab, 2010). They also observed that the article availability decreases on the downstream towards the retail shelves. Collaboration and communication between supplier and retailer provoke fewer problems regarding article availability. In an empirical study, Ehrenthal and Stölzle (2013) optimize the flow of goods by simplifying and structuring the tasks for the store personnel and bundling store deliveries and shelf replenishment. After the implementation of these operational changes, OOS was mainly caused by erroneous orders instead of fulfillment and replenishment problems.

Decision Support for perishable Goods. Retailers use automated ordering systems for most items. However, it is often the case that orders for perishable items are based on the experience and judgment of the store manager as the systems are not adapted to perishable goods (van Donselaar et al., 2006) which has several drawbacks:

1. The decision process is not transparent, i.e., store clerks in different stores apply different rules for different article categories, and the decisions may not be as accurate as desired.

2. The manual decision process is often quite time consuming and does not scale. Some store clerks might consider comparable historic data (e.g. sales of previous weeks) as well as various factors like price, promotions, product quality (e.g. based on the origin country) or weather data if such data is available.

3. The skills of the store managers are not consistent across all stores of a retail chain which makes this approach unreliable.

4. Judgmental forecasts are often less accurate and more biased than statistical forecasts (Fildes et al., 2009; Syntetos et al., 2010, 2016). In particular, if they are not carried out by experts.

Hence, the usage of an automated decision support system that implements the respective decision rules of the store managers leads to a competitive advantage for the retailer as it reduces the workload of the personnel. In addition to that, an improved decision quality has also a significant impact on the operational performance (van Woensel et al., 2007; Ehrenthal and Stölzle, 2013).

Decisions are based on the estimation of future demand, i.e., demand forecasts. Thus, the performance of a supply chain also depends on the accuracy of the demand forecasts (Adebanjo and Mann, 2000; Adebanjo, 2009) which is reflected by the fact that supply chain forecasting is an active field of research (Fildes et al., 2008; Syntetos et al., 2016). A study in the fast moving consumer goods sector reveals increased product availability, lower inventory levels along the supply chain and more effective use of current capital assets as major benefits of effective forecasting (Adebanjo and Mann, 2000). van Donselaar et al. (2006, 2010) argue that automated ordering systems should be customized for specific product groups in order to provide more reliable results which motivates research dedicated to decisions support systems for perishable goods.

Advances in Information Technology and Data Analysis. Retailers accumulate very large datasets (e.g. sales history) over the years that can also be enhanced by external information like calendric events (Hofmann and Rutschmann, 2018). Recent advances in information technology and developments in the area of large-scale data analysis provide new opportunities for exploiting the data and optimizing short-term decisions related to perishable goods. The benefits of decision support systems (DSS) (Holsapple and Whinston, 1996; Holsapple and Sena, 2005; Power, 2008) in the context of supply chain management depend on the age of the dependent fact data as its value decreases between the occurrence of the respective business event and the executed action (Hackathorn, 2004; Watson, 2009). Traditional business intelligence (BI) systems (Chaudhuri et al., 2011) are too slow at gathering data that is relevant for short-term day-to-day decisions (Sahay and Ranjan, 2008; Hahn and Packowski, 2015). However, this is necessary in the context of decision support for perishable goods. Traditional BI systems access the data warehouse rather than the operational databases that are optimized for online transaction processing (OLTP) and contain the most recent data.

This separation was necessary as the requirements of online analytic processing (OLAP) (e.g. filtering, aggregation, drill-down, pivoting) are different from OLTP. Hence, an ETL (extract transform load) process is necessary for replicating the data into the data warehouse that is accessible by BI systems. In the last decade, the developments of database technology led to in-memory databases that are capable of efficiently handling OLTP as well as OLAP queries (Plattner, 2009; Sikka et al., 2012) based on fine-granular data. Maintaining the data in-memory allows using data structures that are not suitable for disk-based databases and reduces the latency which makes real-time analytics possible. Due to these advantages, in-memory databases become more popular in supply chain management. The largest class of benefits of real-time BI is related to enhanced operational decisions (Sahay and Ranjan, 2008).

Decision support for perishable goods is an application scenario that can be enhanced with a data-driven DSS. In particular, a DSS that supports a retailer at different organizational levels during the planning process by providing demand forecasts as well as decision recommendations is required in order to standardize and optimize the process. Such a DSS needs to access a satisfying amount of historical as well as most recent point-of-sale (POS) data from the operational database in order to apply techniques and methods like pattern recognition, statistical analysis, regression analysis or predictive modeling. Thereby, POS data needs to be aggregated in real-time to the required temporal and organizational level. Hence, all prediction models are able to access the same operational data that contains near real-time information of the sales. Those requirements are met by state-of-the-art technology.

To summarize, retailers offering perishable goods can expect various benefits from dedicated decision support for such goods (van Donselaar et al., 2006). First, the decision process can be standardized and more transparent. Second, the workload of store managers can be reduced as decision recommendations are provided by a system. Third, by relying on statistical methods, the quality of the decisions can be improved, which may result in fewer stock-outs as well as less discarded goods. DSSs for perishable goods are not yet universally established in the retail industry as they are only enabled by advances in information technology in the last decade that allow the support of day-to-day decisions by exploiting a large data foundation containing the most recent information.

1.2 Problem Description

Our research is motivated by the requirements of retailers offering baked goods. More specifically, we will focus on the case of an industrialized bakery. Baked goods are classified as daily fresh items that are not only daily ordered but also have a high number of sales (van Donselaar et al., 2006). A major cost factor for baked goods are excessive stock levels that lead to marked down or thrown away items as the shelf-life of baked goods is usually not longer than one day. Daily production and ordering is necessary as the freshness of baked goods decreases rapidly, which allows selling only on the day of production. It is even the

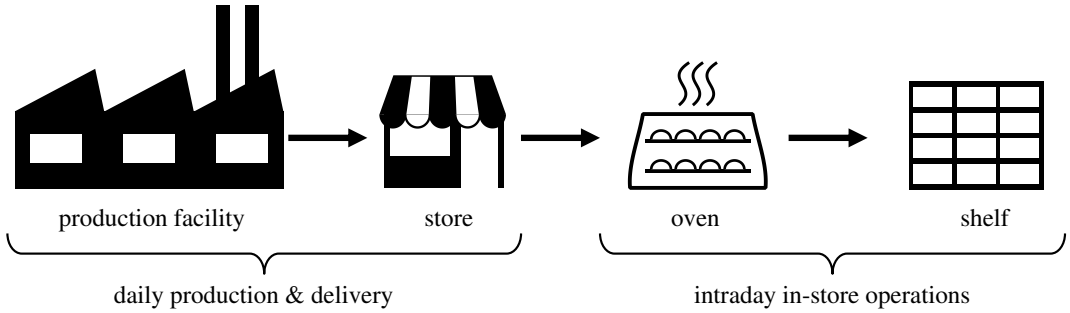


Figure 1.1: The figure depicts the typical supply chain in the bakery domain. The goods are produced in a production facility from which the stores are delivered on daily basis. A part of the assortment needs to be further processed in the stores (e.g. baked) and subsequently placed on the shelves.

case that some baked goods are only delivered in a pre-baked state which means that they need to be baked during the day in the stores. The typical supply chain in the bakery domain is depicted in Figure 1.1. This thesis is primarily concerned with the development and performance assessment of solution approaches concerning the following short-term operational decisions:

Daily order quantity. The order or production quantity of each article needs to be determined on a daily basis. The lead time for baked goods is only one day, which means that short-term demand estimations are required.

Intraday baking / shelf replenishment. Some goods are not ready for sale when they arrive at the store and need to be baked during the day and consequently placed on the shelves. Hence, a baking plan, which aligns the baking process with the customer demand, needs to be provided to the store clerks.

The characteristics of the supply chain and the considered retailers allow to emphasize the decisions (e.g. order quantities) as the primary influence on the performance. The supply chain is short and quite agile as all important parts of it are operated by the company, i.e., production, distribution and stores. This reduces the barriers of collaboration between different parts of the supply chain. For instance, trust is not an issue and unfiltered access to real-time demand information is given to all parties of the supply chain using state-of-the-art information technology in order to make the demand visible. With respect to baked goods, we do also not face the issue of inaccurate inventory levels as the perishability of the goods does not allow to keep inventory (Holweg et al., 2005), i.e., the goods are sold or discarded on the day of production. Hence, challenges like the bullwhip effect (Lee et al., 1997) are very unlikely and can be ignored.

It is also noteworthy that the final products (e.g. buns, breads) are highly perishable while this is not entirely true for the raw materials (e.g. wheat flour, sugar, salt) of baked goods that can be persisted for longer periods. The fact that the lead time of baked goods is only one day enables decision recommendations based on most recent demand observations.

The aforementioned characteristics underline that it is actually possible to execute short-term decision recommendations. Moreover, retailers typically operate numerous stores that offer a comparable assortment within a geographically restricted region and generate a large data foundation.

1.3 Research Questions

This thesis is concerned with data-driven decision support for perishable goods in the retail industry as described in the previous section. Thereby, we consider the complete process from data to decision. Hence, the central question of our research is:

RQ1 How can available data be leveraged in order to support and optimize operational decisions in the present application scenario?

The question is generally relevant from a practical point of view as retailers accumulate large datasets while it is fairly unknown how the data can be exploited to enhance operations and how data usage affects the performance. In order to contribute to the wide-ranging research question RQ1, we address the following more specific research questions:

RQ2 Are data-driven methods for inventory management of perishable goods a viable alternative to model-based approaches?

RQ3 Can the typically separated phases of estimation and optimization be integrated into a single optimization problem?

The choice of order quantities is ultimately an inventory decision. The literature on inventory management mainly assumes specific distributions, which is problematic as neither the type of the demand distribution nor its parameters are known in real-world applications. Hence, we investigate if it is possible to discard such likely imprecise assumptions and rely on large datasets that enable the use of empirical distributions. In this context, we also examine if the integration of the typically separated steps of demand estimation and decision optimization is feasible and reasonable.

RQ4 Has the forecast accuracy of a prediction model a noticeable influence on the operational performance?

While (point) forecasts are an essential input for operational decisions, they do hardly reflect them. For example, it makes sense to add safety stock for daily order quantities in order to increase the service level. Another example are the intraday decisions that are reflected in baking schedules, which take not only the demand estimations but also other restrictions like the availability of the ovens into account. Consequently, it is possible that the effect of more accurate predictions diminishes in a succeeding optimization step. Moreover, aside from established forecast accuracy measures, we also assess the performance based on other key figures like costs or achieved service level.

RQ5 Are Machine Learning methods suitable for retail demand forecasting? What factors affect the performance of Machine Learning methods?

Business forecasting literature is dominated by statistical time series forecasting models and the results of comparative studies (e.g. Ahmed et al. (2010); Makridakis et al. (2018a)) indicate that pure Machine Learning methods are no viable alternative. However, the characteristics of our application scenario (e.g. amount of data) should benefit Machine Learning methods compared to (univariate) time series models. We apply Machine Learning methods and identify factors influencing the performance with respect to data availability.

1.4 Contributions & Published Work

By answering the research questions outlined in the previous section, we contribute in many ways to the state of business forecasting, more precisely retail forecasting, and inventory management of perishable goods:

- We present the requirements of decision support for perishable goods in the bakery industry and propose solution approaches that cover the whole process from data to operational decisions. We also empirically evaluate the proposed solution approaches on real-world datasets.
- We analyze Machine Learning models for large-scale demand forecasting on daily retail data of fast moving goods.
  - We show that Machine Learning methods are a viable alternative to statistical time series models.
  - We analyze factors that influence the performance of Machine Learning methods.
  - We show that it can be beneficial to model forecasting as a classification problem rather than a regression problem.
- We connect Machine Learning and Operations Research.
  - We incorporate Machine Learning models and inventory management models for perishable goods. In this regard, we propose data-driven integrated estimation and optimization solution approaches for the single-product newsvendor problem as well as the multi-product newsvendor problem with substitution.
  - We show that the initial forecast accuracy has a significant influence on the operational performance.
  - We show that data-driven approaches outperform their model-based counterparts.

The thesis is based on articles that are already published in international journals and a couple of working papers. I have been a major contributor to all of the following papers:

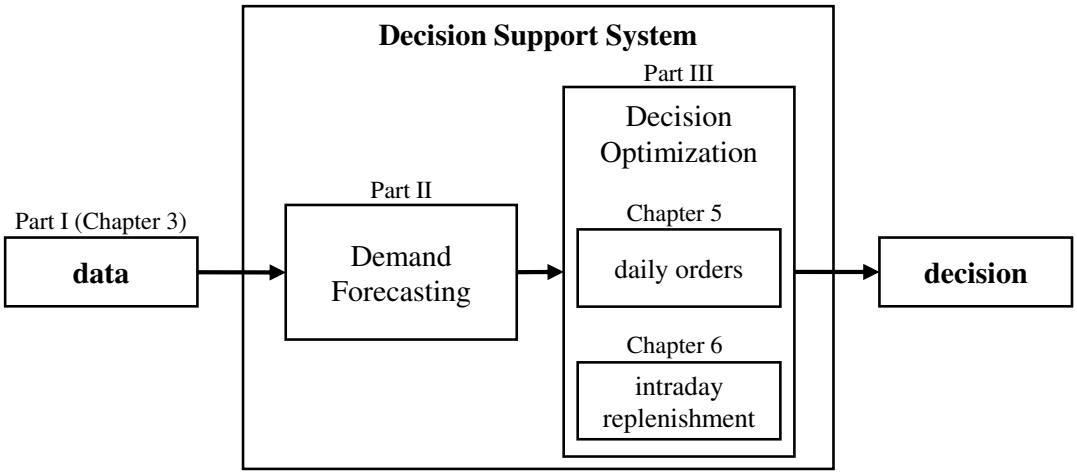


Figure 1.2: Overview of the Decision Support System.

- J. Huber, A. Gossmann, and H. Stuckenschmidt. Cluster-based hierarchical demand forecasting for perishable goods. *Expert Systems with Applications*, 76:140–151, 2017.
- J. Huber, S. Müller, M. Fleischmann, and H. Stuckenschmidt. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278(3): 904–915, 2019.
- J. Huber and H. Stuckenschmidt. Daily Retail Demand Forecasting using Machine Learning with Emphasis on Calendric Special Days. Submitted to *International Journal of Forecasting (under review)*, 2018-2019.
- J. Huber and H. Stuckenschmidt. Intraday Shelf Replenishment Decision Support for perishable Goods. *Working Paper (unpublished)*, 2019.
- S. Müller, J. Huber, M. Fleischmann, and H. Stuckenschmidt. Data-driven Inventory Management under Customer Substitution. *Working Paper (unpublished)*, 2019.

1.5 Outline

While we formulated the research questions starting from the decisions, the structure of this thesis follows the computational process from data to decision as depicted in Figure 1.2. The thesis is divided into four main parts:

Part I: Motivation & Foundation

The first part provides the motivation and foundations of our research concerning *Data-driven Decision Support for perishable Goods*. In *Chapter 1: Introduction* (this chapter), we provide a description of the application scenario that is the subject of our research. Additionally, we briefly discuss the challenges and opportunities for improvement that motivate our work and

lead to the accompanying research questions. *Chapter 2: Preliminaries* contains a description of the theoretical and methodical foundations that are the basis of our solution approaches. This predominantly includes time series forecasting and Machine Learning. In addition, evaluation criteria and evaluation schemes, which we use for the empirical evaluations, are presented and discussed. In *Chapter 3: Data Foundation & Characteristics*, we give an overview on the types of data that are typically available to retailers in this application domain and can be used for operational decision support. We also introduce the datasets that are used for the empirical evaluation of our solution approaches and conduct a brief explorative data analysis in order to illustrate prevalent data characteristics.

Part II: Forecasting

We dedicate a self-contained part to demand forecasting (estimation) as this is a crucial input for operational decisions. To this end, we propose approaches to formulate forecasting as a supervised Machine Learning task and exploit the characteristics of the large-scale demand forecasting scenario (see *Chapter 4: Large-scale Forecasting*). This includes a transformation of the regression problem to a classification problem as well as an identification of levels of data usage. We conduct a comprehensive empirical evaluation in order to illustrate the viability of Machine Learning.

Part III: Decision Support

After the consideration of data-driven approaches for demand estimation in *Part II: Forecasting*, we focus on the optimization of daily decisions and intraday decisions as outlined in the problem description. Consequently, we develop solution approaches for daily decisions in *Chapter 5: Daily Decision Support*. Our methods are based on variants of the newsvendor model, which is an inventory management model for perishable goods. In *Section 5.3: Single-Product Newsvendor*, we study how data can be exploited for decision optimization while the daily order quantity of each product is optimized independently. In *Section 5.4: Multi-Product Newsvendor with Substitution*, we propose and analyze methods for the joint optimization of order quantities of substitutable products. This is reasonable as high substitution rates are observed for baked goods in the event of shortages (e.g. van Woensel et al. (2007)). *Chapter 6: Intraday Decision Support* targets the last step of the considered bakery supply chain that is relevant for a subset of the assortment. Therefore, we develop a method for the generation of baking plans based on daily and intraday demand forecasts.

Part IV: Wrap-up

We conclude the thesis in *Chapter 7: Conclusions* by revisiting our central research questions (see Section 1.3) and summarizing the key results of the presented research. We also briefly discuss practical and managerial implications of our work. Finally, we outline promising future research directions that extend or enhance our research.

2 Preliminaries

In this chapter, we provide a description of the theoretical and methodical foundations that are the basis of our solution approaches. This includes time series forecasting (see Section 2.1) and Machine Learning (ML) (see Section 2.2). In addition, we present and discuss evaluation criteria and evaluation schemes that we use for the empirical evaluation (see Section 2.3).

2.1 Time Series Forecasting

Retailers make many operational decisions based on forecasts that are calculated by time series forecasting methods (see Section 4.2 and Part III). Generally, time series forecasting is concerned with the prediction of the next values, or even the distribution of the next values, of a sequence of uniformly spaced time instants $Y = (y_1, \ldots, y_n)$ with $y_t \in \mathbb{R}$ (Hyndman and Athanasopoulos, 2014). We denote the prediction with $\hat{y}_{t+h}$ with h being the forecasting step.

2.1.1 Methods

We consider several standard benchmark methods as well as more sophisticated statistical time series models. We focus on methods that explicitly handle seasonal data as the demand of the considered products is subject to a strong weekly seasonality (see Section 3.3). The review of De Gooijer and Hyndman (2006) implies that the most popular traditional approaches are exponential smoothing models (Gardner, 1985, 2006) and auto regressive integrated moving average (ARIMA) models (Box and Jenkins, 1976).

2.1.1.1 Baseline Methods

The considered simple baseline methods are *S-Naïve*, *S-MA* and *S-Median*. Those variants are not only standard benchmarks for time series forecasting but also common in the bakery industry as they are easily understandable and cover the prevalent weekly seasonality. The length of the seasonality is specified by m, e.g., $m = 7$ for daily data (Monday - Sunday).

Seasonal-Naïve. The forecast is set to the last observed value from the same part of the season:

$$S\text{-}Naïve : \hat{y}_{t+h} = y_{t+h-m} \tag{2.1}$$

Seasonal Moving Average. The seasonal moving average method (*S-MA*) sets the forecast to an average of the last observations from the same part of the season which is an advantage compared to *S-Naïve* that relies on a single observation. However, we need to set k, which controls the number of considered values:

$$S\text{-}MA : \hat{y}_{t+h} = \frac{1}{k}\sum_{i=1}^{k} y_{t+h-mi} \tag{2.2}$$

Seasonal-Median. The method is an alternative to *S-MA* as it employs a rolling median instead of a rolling average, which makes this method more robust with respect to outliers:

$$S\text{-}Median : \hat{y}_{t+h} = \text{median}(\{y_{t+h-lm} \,|\, l \in \{1, \ldots, k\}\}) \tag{2.3}$$

2.1.1.2 Statistical Time Series Models

With respect to statistical methods, we rely on exponential smoothing models and autoregressive integrated moving average (ARIMA) models. Both model families have shown to perform reasonably well in comparative studies (e.g. Makridakis et al. (2018a)) and are also well-suited for seasonal data.

Seasonal Autoregressive Integrated Moving Average. Autoregressive integrated moving average (ARIMA) and its seasonal variant *S-ARIMA* represent a widely used forecasting method. The autoregressive part (AR) of ARIMA represents a linear combination of past values, while the moving average part (D) is a linear combination of past forecast errors. The time series must be stationary, which can be achieved by differencing (I). The seasonal model can be specified as ARIMA$(p, d, q, P, D, Q)m$. Here, p (P) represent the order of the non-seasonal (seasonal) auto-regressive part, d (D) are the orders of non-seasonal (seasonal) differencing and q (Q) are the order of the non-seasonal (seasonal) moving average part. Moreover, m states the periodicity of the time series. The seasonal ARIMA model is defined as follows:

$$\phi(B)\Phi(B^m)(1-B)^d(1-B^m)^D y_t = \mu + \theta(B)\Theta(B^m)e_t \tag{2.4}$$

The operator B is the backshift operator, i.e., By_t = y_{t-1}, μ is a constant and e_t are error terms. The auto-regressive ($\phi(B)$, $\Phi(B^m)$) and moving average ($\theta(B)$, $\Theta(B^m)$) parts are expressed as polynomials.

$$\phi(B) = 1 - \phi_1 B^1 - \phi_2 B^2 - \ldots - \phi_p B^p \tag{2.5}$$

$$\Phi(B^m) = 1 - \Phi_1 B^{1m} - \Phi_2 B^{2m} - \ldots - \Phi_P B^{Pm} \tag{2.6}$$

$$\theta(B) = 1 - \theta_1 B^1 - \theta_2 B^2 - \ldots - \theta_q B^q \tag{2.7}$$

$$\Theta(B^m) = 1 - \Theta_1 B^{1m} - \Theta_2 B^{2m} - \ldots - \Theta_Q B^{Qm} \tag{2.8}$$

Setting the parameters (p, d, q, P, D, Q, m) properly requires statistical knowledge and depends on the nature of the data. Hyndman and Khandakar (2008) and Rojas et al. (2008) propose methods that automatically determine the parameters. We rely on an implementation of a method developed by Hyndman and Khandakar (2008) that automatically selects the ARIMA model having the lowest Akaike Information Criterion (AIC) (Akaike, 1974). AIC is a measure for describing the relative quality of a statistical model for a dataset by estimating the information loss and considering the complexity of the model. In order to select the parameters, they use a step-wise approach and traverse the space of possible models in an efficient way until the optimal model is found. In our experiments, we employ the `auto.arima()` function from the `forecast` package (Hyndman and Khandakar, 2008) for the statistical software R (R Core Team, 2017) that implements the approach.

Exponential Smoothing. Exponential smoothing methods calculate the forecast by computing a weighted average of past observations. The weights decay as the observations get older. Hyndman et al. (2002, 2008) propose innovation space models that generalize exponential smoothing methods (*ETS*). The model family comprises 30 models that cover different types of errors, seasonal effects and trends (e.g. none, additive, multiplicative). We use the `ets()` function from the `forecast` package (Hyndman and Khandakar, 2008) for the statistical software R (R Core Team, 2017) that selects the model with the lowest AIC. Exemplary, the model *ETS(ANA)* (additive error term, no trend, additive seasonality) can be used for many time series in our application scenario. The model can be written in the error correction form that consists of terms for the time series y_t, the level l_t and the seasonality s_t. The frequency of the seasonality is controlled by m. The smoothing parameters α and γ control the effects of the errors e_t.

$$y_t = l_{t-1} + s_{t-m} + e_t \tag{2.9}$$

$$l_t = l_{t-1} + \alpha e_t \tag{2.10}$$

$$s_t = s_{t-m} + \gamma e_t \tag{2.11}$$

2.1.2 Forecast Uncertainty

The predictions computed by the methods introduced in the previous section are point forecasts. The forecasts will as frequently underestimate as overestimate the actual demand if the models are correctly specified and unbiased. With respect to decision optimization, it is necessary to consider the uncertainty and the probability distribution related to a forecast. In forecasting literature, the uncertainty of a forecast is considered by density forecasts (Tay and Wallis, 2000) or expressed by prediction intervals (Chatfield, 2001). Prediction intervals describe the range into which a prediction falls with a prescribed probability, while density forecasts provide the probability distribution of the future value. Chatfield (2001) provides

an overview of approaches for calculating prediction intervals. The approaches are generally based on forecast errors:

$$e_{t+h} = y_{t+h} - \hat{y}_{t+h} \tag{2.12}$$

Thereby, it is important to distinguish between in-sample errors, i.e, errors of one-step predictions in the training set, and out-of-sample forecast errors. The in-sample-errors, which are in fact the difference between the model fit and the actual values, are called residuals. They allow drawing conclusions about the uncertainty of the predicted values. However, they are only comparable with one-step out-of-sample forecast errors as errors accumulate as the forecast horizon increases. The prediction intervals can be calculated with a model dependent formula that typically assumes a standard normal distribution of the forecast errors (Hyndman and Athanasopoulos, 2014). Thereby, the uncertainty related to the model selection, model specification, and parameter estimation is not considered. This often leads to too narrow prediction intervals and unsatisfactory results (Chatfield, 2001; Lam and Veall, 2002; De Gooijer and Hyndman, 2006). An alternative are re-sampling methods based on empirical errors, which are more computationally intensive but can provide more accurate results (Chatfield, 2001). Fildes et al. (2008) argue that empirical estimates of quantiles are more accurate and that out-of-sample errors are the best estimate for uncertainty. Alternatively, quantile regression allows the direct prediction of a specific quantile of the distribution (Koenker, 2005).

2.1.3 Hierarchical Forecasting

Hierarchical forecasting (Gross and Sohl, 1990) can be applied to exploit the structure of time series data. The main approaches are top-down and bottom-up forecasting. The top-down approach requires forecasting at an aggregate level and allocating the forecasts to the lower level time series (derived forecasts). The bottom-up approach requires forecasting at a lower level and summing the forecasts to obtain the aggregate level forecasts (cumulative forecasts). The applied approach depends on the objective of the forecast. According to Kahn (1998), top-down forecasting is preferred for strategical planning (e.g. budgets) while bottom-up forecasting is preferred for tactical forecasting where detailed forecasts are required (e.g. production and distribution).

2.2 Machine Learning

Machine Learning (ML) methods are by definition able to learn patterns from data without imposing many requirements on the data generating process (Hastie et al., 2009). They are able to learn patterns from data by exploiting large datasets, which makes them suitable for big data applications. Supervised ML distinguishes between *regression* and *classification*. The difference is that the target values of a *regression* problem are quantitative, while the target values of a *classification* task are qualitative. In the following, we focus on the problem formulation for *regression* tasks as this is the natural problem type for time series forecasting. However, the general concepts are identical for all supervised ML tasks.

The general goal is to approximate a function $f(\cdot)$ that models the relation between a vector (or matrix) of quantitative input variables, called *features*, $X \in \mathbb{R}^p$ and a quantitative output variable, called *target*, $Y \in \mathbb{R}$ (Hastie et al., 2009).

$$\hat{Y} = f(X) \tag{2.13}$$

The predicted value is denoted by $\hat{Y}$ while the observed value is Y. In order to employ a ML algorithm, we need to provide a (training) dataset $\mathcal{D} = \{(x_k, y_k)\}_{k=1..n}$ consisting of a set of tuples containing features $x_k \in \mathbb{R}^m$ that describe a target $y_k \in \mathbb{R}$. $\mathcal{D}$ can also be expressed as a pair $(\mathbf{X}, \mathbf{Y})$ of a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and a target vector $\mathbf{Y} \in \mathbb{R}^n$.

Based on the observed training data, a ML algorithm approximates a function $f(\cdot)$ by minimizing a loss function $\mathcal{L}(Y, f(x))$ that assesses the fit of $f(\cdot)$. A loss function $\mathcal{L}$ is typically a globally continuous and differentiable function, e.g., the squared loss (L2-norm) can be used for regression tasks:

$$\mathcal{L}(Y, f(X)) = (Y - f(X))^2 \tag{2.14}$$

For classification problems, the cross-entropy loss $-\sum_{k=1}^{K} g_k \log(f(x)^{(k)})$ is typically used as loss function where g_k is a binary indicator for class k and $f(x)^{(k)}$ the predicted probability for class k of the model $f(\cdot)$ given a feature vector x.

The parameters θ of $f(\cdot)$ need to be tuned in order to minimize the loss function $\mathcal{L}$. A common problem is that the approximated function $f(\cdot)$ has a much lower error on the training dataset $\mathcal{D}^{train}$ than on some unseen test dataset $\mathcal{D}^{test}$, i.e., it does not generalize well. This phenomenon is called overfitting. In order to control and prevent that, a validation dataset $\mathcal{D}^{valid}$ is usually retained, which can be used to test if the model does overfit the training dataset.

2.2.1 Time Series Forecasting

Besides traditional methods (see Section 2.1), data-driven approaches like artificial neural networks (ANNs) are also considered for time series forecasting (Zhang et al., 1998; Zhang, 2012). We extend the introduced notion of time series data and forecasting and adapt it to ML. Time series forecasting can be framed as a supervised ML task. In Section 2.1, we introduced a univariate time series $Y = (y_1, \ldots, y_n)$ with $y_t \in \mathbb{R}$. Additionally, each data point y_t of a time series Y can be enriched by explanatory variables $X = (x_1, \ldots, x_n)$, $x_t \in \mathbb{R}^p$ comprising information that is not contained in the original time series but can be exploited to understand and model the apparent patterns in Y. Time series data along with the explanatory information can be transformed to feature and target pairs that can be processed by a ML algorithm. For instance, a time series $(y_1, \ldots, y_n)$ representing an autoregressive $AR(a)$ process can be formulated as follows (Adya and Collopy, 1998; Zhang et al., 1998):

$$\mathbf{X} = \begin{bmatrix} y_1 & y_2 & \cdots & y_a \\ \vdots & \vdots & \vdots & \vdots \\ y_{t-a} & y_{t-a+1} & \cdots & y_{t-1} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n-a} & y_{n-a+1} & \cdots & y_{n-1} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_{a+1} \\ \vdots \\ y_t \\ \vdots \\ y_n \end{bmatrix} \tag{2.15}$$

Hence, the lagged time series observations build the feature matrix $\mathbf{X}$. For the present application scenario, each time series is enhanced with explanatory information. Therefore, $\mathbf{X}$ can be extended with additional feature data:

$$\mathbf{X} = \begin{bmatrix} y_1 & y_2 & \cdots & y_a & x_{a+1,1} & \cdots & x_{a+1,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-a} & y_{t-a+1} & \cdots & y_{t-1} & x_{t,1} & \cdots & x_{t,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-a} & y_{n-a+1} & \cdots & y_{n-1} & x_{n,1} & \cdots & x_{n,p} \end{bmatrix} \tag{2.16}$$

The present modeling is a multiple regression as a single target variable depends on multiple variables covering autoregressive and external information. A ML method can be employed to approximate the functional relation in a data-driven fashion. Thereby, it does not distinguish between the origin of the feature variables, i.e., autoregressive or external, as the semantics are hidden to the method.

2.2.2 Methods

In this section, we briefly introduce ML methods for function (2.13) that we use in this thesis. In particular, we consider ANNs and gradient boosted regression trees. Those types of models have in common that they rely on a gradient-based approach to optimize the model parameters θ. However, all methods process data in a different way as we outline below.

2.2.2.1 Artificial Neural Networks

ANNs are data-driven models that can approximate any continuous function (Hornik, 1991), making them suitable for forecasting if it is difficult to specify the underlying data generation process. They are mathematical models that are inspired by biological brains and consist of nodes that are connected by weighted edges, which are represented by matrices and vectors. We consider two distinguishable types of ANNs: feed-forward ANNs and recurrent ANNs.

Feed-forward Neural Networks. Feed-forward Neural Networks (FNNs), e.g., a multilayer perceptron (MLP), are the most popular neural network architecture for time series forecasting over the last decades (Zhang et al., 1998). In a FNN having L hidden layers ($L \geq 1$), the output of each layer $h^{(k)}(x)$ gets passed to the next layer ($1 \leq k \leq L+1$):

$$h^{(k)}(x) = \sigma^{(k)}(b^{(k)} + W^{(k)}h^{(k-1)}(x)) \tag{2.17}$$

The output of the input layer is defined as $h^0(x) = x$ while the output of the last layer represents the prediction of the network, i.e., $f(x) = h^{(L+1)}(x)$. The output of each layer is connected with a fully connected weight matrix $W^{(k)}$ to the next layer. The input of a layer gets adjusted with the biases $b^{(k)}$ of each neuron before it passes an activation function $\sigma^{(k)}$.

Recurrent Neural Networks. Recurrent Neural Networks (RNNs) process the input features in sequential order and apply the same network to each step in a sequence. RNNs maintain an internal memory that allows them to track dynamic patterns. We use a variant of RNNs, called long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), which has a sophisticated memory concept based on input gates i_t, output gates o_t, forget gates f_t, and a cell state c_t that allows tracking dynamic patterns:

$$f_t = \sigma_{sigmoid}(W_f x_t + U_f h_{t-1} + b_f) \tag{2.18}$$

$$i_t = \sigma_{sigmoid}(W_i x_t + U_i h_{t-1} + b_i) \tag{2.19}$$

$$o_t = \sigma_{sigmoid}(W_o x_t + U_o h_{t-1} + b_o) \tag{2.20}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_{tanh}(W_c x_t + U_c h_{t-1} + b_c) \tag{2.21}$$

$$h_t = o_t \circ \sigma_{tanh}(c_t) \tag{2.22}$$

The operator $\circ$ is the Hadamard product, i.e., element-wise multiplication of matrices and vectors having the same dimension. The parameters of an LSTM unit are W, U, and b.

For both types of ANNs, the input of a node in the network gets passed through an activation function. Thus, we list frequently used activation functions:

- logistic function: $\sigma_{sigmoid}(x) = \frac{1}{1+e^{-x}}$
- hyperbolic function: $\sigma_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- rectified linear activation: $\sigma_{relu}(x) = x^+ = \max(0, x)$
- exponential linear activation: $\sigma_{elu}(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^x - 1 & \text{otherwise} \end{cases}$
- linear function: $\sigma_{linear}(x) = x$
- softmax function: $\sigma_{softmax}(x) = [\frac{exp(x_1)}{\sum_c exp(x_c)} \cdots \frac{exp(x_C)}{\sum_c exp(x_c)}]$

While the activation functions of an LSTM cell are specified, this is not the case for FNNs/MLPs, which requires selecting the activation functions during the model building process. In principle, all activation functions can be used at any layer. For time series forecasting, $\sigma_{linear}(x)$ is often used at the output layer, but it is also possible to use $\sigma_{relu}(x)$ to avoid negative predictions. The activation function $\sigma_{softmax}(x)$ is used for classification and normalizes the inputs to a probability distribution over the target classes.

The parameters (weights) of an ANN can be trained with a stochastic gradient-based algorithm. For this purpose, we use the stochastic gradient-based algorithm ADAM proposed by Kingma and Ba (2015). The performance of an ANN also depends on its initial weights, which are randomly set. In order to reduce the variance and to obtain more robust results, we employ an ensemble of ANNs with the *median* ensemble operator, as this approach is robust to the initial weights and provides reliable results (Barrow et al., 2010; Kourentzes et al., 2014).

Bergmeir et al. (2018) show that k-fold cross-validation is suitable to control overfitting when ML methods are employed. We employ k-fold cross-validation (Barrow and Crone, 2016; Bergmeir et al., 2018) on the training dataset in order to validate and design the models as the ordering of the observations does not have to be preserved.

ANNs are also able to deal with seasonal time series data. In order to encode a deterministic seasonality, we use trigonometric functions as features, as proposed by Crone and Kourentzes (2009). This is a parsimonious approach that requires only two additional input variables per seasonality. Additionally, the approach is non-parametric as no seasonal indices need to be estimated. The two variables are $x_{i,1}$ and $x_{i,2}$ in period i, with m representing the frequency of the seasonality:

$$x_{i,1} = sin(2\pi i/m) \tag{2.23}$$

$$x_{i,2} = cos(2\pi i/m) \tag{2.24}$$

2.2.2.2 Gradient Boosted Regression Trees

Decision trees (DTs) are simple binary trees that map an input to the corresponding leaf node. Since the introduction of classification and regression trees, several approaches have been developed that combine multiple DTs for one prediction (e.g. random forest (Breiman, 2001)). Gradient boosted regression trees have gained much interest in recent years and are an alternative to ANNs for structured data. The most popular implementations are *LightGBM* (Ke et al., 2017a) and *xgboost* (Chen and Guestrin, 2016). Like any boosting algorithm, they train a series of simple models $f_k(x)$ (i.e. decision trees) based on accumulated residuals of the previous model $\mathcal{L}^{(k)}(y, \hat{y}^{(k-1)} + f_k(x))$. Hence, the prediction is the sum of all trained simple models $f_k(x)$, i.e., $f(x) = \sum_{k=1}^{K} f_k(x)$.

2.3 Performance Assessment

In this section, we outline and discuss performance measures and evaluation schemes that are used to assess the prediction and decision quality of our solution approaches.

2.3.1 Evaluation Criteria

We rely on established forecast accuracy measures (see Section 2.3.1.1) and additional criteria that are more suitable indicators for the assessment of the operational performance (see

Section 2.3.1.2). We define all measures based on target values $y_1, \ldots, y_N$ and predictions $\hat{y}_1, \ldots, \hat{y}_N$. We can assume that target values (y_n) and predictions ($\hat{y}_n$) are larger than zero as we focus on fast moving goods.

We also test if the performance differences are statistically significant. For this purpose, we employ the Wilcoxon signed-rank test (Wilcoxon, 1945) to determine if there are statistically significant differences among the evaluated methods at 0.05 significance level. It is a rank-based test that does not require assumptions on the distributions of the key figures. If not stated otherwise, we underline the best performance for each metric and print results that that do not differ from the one of the best method at a significance level of 5% in bold face.

2.3.1.1 Forecast Accuracy

A variety of forecast accuracy measures have been proposed that have different strengths and weaknesses (Hyndman and Koehler, 2006; Hyndman and Athanasopoulos, 2014). The different error measures can be grouped into scale-dependent errors, percentage errors and scaled errors.

Scale-dependent errors include the mean absolute error (MAE) and the root mean square error (RMSE). They are typically used to compare results on datasets having the same units and comparable scales. In general, minimizing the MAE leads to the median while minimizing the RMSE yields the mean (Gneiting, 2011) of the distribution, which is often desired. RMSE penalizes larger errors more than smaller errors in contrast to MAE. An advantage is that both measures are always defined and rather easy to interpret.

$$MAE = \frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n| \tag{2.25}$$

$$RMSE = \sqrt{\frac{1}{N} \sum\nolimits_{n=1}^{N} (y_n - \hat{y}_n)^2} \tag{2.26}$$

Percentage errors include the mean absolute percentage error (MAPE) and the symmetric mean absolute percentage error (SMAPE). The measures allow the comparison of time series at different scales but are not defined if the target value and/or the prediction is zero. However, this is not an issue with respect to our research as the considered items are sold in rather high quantities every day. Additionally, the mean percentage error (MPE) can be used as a bias indicator as negative and positive errors offset each other.

$$MAPE = 100 \cdot \frac{1}{N} \sum_{n=1}^{N} \frac{|y_n - \hat{y}_n|}{y_n} \tag{2.27}$$

$$SMAPE = 100 \cdot \frac{1}{N} \sum_{n=1}^{N} \frac{|y_n - \hat{y}_n|}{(y_n + \hat{y}_n)/2} \tag{2.28}$$

$$MPE = 100 \cdot \frac{1}{N} \sum_{n=1}^{N} \frac{y_n - \hat{y}_n}{y_n} \tag{2.29}$$

Scaled errors like the mean absolute scaled error (MASE) (Hyndman and Koehler, 2006) have been introduced as an alternative to percentage-based errors when comparing forecasts on datasets having different scales or units. The absolute forecast errors are scaled based on the error of a simple forecast method, i.e., in our case the seasonal naïve forecast, on the training data. The scaled error is smaller than one if the error of the evaluated method is smaller than the error of the simple reference method on the training data, e.g., the seasonal naïve forecast $\hat{y}_t = y_{t-m}$ for ordered observations y_t with seasonality m (e.g. $m = 7$ for the weekly seasonality of daily data).

$$MASE = \frac{1}{N} \sum_{n=1}^{N} \frac{|y_n - \hat{y}_n|}{\frac{T}{T-m} \sum_{t=m+1}^{T} |y_t - y_{t-m}|} \tag{2.30}$$

In this research, we address the problem of decision support for perishable goods. Thus, the unit of the data is typically the number of items of a specific product which enables the use of MAE and RMSE. However, as we compute the key figures across different stores, we also compute percentage-based errors and scaled errors as the scale of the demand can differ significantly. By considering different types of error measures, it is less likely that the results are mostly influenced by a small subset of the evaluated time series which makes the results more robust. While other evaluation criteria are available, we selected the aforementioned criteria as they are widely used and easy to interpret. For instance, SMAPE and MASE were used for the evaluation of point forecasts in the recent M4 Forecasting Competition (Makridakis et al., 2018b). Kolassa (2016) highlights that MAE and MASE are not suitable measures for count data and proposes to evaluate the entire predictive distribution rather than single functionals (e.g. mean). However, the study focuses on intermittent demand data instead of fast moving goods that are sold in higher volumes.

2.3.1.2 Operational Performance

In addition to the point forecast accuracy measures, we consider performance metrics that are reasonable indicators for the operational performance of the supply chain. In order to satisfy the customers and to generate revenue, it is important to fulfill a large portion of the demand. Hence, we consider the fill rate (FR) and the service level (SL).

$$\mathrm{FR} = 1 - \frac{1}{N} \sum_{n=1}^{N} \frac{(y_n - \hat{y}_n)^+}{y_n} \tag{2.31}$$

$$\mathrm{SL} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(y_n \leq \hat{y}_n) \tag{2.32}$$

The fill rate is a quantity-based service level as it measures the proportion of the demand that can be fulfilled. The service level (SL) is event-based and indicates the probability that the complete demand can be served. The expected service level for unbiased point forecasts is 50%. A higher fill rate and a higher service level is linked to a higher number of items that have to be discarded. Thus, we also measure the loss (overage) rate (LR).

$$\mathrm{LR} = \frac{1}{N} \sum_{n=1}^{N} \frac{(\hat{y}_n - y_n)^+}{y_n} \tag{2.33}$$

The overage rate indicates the percentage of goods that need to be discarded in relation to the actual demand. As either the fill rate or the overage rate can be manipulated, we can sum the deviation from the optimum of both key figures, i.e., $(1 - \mathrm{FR}) + \mathrm{LR}$. In order to balance overage and underage, it is necessary to consider costs and apply an asymmetric loss function.

In many cases, the most relevant performance indicator is the actual profit or costs, i.e., deviation from the maximum profit, that are associated with a decision. A decision reflects the chosen order quantity q for an article having demand d. A retailer sells items for price p which cost c per unit (e.g. production costs). Hence, the retailer bears opportunity costs or underage costs $u = p - c$ for the unfilled demand, i.e., the chosen order quantity q is smaller than the demand d. The underage costs can be interpreted as the profit margin of an item. Contrary, items that are not sold cause overage costs $o = c$ if salvage value or waste management costs are ignored.

$$\mathrm{Profit} = p \cdot max(q, d) - q \cdot o \tag{2.34}$$

$$\mathrm{Cost} = u \cdot (d - q)^+ - o \cdot (q - d)^+ \tag{2.35}$$

A general issue with evaluating profits and costs is that exact parameters are hardly known and that long-term effects are difficult to estimate, e.g., customers might switch stores when the demand is frequently not completely fulfilled. Moreover, the point forecast (e.g. expected demand) only maximizes the profit if costs of underestimation and overestimation are equal ($o == u$). If this is not the case, the optimal order quantity can be obtained by applying a newsvendor model (Silver et al., 2017).

2.3.2 Evaluation Schemes

In order to assess the performance of the methods, we need to split an available dataset into a training set and a test set. The training set is used for the selection and training of the models (e.g. fitting the coefficients) while the test set is used to measure the out-of-sample performance. As auto-correlation can be prevalent in time series data, it is important that observations in the test set temporally succeed the training set (see Figure 2.1). Otherwise, information leakage is possible which falsifies the results. Hence, standard k-fold cross-

validation (see Figure 2.2) that is frequently used in the context of the evaluation of ML models is not suitable to measure the out-of-sample performance.

Statistical time series models are frequently selected by minimizing an information criteria like AIC (Akaike, 1974) (e.g. Hyndman and Khandakar (2008)) which considers the complexity and fit of a model. For ML methods, cross-validation is used for model selection. Hence, the full training set needs to be further split into a training set (i.e. a subset of the full training set) and a validation set. Typically, roughly 80% of the data is used for training and the remaining 20% for testing (Hastie et al., 2009; Hyndman and Athanasopoulos, 2014). Bergmeir et al. (2018) show that k-fold cross-validation can be used for selecting ML models in the context of time series forecasting.

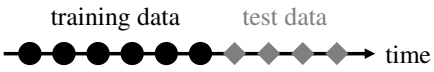


Figure 2.1: Split of training data and test data. The test data succeeds the training data.

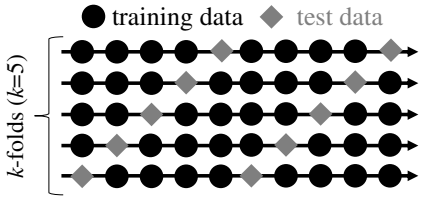


Figure 2.2: Illustration of k-fold cross-validation.

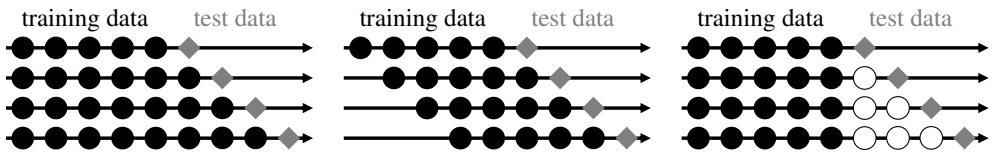


Figure 2.3: Evaluation schemes.

Once the models have been selected, the next step is to determine the out-of-sample performance. In general, a rolling-origin evaluation is typically applied in order to increase the robustness of the results. We consider several evaluation schemes as illustrated in Figure 2.3:

Standard The standard approach is to shift the origin of the forecast by one step (e.g. a day) after each prediction (Hyndman and Athanasopoulos, 2014). Thereby, the size of the training set increases as the most recent observation is added to the training set. After each step of the evaluation, the model is trained with all available data.

Standard (fixed size) A small change to the standard approach is to maintain a constant size of the training data. Hence, the oldest observation is discarded as a new observation is added to the training data.

Fixed train, rolling test Another alternative is to train the models only on the initial training set, but the evaluation is performed in a rolling-origin fashion. Hence, most recent observations can serve as input for the models, but the models are not trained with those observations.

The standard approach should lead to the most accurate results as the model is frequently trained with all data that is available but this approach is very computationally expensive and might not be feasible in a productive setting for certain types of models. The variant with a fixed training size allows to investigate the effect of the size of the training data on the performance, e.g., some model types might require or benefit more from a larger training set than others. The motivation of the last evaluation scheme is to simulate a more realistic application of the models as it cannot be expected that the models are trained after a new observation becomes available. Hence, the models are not frequently (e.g. daily) trained, but new observations can be used as input for the model. However, if the test set covers a long time span (e.g. several months of daily data), an update of the models (e.g. once a month) can be considered.

When we compare methods in this thesis, we always compare them on the same test set (i.e. empirical observations) and the evaluated methods do only incorporate data that would be available beforehand in a productive application scenario. In some cases, we alter the evaluation schemes in order to highlight characteristics of specific model types, but this is always explicitly mentioned and explained.

3

Data Foundation & Characteristics

In this chapter, we describe the data that is typically available to retailers and can be exploited for decision optimization (see Section 3.1). Subsequently, we provide an overview of the empirical datasets that we use for the evaluation of our solution approaches (see Section 3.2). Moreover, we analyze and report the most prevalent characteristics of the empirical datasets in order to justify the solution approaches introduced and evaluated in later parts of the thesis (see Section 3.3). The chapter concludes with a summary in Section 3.4.

3.1 Data Sources & Features

The performance of data-driven methods depends on the scope and quality of the data. With respect to decision optimization at the store-article level in the retail domain, various data sources are available that can be used for building prediction models. The most important information source is the enterprise resource planning system containing master data and transactional data, but also external data sources can be considered:

Master data comprises information about the available stores and products. Stores have a fixed location (i.e. address) that is required to enhance the data with external information. Master data also contains the opening times as they vary among the stores, e.g., not all stores open on Sundays or public holidays. Stores are also assigned to predefined store classes that roughly represent their characteristics, e.g., located in a mall, associated with a supermarket, or a coffeehouse. The products are assigned to specific categories and allow obtaining the aggregated demand on category level.

Transactional data contains the sales of the products which are subject to decision optimization. The sales can be annotated with additional information like the selling price or promotional information. In our application scenario, the bakery also distributes coupons that are valid for a couple of weeks. The information about active coupon periods (i.e. days when coupons are valid) is available as a binary indicator variable at company level. It is also possible to obtain stock-outs from transactional data if delivery quantities and overages are tracked.

External data comprises location-specific data and calendrical information. The calendar allows obtaining the day of the week or the day of the year as we deal with multiple seasonalities. It also contains the public holidays, noted special days, and school holidays. The calendar information depends on the federal state in which the store is

located. In addition to the store classes obtained from the master data, the data can be enriched with location-specific features that describe the local environment of each store. Moreover, weather data (e.g. temperature) can also be considered if accurate short-term forecasts can be obtained.

Hence, for decision optimization we cannot only rely on univariate time series data reflecting the sales but are also able to incorporate additional knowledge that can help to explain demand patterns. However, based on those data sources, we need to derive features that serve as input for prediction models. A brief overview of the considered features that we use for the prediction models, which we evaluate in this thesis, is provided in Table 3.1. The selected features are based on domain knowledge, which we succinctly explain.

The demand of perishable goods is subject to a strong weekly seasonality which makes it reasonable to incorporate lagged sales and features that can be derived from it like a rolling mean or a rolling median which tend to be more robust. The opening times affect the sales as shorter opening hours can be an indicator for lower sales volumes. But also knowledge about the specific hours when the store is open (e.g. only in the morning) can be useful as the demand of the products follow distinct intraday patterns. The calendar information is important to model deterministic seasonalities, public holidays, other special days, and school holidays.

Moreover, when we build models that process data of multiple time series it is possible to incorporate features that allow the prediction model to differentiate among the time series while still being able to learn general patterns. In particular, we consider features that describe the stores and their location. In the master data, the stores are already assigned to classes that characterize and implicitly cluster stores. Additionally, we incorporate information about the location and environment of the stores. For instance, stores in the city differ from stores in suburbs or stores close to schools are more affected by school holidays than stores that are not close to a school. We can also derive features from transactional data like a general weekday pattern (working day, Saturday, Sunday) that enables demand-driven clustering. In order to distinguish between the articles, we include features describing the product category.

3.2 Datasets

For our empirical analysis, we rely on proprietary datasets from the bakery domain that have been provided by industry partners. In this section, we provide an overview of the datasets and indicate how they are used in this thesis (see Table 3.2). While all datasets are related, they have been tailored to address specific research questions and are used to focus on the challenges of specific application scenarios. The datasets share many characteristics that are usually prevalent in this application domain.

Data Source	Features
Master Data	store class, product category, opening times (day, hours/duration)
Transactional Data	lagged sales, rolling median of sales, binary promotional information
External: Calendar	day of year, month, day of month, weekday, public holiday, day type, bridge day, nonworking day, indicators for each special day, school holidays
External: Weather	temperature (minimum, mean, maximum) and cloud cover of lagged days and target day
External: Location	general location (city, suburb, town); in proximity to the store: shops (numbers and types: bakeries, butcher, grocery, kiosk, fast-food, car repair), amenities (worship, medical doctors, hospitals), leisure (playground, sport facility, park), education (kindergarten, school, university)

Table 3.1: Overview of the feature groups considered for the machine learning methods.

ID	Level	Temp. Aggregation	Stores	Articles	Series	Length (max)	Weekdays
v1	SC	daily	141	8 categories	1128	1004 (2.75 years)	Mon-Sun
v2	SA	daily, hourly profiles	5	11 (6 buns, 5 breads)	55	528 (1.7 years)	Mon-Sat
v3a	SA	hourly	9	12 (6 buns, 6 breads)	108	987 (2.7 years)	Mon-Sun
v3b	SA	hourly	9	14 (intraday baking)	121	987 (2.7 years)	Mon-Sun

Table 3.2: Overview of the evaluated datasets (SA: store-article, SC: store-category).

Dataset v1 The first dataset comprises 1128 time series at the store-category level over 33 months. The dataset is used to evaluate the general suitability of data-driven prediction models for large-scale demand forecasting in the retail domain (see Section 4.4). It is well-suited for this purpose as it covers over $820k$ observations from 8 product categories of various types of baked goods (e.g. buns, breads, viennoiseries, cakes, snacks) in 141 stores.

Dataset v2 The second dataset comprises eleven stock-keeping units, namely, five breads and six buns, for five stores over a period of 88 weeks, where each store is open from Monday to Saturday. Even though this is a rather small data excerpt, it already contains roughly $28k$ observations at the daily level. The size of the dataset is still manageable and makes more exhaustive experiments feasible compared to the larger datasets. We use it for the evaluation of various aspects concerning daily retail demand forecasting using Machine Learning (see Part II) and for the analysis of the single-product newsvendor model (see Section 5.3).

Dataset v3a The dataset v3a comprises the six most frequently sold stock-keeping units from the product categories buns and breads for nine stores over a period of 141 weeks which accumulates to close to $100k$ observations. The stores in this dataset open on every weekday, but some breads are not sold on Sundays. Hence, the dataset is significantly larger than dataset v2 as it contains not only more time series but also longer time series. We use this dataset to evaluate solution approaches for the multi-product newsvendor model (see Section 5.4).

Dataset v3b The fourth dataset is closely related to the third dataset (v3a) but the assortment differs as the purpose of this dataset is the analysis of solution approaches for intraday decision support (see Chapter 6). Hence, the dataset contains articles that are relevant for the intraday baking plan. Aside from some buns, the intraday baking assortment also contains other product types like pretzels, croissants, or meat for sandwiches.

All time series represent sales of fast-moving goods, i.e., the items are sold several times per day, and missing values are also no important issue. Additionally, all datasets can be enriched with the explanatory feature data outlined in the previous section (see Section 3.1 and Table 3.1) as only the location of the store and dates are required to map the feature data. Thus, the data foundation is quite comprehensive and enables a variety of empirical experiments. In particular, we evaluate data-driven models and compare them with more established approaches like time series models or linear regression.

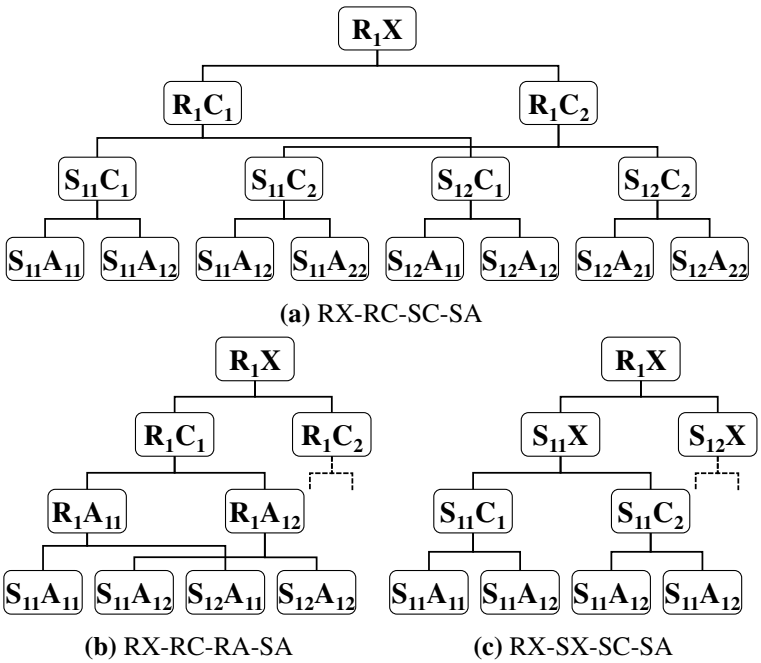
3.3 Data Characteristics

In this section, we highlight some prevalent characteristics of the application domain and the empirical datasets. In particular, we discuss hierarchies and possibilities to construct them from data as well as the seasonalities of the data. Section 3.3.1 is based on the clustering part of the paper *"Cluster-based hierarchical demand forecasting for perishable goods"* by Jakob Huber, Alexander Gossmann and Heiner Stuckenschmidt (Huber et al., 2017). I contributed the respective part of the paper.

3.3.1 Hierarchies & Article Clusters

The organizational structure of retailers can often be presented as a hierarchy (see Figure 3.1). In the retail domain, the stores are typically grouped into regions having their own distribution centers. Moreover, the articles that are offered by the retailer build a hierarchy itself, i.e., several articles can be grouped into an article category. Hence, various hierarchies can be built and exploited for decision support. Demand observations at the store-article level can be aggregated to various higher levels of the hierarchy:

- Region (RX): The total quantity sold within the region.
- Region Category (RC): The total quantity sold of articles of a specific category (cluster) within the region.
- Region Article (SA): The total quantity sold of a specific article within the region.
- Store (SX): The total quantity sold at a specific store.
- Store Category (SC): The total quantity sold of articles of a specific category (cluster) at a specific store.



(a) RX-RC-SC-SA

(b) RX-RC-RA-SA **(c)** RX-SX-SC-SA

Figure 3.1: The figures illustrate hierarchies that can be used for hierarchical forecasting. Each node refers to a time series of a certain level that is identified by two letters. The first letter specifies a region (R) or a store (S). The second letter specifies an article (A), an article group (C), or the group of all articles (X).

- Store Article (SA): The total quantity sold of a specific article at a specific store.

While our goal is decision optimization at the store-article level (see Section 1.2), other levels of the hierarchy are also of interest. For instance, the category levels (RC + SC) can be used to monitor the demand within a group of items and to validate the predictions at article level (RA + SA). Studies indicate that perishable articles (e.g. baked goods) have a high substitution rate in case of stock-outs, e.g., customers buy another article of the same category in the same store (van Woensel et al., 2007). Hence, it is important to maintain a high service level for at least one product of a cluster of substitutable items if not for every article. By risking that some articles are out-of-stock, the total amount of waste can be limited while the expected revenue loss might be acceptable due to the substitution effects (van Donselaar et al., 2006). Predicting the aggregated demand of the complete cluster might lead to more accurate forecasts, which helps to reduce the risk of excessive stock levels and stock-outs for the whole article group, because the time series of single articles can be more volatile and distorted due to various effects (e.g. stock-outs). Moreover, forecasts for a group of substitutable items are valuable if the assortment is changing or some articles are temporarily not available due to delivery problems or item damage. For instance, the demand forecast of an article group can be used to estimate the demand for a new article if a seasonal article gets replaced. For completeness, we also introduce the total aggregation of sales (SX + RX) which are not directly linked to the optimization of order quantities but are relevant for revenue forecasts. For instance, revenue forecasts at store level are relevant for staffing decisions.

3.3.1.1 Article Clustering Approach

In order to gain the most benefit from a hierarchy, it is important to rely on meaningful article clusters. Therefore, we aim to identify groups of comparable products that are potentially substitutes and also beneficial with respect to demand estimation.

For example, Kalchschmidt et al. (2006) cluster customers of warehouses (e.g. stores) according to various criteria (e.g. weekly sales pattern, penetration rate) in order to obtain homogenous groups. Zotteri et al. (2005) cluster time series based on their characteristics (e.g. demand pattern) rather than more intuitive but misleading features like the allocation to a distribution center (e.g. geographical proximity). The demand of each group becomes less uncertain and variable, which leads to more accurate predictions at company level.

We propose to detect article groups automatically by clustering articles according to their intraday sales patterns. In order to perform a cluster analysis, we transform the point-of-sales data into feature vectors $P_{a,q,w}$ representing the intraday sales pattern for each article a in a specific quarter q on each weekday w. Hence, each article is represented by 24 (Monday - Saturday) or 28 (Monday - Sunday) vectors.

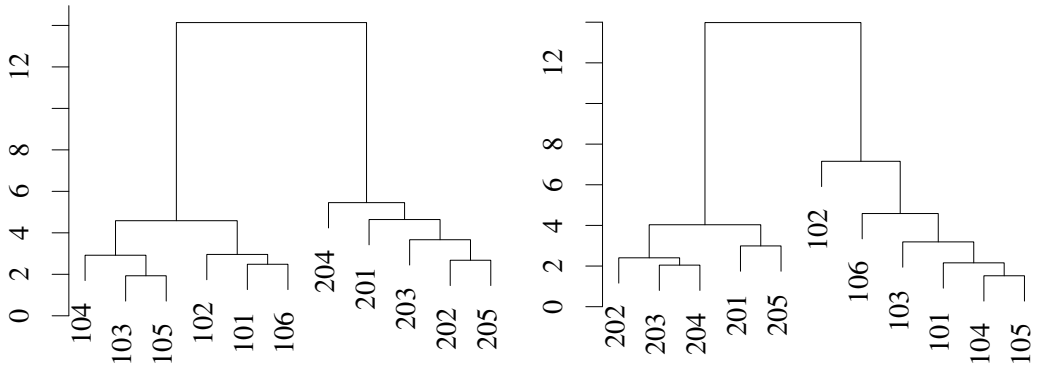
$$P_{a,q,w} = (p_{a,q,w,1}, p_{a,q,w,2}, \ \ldots, \ p_{a,q,w,T}) \tag{3.1}$$

We introduce a vector for each weekday and quarter in order to reveal possible differences in the demand patterns and to cover seasonal aspects. For instance, the demand patterns of working days and weekends could be distinguishable. Moreover, different environmental factors (e.g. weather conditions) might cause different demand patterns in the summer compared to the winter. The length of $P_{a,q,w}$ depends on the maximal number of hours T during which the stores are open. Each element $p_{a,q,w,t}$ represents the average relative proportion of the total daily sales that is sold in the respective hour t.

$$p_{a,q,w,t} = \frac{s_{a,q,w,t}}{\sum_t s_{a,q,w,t}} \tag{3.2}$$

The variable $s_{a,q,w,t}$ represents the total sales of an article a in quarter q on weekday w and hour t. We cluster the generated features with the k-means algorithm. The algorithm ensures that each vector is assigned to exactly one cluster (strict partitioning). Moreover, the center of a cluster can be interpreted as a general demand pattern of the allocated articles. In order to apply the algorithm, one has to set the number of clusters k. It is noteworthy that the number of clusters should be aligned to the characteristics of the demand patterns. Therefore, we suggest applying agglomerative hierarchical clustering in a preceding step as this helps to reveal a hierarchical structure and to determine a suitable number of clusters. A suitable linkage criterion for our use case is Ward's method (Ward Jr, 1963) which merges clusters so that the within cluster variance is minimal.

After the feature vectors are allocated to clusters using the k-means algorithm, we determine the final article groups by majority vote. This is necessary as each article is represented by several feature vectors, and it is not guaranteed that all feature vectors are part of the same



(a) Working Days (Mon - Fri) **(b)** Weekend (Sat)

Figure 3.2: Hierarchical cluster analysis based on intraday sales patterns of articles on working days and the weekend. The dataset yields two main clusters which match the two article categories.

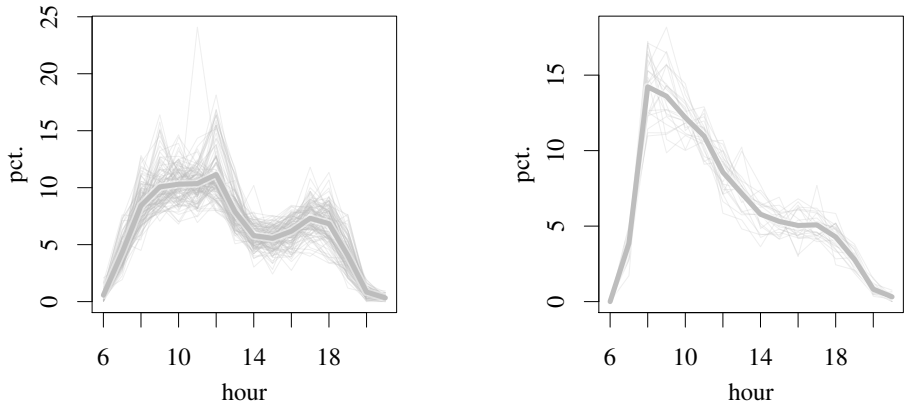
cluster. Thus, we assign an article to the cluster to which most of its feature vectors belong. The obtained article clusters can be complemented with the organizational structure to build the hierarchy.

3.3.1.2 Article Clustering Evaluation

We apply the proposed clustering approach to dataset v2, which contains 6 buns (ids: 101-106) and 5 breads (ids: 201-205). The hierarchical cluster analysis reveals that the demand patterns of working days are distinguishable from weekends. Moreover, we observe that the two groups of articles match the article groups buns and breads (see Figure 3.2), i.e., the demand patterns of buns and breads are clearly distinguishable. Based on these observations, we decide to split the feature vectors into one set that contains feature vectors of working days and another set that contains all feature vectors of weekend sales patterns. For each set of vectors, we apply k-means with $k = 2$.

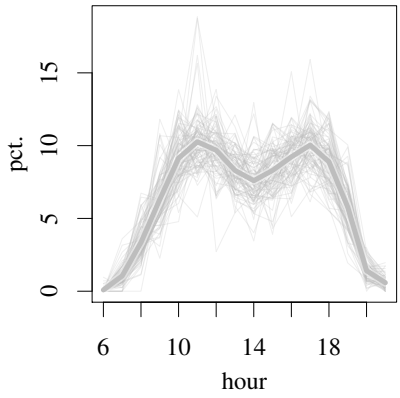
The results of the cluster analysis are depicted in Figure 3.3. Overall, the resulting clusters are quite pure and accurate compared to the given article category assignment. In this case, we use the original category assignment as the gold standard as the two categories already contain substitutable goods and thus are reasonable clusters. However, this has not to be the case in other scenarios. The cluster analysis shows that the demand patterns for buns (see Figures 3.3a & 3.3b) are distinguishable from breads (see Figures 3.3c & 3.3d). Based on these results, articles 101-106 (buns) and articles 201-205 (breads) can be grouped. Those clusters can be used for hierarchical forecasting.

It is also mentionable that the patterns for different buns (breads) are similar, which underlines the assumption that they have comparable characteristics with respect to the customer demand. The clusters show that buns are mostly sold in the mornings, while the demand for breads is higher in the afternoon. This suggests that buns are the preferred product in the morning, whereas bread sales are rather equally distributed over the day. Moreover, we

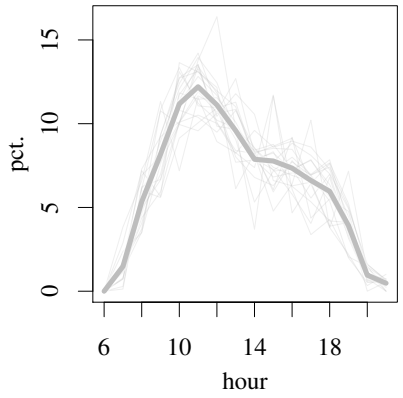


(a) Buns (working days)

(b) Buns (weekend)



(c) Breads (working days)



(d) Buns (weekend)

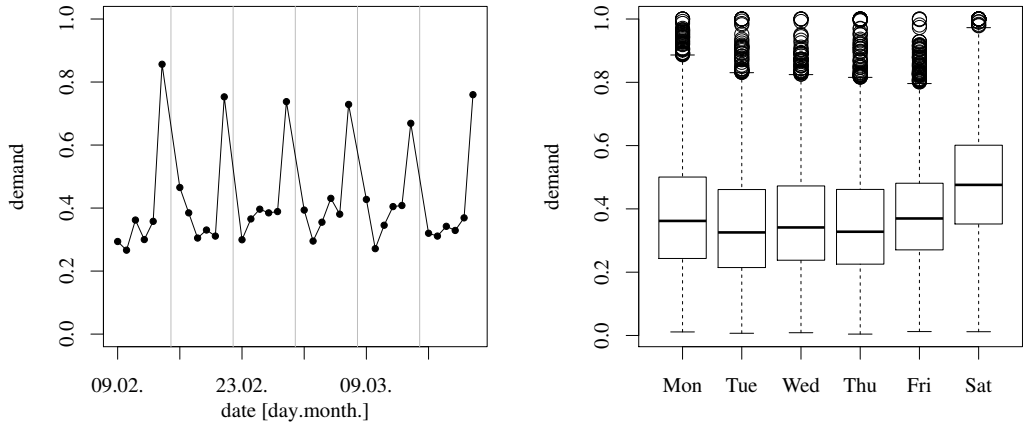
Figure 3.3: The intraday sales patterns are clustered using k-Means. The clusters match the article categories and the type of the weekday.

observe peaks during lunchtime and in the afternoon, which seem to be related to the working times of employees in Germany. We also observe that the demand patterns of working days are different from the weekend. On Saturday, the demand for buns is very high in the mornings and drops continuously during the day. For breads, we do not observe the second peak in the afternoon that we see on working days. For all clusters, we observe that the sales drastically decrease during the last opening hours due to less demand. Hence, running out-of-stock during the last hour of the opening hours may not have a big impact on the revenues and might be acceptable if it decreases the amount of discarded goods.

3.3.2 Seasonalities

The results presented in the previous section illustrate that an intraday seasonality is prevalent while the demand on working days differs from the weekend. An analysis of daily demand data confirms the weekly seasonality on dataset v2. The demand on working days (i.e. Monday - Friday) is on a comparable level, while the demand on the weekend (Saturday) is higher. Figure 3.4 shows the strong weekly seasonality of demand for (a) a representative product and (b) a box plot that confirms this pattern for all time series. While the median demand on Tues-

days and Thursdays is the lowest, it is slightly higher on Mondays, Wednesdays and Fridays. The median demand on Saturday is higher than it is for all other days. The standard deviation of demand does not vary strongly across the weekdays. A subtle yearly seasonality is also present, which is mostly reflected by public holidays (e.g. Christmas) and a slightly lower demand during the summer.



(a) Exemplary demand time series of six weeks

(b) Boxplot of demand per weekday

Figure 3.4: The demand shows a strong weekly seasonality. The demand levels for working days (Mon-Fri) are comparable, while the demand level on the weekend (Sat) is noticeably higher.

3.4 Summary

In this chapter, we outlined data sources which are usually available in the retail industry and how operational data from the enterprise resource planning system can be enhanced with additional explanatory data that can be useful for the construction of prediction models that are part of a data-driven decision support system. The available datasets, which only represent a rather small excerpt of the respective company, illustrate that this application domain offers large amounts of data. Hence, we are interested how this data can be exploited for better decision making.

We also emphasize specific characteristics of the application domain. For instance, time series data at lower levels can be aggregated or grouped to build hierarchies. Different levels in the hierarchy are relevant for different planning phases but hierarchies can also be used to validate predictions or to reduce computational costs. For instance, the aggregated demand of a category of substitutable products can be used to monitor the demand at article level. This is reasonable as substitution rates for baked goods are higher than for other article categories and thus maintaining a high service level for only one article per group might be reasonable in order to limit the amount of discarded goods without sacrificing the revenues.

To this end, we propose a clustering approach for the identification of potentially substitutable articles. The empirical evaluation, using different types of baked goods, indicates that

it is indeed possible to cluster articles based on their intraday demand patterns. For instance, the patterns of buns and breads are clearly distinguishable, which confirms the expectations of domain experts. Hence, the identified clusters contain substitutional articles which is convenient as the substitution rates in case of stock-outs are high for perishable goods. Intraday sales patterns provide also implications for store operations, e.g., shelf replenishment needs to be aligned with periods of high demand. Moreover, intraday sales patterns can be used to decensor sales data in case of stock-outs (Lau and Lau, 1996).

We also highlight that the demand for baked goods is subject to a strong weekly seasonal demand pattern which justifies the focus on prediction models that are suitable for seasonal time series data.

Part II

Forecasting

4

Large-scale Forecasting

The research presented in this chapter is based on a paper titled *"Daily Retail Demand Forecasting using Machine Learning with Emphasis on Calendric Special Days"* by Jakob Huber and Heiner Stuckenschmidt. Additionally, this chapter covers aspects from the paper *"Cluster-based hierarchical demand forecasting for perishable goods"* by Jakob Huber, Alexander Gossmann and Heiner Stuckenschmidt (Huber et al., 2017). I have contributed the parts of the papers that are contained in this work.

4.1 Introduction

Demand forecasts are an essential input for many operational decisions (e.g. see Part III). In this chapter, we study the suitability of Machine Learning (ML) methods for large-scale demand forecasting, e.g., the use case that is the subject of this thesis (see Section 1.2).

The application scenario of a retailer offering fast moving goods has several characteristics that make it obvious to explore the competitiveness of ML methods and concepts. Retailers operate numerous stores and offer a broad assortment. Hence, they accumulate large amounts of data describing business transactions. The datasets that we use for the empirical evaluation are already relatively large despite the fact that they only represent an excerpt of the respective company (see Section 3.2). The transactional data from the retailers can not only be enhanced with additional explanatory data but it can also be expected that time series (e.g. at the store-article level) are similar:

- First, the stores belong to the same company which means that they share many characteristics including the branding, pricing, and assortment. Even though the stores are not exactly the same, they belong to common classes depending on their location and facility equipment.

- Second, the stores are located in a geographically restricted area which means that also the customers as well as the market environment (e.g. comparable competitors) of the stores share similarities and the external influences (e.g. weather) are fairly comparable.

In consequence of those assumptions, a very large data pool is available for exploitation and building of prediction models. From existing literature it is unclear if ML methods are able to outperform established approaches for retail demand forecasting (see Section 4.2). We suspect that the characteristics of the present use case play to the strength of ML. To this end,

we propose and evaluate various modeling possibilities in the context of a real-world daily demand forecasting application (see Section 4.3).

We are especially interested in daily demand forecasts at the store-article level for the next day(s) because they are required for decision making concerning order quantities. Short-term predictions are sufficient as the supply chain is quite agile, i.e., the lead time for the considered articles is only one day. Consequently, we perform an empirical evaluation that targets different aspects of the use case.

In Section 4.4, we leverage our largest dataset (dataset v1, see Section 3.2) to conduct an analysis of daily forecasts at the store-category level. Forecasts at the store-category level are important as they are input for operational planning by managers as they follow a top-down planning approach. This is reasonable because it is possible to rely on the customer's willingness to substitute in case of a stock-out (van Woensel et al., 2007). Moreover, we emphasize the challenge of demand forecasting on specials days that are subject to vastly different demand patterns. While dataset v1 allows illustrating the usefulness of ML, it lacks information about demand at the store-article level that we need for decision optimization, and it is also too large to conduct more comprehensive experiments.

For the aforementioned reason, we rely on dataset v2 for the experiments in the remaining sections. This dataset contains information at the store-article level and its size makes more exhaustive experiments practically feasible. In Section 4.5, we study the effect of different levels of data usage which comprise the scope of the model, the length of the demand history, and the value of explanatory feature data. In this context, we also study different formulations of the forecasting task as a supervised learning problem including various options to transform the regression problem to a classification problem. In Section 4.6, we investigate the possibility to leverage hierarchical forecasts in order to reduce computational costs and illustrate the viability of ML at different levels of the organizational hierarchy and the article hierarchy.

We conclude this chapter in Section 4.7 by summarizing and discussing the results of the empirical evaluation. In particular, we elaborate on the viability of ML in a large-scale demand forecasting scenario and highlight criteria that affect the performance of ML.

4.2 Related Work

Our research is concerned with demand forecasting for fast moving perishable goods in the retail industry which is a large-scale demand forecasting scenario. Thus, we review the literature on time series forecasting using ML (see Section 4.2.1). Moreover, we focus on time series applications that are related to the characteristics of our use case. Section 4.2.2 summarizes literature that is concerned with forecasting for special occasions while Section 4.2.3 recaps literature on hierarchical forecasting. We refer to Fildes et al. (2018) for a general literature review on retail forecasting. At the end of this section, we briefly outline research gaps (see Section 4.2.4).

4.2.1 Forecasting using Machine Learning

Statistical time series methods (e.g. exponential smoothing, ARIMA models) have been successfully applied to many forecasting problems and there is no definite evidence that they are inferior to ML methods (e.g. Ahmed et al. (2010); Crone et al. (2011); Makridakis et al. (2018a)). The results of the most recent M4 Forecasting Competition suggest that (combinations of) statistical methods outperform pure ML methods while a hybrid approach performed best for forecasting of univariate time series (Makridakis et al., 2018b). Ahmed et al. (2010) compare a variety of ML methods including artificial neural networks (ANNs) and regression trees on a subset of the monthly time series of the M3 competition. They conclude that ML methods, especially ANNs, are contenders to classical statistical models. Makridakis et al. (2018a) conclude in a similar study that ML methods are inferior to statistical forecasting methods. However, the findings of Crone et al. (2011) highlight that no approach works best under all circumstances.

The most popular ML models with respect to time series forecasting are ANNs. ANNs have been extensively studied in the context of time series forecasting for more than two decades (Adya and Collopy, 1998; Zhang et al., 1998). Alon et al. (2001) report that ANNs are superior to ARIMA models and multiple regression for forecasting monthly aggregate retail sales with a strong trend and seasonal patterns. The study of Chu and Zhang (2003) emphasizes that deseasonalization is preferable over other modeling options if ANNs are applied while Crone and Kourentzes (2009) were able to model deterministic seasonality with trigonometric functions which suggests that deseasonalization is not always required. Aburto and Weber (2007) propose a hybrid demand forecasting approach for retail sales based on ARIMA and ANNs whereby the ANNs are trained on the residuals of the ARIMA model. Doganis et al. (2006) forecast the demand of short shelf-life products with a radial basis function ANN whose variables are selected using evolutionary computing techniques. The proposed model produces more accurate predictions than various linear reference methods. Contrary, Carbonneau et al. (2008) report that recurrent neural networks perform better than support vector machines but do not outperform traditional approaches like moving average or linear regression in the context of monthly demand forecasting of a supply chain.

The vast majority of the comparative studies do not exploit the strength of data-driven ML methods because the results are mostly based on univariate time series forecasting. For instance, studies based on the M3 or NN3 dataset only cover monthly time series with 14 to 126 samples. Thus, the derived training dataset is also quite small, which quickly leads to an unfavorable ratio between the number of observations and the parameters of the model.

4.2.2 Forecasting on Special Occasions

Forecasting in the retail domain primarily focuses on promotions rather than special days. However, the requirements for promotional forecasting are similar and public holidays as well as major festivities are frequently considered in the proposed models. Cooper et al. (1999) present a promotional forecasting systems for weekly retail data based on a regression-style

model that incorporates dummy variables for public holidays. van Heerde et al. (2002) and Divakar et al. (2005) discuss the possibility to vary the scope of the model and to fit it to different levels of aggregation. Gür Ali et al. (2009) present a study for weekly forecasts of perishable goods having a durability of several days at store-product level. They state that data pooling improves the results while only more sophisticated methods (e.g. regression trees) benefit from more detailed input. According to van Donselaar et al. (2016), models that are fitted over multiple categories are only more accurate if the data foundation is sufficient. Huang et al. (2014) propose a regression model for aggregated retail data and observe that the accuracy of the evaluated approaches is rather comparable for normal weeks while improvements are possible during promotional periods. Ma et al. (2016) and Ma and Fildes (2017) report that integrating more data, i.e., cross-categorical information, leads to more accurate predictions. They also suggest that regular re-fitting and variable selection is required.

Trapero et al. (2013, 2015) report that judgmental adjustments for promotions add value but are not better than statistical models. Kourentzes and Petropoulos (2016) stress the importance of an automatized approach for promotional forecasting. Ramanathan and Muyldermans (2010) present promotional factors which influence the demand. Those factors include special days (upcoming holidays, festivals: Easter / Christmas), seasonal factors (e.g. temperature), and promotional factors. An evaluation based on structural equation modeling leads to the conclusion that the relevant factors depend on the product or product family (Ramanathan and Muyldermans, 2011). van Heerde et al. (2000) observe that also pre- and post-promotion effects noticeably influence sales.

The aforementioned studies are based on weekly data as this level of granularity is sufficient for many operational decisions. While some special days are frequently modeled using binary dummy variables, they were not of specific interest. We suspect that a reason for this is that the effect of special days is mitigated on the weekly level and possibly dominated by the promotions. The few studies in business forecasting literature that are dedicated to daily retail forecasting did also not emphasize the challenges related to special days. Taylor (2007b) uses exponential smoothing to compute prediction intervals for daily supermarket sales. Public holidays and periods with unusual demand are explicitly excluded from the evaluation as the considered methods are not designed for those scenarios. Di Pillo et al. (2016) employ support vector machines to forecast daily retail data but do not address the challenges related to special days. Arunraj and Ahrens (2015) develop an S-ARIMAX model that incorporates binary dummy variables for holidays for the prediction of daily banana sales in a single store but are also not concerned with the forecasting accuracy on special days. Kolassa (2016) discusses challenges related to the evaluation of intermittent daily retail data and argues that it is important to evaluate predictive distributions instead of specific functionals for such data.

Calendrical special days are explicit subject in the context of intra-day load forecasts. Srinivasan et al. (1995) highlight the importance of modeling special days for hourly load forecasting using a fuzzy ANN. They employ a dedicated model for each of the three day types which are weekdays, Saturdays, and Sundays plus public holidays. Similarly, Wang

and Ramsay (1998) also train one model per day type and report that the errors for public holidays are the highest which is justified with the fact that those days fall in different seasons. Kim (2013) incorporates special days in a double seasonal ARIMA model by treating special days that are subject to a similar pattern identically. Cancelo et al. (2008) also highlights the importance of treatments for special days and events. Soares and Medeiros (2008) identify a total of 15 day types including the weekdays, days before and after public holidays, and bridge days. Panapakidis (2016) proposes to cluster special days according to their load pattern. The cluster information as well as the average load of reference days is fed into an ANN. Barrow and Kourentzes (2018) model non-calendrical special days in the context of call center call arrivals with an ANN. They report the superiority over standard statistical models and provide empirical evidence that it is better to incorporate special days in the model rather than building a separate models for special days.

4.2.3 Hierarchical Forecasting

Kahn (1998) argues that direct forecasts are preferable over hierarchical forecasts if the data characteristics (e.g. seasonality, trend) differ. An aggregated time series does not necessarily reflect the characteristics of the time series at lower levels, which leads to inaccurate forecasts. While the bottom-up approach leads to better forecasts at the lower level, the errors might aggregate and lead to poor forecasts at intermediate and top-levels. Thus, hierarchical forecasting works best if the low level time series share the same pattern. With respect to top-down forecasting, he proposes to proportion the forecast based on seasonal indices. Viswanathan et al. (2008) perform a simulation study to examine the relative effectiveness of top-down and bottom-up forecasting for substitutable products. At product level, the top-down approach is preferred if the degree of substitutability is high. However, if the variability of the demand proportions is high, direct forecasts are required as an accurate demand allocation is not possible. At product group level, the direct forecast outperforms the bottom-up approach when the demand variability at product level is high and the degree of substitutability between products decreases. Williams and Waller (2011) conduct a study based on weekly point-of-sales (POS) data of cereals which are a fast moving consumer good. They conclude that the bottom-up approach is preferred for forecasting each stock-keeping unit at store level and region level if POS data is available. Widiarta et al. (2009) report that the differences of bottom-up and top-down strategies are not significant if the demand processes (e.g. MA(1)) are identical at all levels.

Kalchschmidt et al. (2006) cluster customers of warehouses (e.g. stores) according to various criteria (e.g. weekly sales pattern, penetration rate) in order to obtain homogenous groups. The demand of each group becomes less uncertain and variable, which leads to more accurate predictions at company level. Zotteri et al. (2005) analyze the impact of the aggregation level (chain, store and cluster level) on the forecasting performance. The objective is to forecast the demand at chain and store level for each item. They report that clustering is beneficial for fast-moving items, while a bottom-up approach is better for slow-moving items.

The top-down approach has the advantage of minimizing the total number of forecasts, which leads to reduced computational costs.

4.2.4 Discussion

To summarize, we notice that the literature with respect to the retail sector is quite comprehensive but mostly focuses on aggregated data with respect to the organizational (store level vs. company level) and temporal (weekly data vs. daily data) hierarchy. While ML methods are considered in some studies, factors that influence the performance are not investigated and the reported performance was mostly comparatively poor. However, it is frequently stated that incorporating more data as well as data pooling is beneficial. This makes it reasonable to study the viability of ML for data-rich application scenarios.

Moreover, empirical studies in the retail domain focus on promotions rather than special days. The effect of calendrical special days is mitigated on weekly data, and thus was never focus of existing studies. The proposed promotional forecasting models are mostly multivariate causal linear models or univariate time series models with adjustments in a post-processing step (e.g. base-times-lift method). This is typically justified with the enhanced interpretability of the models. In our use case, the accuracy of a model is more important than its interpretability as the forecasts are eventually input for automated decisions. This allows us to investigate data-driven ML methods that are not as easy to interpret and are often considered as a black box. However, this does not hinder judgmental adjustments, which are common in practice if the demand is expected to deviate from the normal pattern. We argue that there is a need to evaluate ML methods on large-scale datasets covering time series that are enriched with explanatory data (e.g. domain knowledge). In order to address this gap, we outline the modeling possibilities of supervised ML for time series forecasting with respect to the learning task (e.g. regression vs. classification) and the scope of the model (e.g. pooled regression). We are neither aware of published work that covers an exploration of modeling possibilities in said direction nor an evaluation of ML approaches on large-scale for daily retail data. Moreover, we study and highlight criteria that influence the performance of ML.

4.3 Methodology

In this section, we propose and discuss various modeling possibilities for the application of ML for large-scale time series forecasting. Recall the definition of a time series $Y = (y_1, \dots, y_n)$ with $y_t \in \mathbb{R}$ where each data point y_t can be enriched by explanatory variables $X = (x_1, \dots, x_n)$, $x_t \in \mathbb{R}^p$ (see Section 2.2). We consider a scenario that comprises a large set $\mathcal{S} = \{(X_1, Y_1), \dots, (X_N, Y_N)\}$ of time series $Y_s \in \mathbb{R}^{n_s}$ and their respective explanatory variables $X_s \in \mathbb{R}^{n_s \times p}$. The learning methods do not explicitly distinguish between the source of the information. Instead, they learn from data which features influence the performance by minimizing a loss function. We allow a varying length n_s of the time series $Y_s \in \mathcal{S}$ while the structure of the external information has to be identical for all time series. The

forecasting task is to predict the next value (y_{n_s+1}) (single-step forecast) or the next h values $(y_{n_s+1}, \ldots, y_{n_s+h})$ (multi-step forecast) (Bontempi et al., 2012; Ben Taieb et al., 2012) of a time series Y_s based on the available information.

4.3.1 Model Scope

We describe in (2.15) and (2.16) in Section 2.2.1 how a dataset D_s can be obtained from $(X_s, Y_s) \in \mathcal{S}$. As we consider multiple time series, we have a set of datasets $\{D_1, \ldots, D_N\}$ which are derived from $\mathcal{S}$ and thus have the same structure. This makes it possible to unify them to one dataset $\mathcal{D} = \cup_{s=1..N} D_s$. Hence, we can vary the scope of the model, i.e., we can train a model $f(\cdot)$ on an arbitrary subset of $\mathcal{D}$. Training a model on an unified dataset can be interpreted as pooled regression because the application scenario offers a very large number of potentially similar time series. We explicitly do not consider modeling the forecasting problem as a multivariate regression problem by treating each time series as a different dependent variable. Multivariate regression is not well-suited as we deal with time series having different lengths and a changing number of time series at a given point in time. For instance, it might be possible that a retailer opens or closes a store.

It is common practice to select and fit models per time series, which is often the most appropriate approach. However, our application scenario allows us to employ pooled regression. Thereby, the size of the training data that is available to the training algorithm can be significantly increased. This makes it more likely that the ML method separates actual patterns from noise in the data and it also reduces the likelihood to overfit the data. In consequence, the trained model should be more robust and ultimately have an improved accuracy. Another advantage is that a globally trained model can be applied to new or short time series (e.g. new stores or articles, changing assortment). Reducing the number of models that need to be maintained (e.g. feature engineering, hyper-parameter optimization, persisting) makes this approach more viable in practice. However, this only makes sense if common patterns are present in multiple time series and can be transferred.

So far, we outlined the advantages of unifying datasets which are derived from different time series. However, we also pointed out that the patterns of the time series should be comparable. Hence, one might suggest that it makes sense to cluster time series in $\mathcal{S}$ and train a model per cluster. However, clustering is an unsupervised task, and it is not trivial to identify features that describe clusters so that the resulting forecasting accuracy is minimized. Moreover, clusters are often fuzzy and do not allow a clear distinction of the groups. Petropoulos et al. (2014) and Kang et al. (2017) propose time series features in order to infer a model that is most suitable to forecast a specific time series. We suggest expanding the feature space with time invariant features (e.g. the location of a store or features of a product) that allow a ML method to implicitly cluster time series while minimizing the loss function. So, this end-to-end training of a model solves the problem of explicitly identifying subsets of $\mathcal{D}$ based on fuzzy clusters. However, ML methods are in principle able to distinguish between differ-

ent time series and adapt the parameters θ of the model accordingly if relevant features are provided.

In summary, the application of ML allows to vary the scope of the model. By unifying datasets derived from time series, it is possible to fit models on larger datasets which makes it more likely to separate noise from actual demand patterns while the risk of overfitting is reduced. In order to distinguish between different time series during the training process, it is possible to add time invariant features that allow the learning algorithm of a model to implicitly cluster time series. A global model is also easier to maintain which makes the approach viable in practice. Hence, we can extend and vary the data foundation on three levels:

- **Time series length.** The available demand history affects the number of samples.
- **Features.** The dataset can be enhanced with additional explanatory information.
- **Model scope.** It is possible to pool data and to increase the scope of the model.

4.3.2 Regression vs. Classification

Supervised ML distinguishes between *regression* and *classification*. It is more natural to model time series forecasting as a regression task as the target values are quantitative. However, it is also possible to convert a regression problem to a classification problem as both are essentially function approximation tasks (Hastie et al., 2009). The transformation of the learning problem consists of the following steps:

1. The target values of the regression problem need to be discretized and binned. Moreover, a numerical value needs to be assigned to each bin.
2. The (ordinal) classes need to be encoded in order to be processed by a learning algorithm. Moreover, a suitable loss function has to be selected.
3. The prediction of the model, e.g., a probability distribution over the classes, has to be transformed to a numeric value which represents the forecast. Hence, the predicted class has to be picked.

Creation of the Classes. The target values G of a classification task are qualitative. Thus, a surjective mapping $\mathcal{G} : Y \rightarrow G$ between quantitative values Y and qualitative values G needs to be defined. The mapping $\mathcal{G}$ may group multiple target values, i.e., values from the dependent variable of the regression problem, which means that the granularity of the mapping also impacts the accuracy. Moreover, it is also necessary to define an inverse function $\mathcal{G}^{-1} : G \rightarrow Y$ in order to obtain a numeric value for the predicted class that represents the forecast. An example is provided in Table 4.1. A possibility is to create a bin after each percentile of the empirical distribution of the quantitative target values in the training data.

Target	Class	From	To	Class Value
1				
2	g_1	1	1	1
3	g_2	2	3	2.5
4	g_3	4	4	4

Table 4.1: Transformation of a regression problem to a classification problem. The values of the dependent variable of the regressions problem, i.e., the target values of the ML task, need to be binned and allocated to classes. Thus, it is necessary to define intervals ([from, to]). Consequently, a numeric class value needs to be assigned to each target class, e.g., the mean of the interval endpoints of the class.

Class	Encoding		
g_1	1	0	0
g_2	0	1	0
g_3	0	0	1

Table 4.2: The *1-hot* encoding allows to directly predict the probability for each target class ($g_1 < g_2 < g_3$).

Encoding of the Classes. After binning the values, we have a multi-class classification problem with K-levels: $G = \{g_1, \ldots, g_K\}$. The classes $g_k \in G$ need to be encoded for the ML algorithm. We consider two strategies: *1-hot* encoding and *ordinal* encoding.

For *1-hot* encoding, the classes are represented by one-hot encoded binary K-dimensional vectors, i.e., each element of the vector represents a class (see Table 4.2). The learning algorithm minimizes the categorical cross-entropy loss and a trained model predicts the probability distribution over the target classes. This is the typical approach for classification problems.

Another option is to exploit the fact that the variables G are ordinal (ordered categorical), i.e., an ordering exists but no metric is appropriate. Hence, we adapt the encoding in order to exploit this information (Cheng et al., 2008; Gutiérrez et al., 2016) (see Table 4.3). For the *ordinal* encoding, the learning algorithm can minimize the mean squared error. The prediction of the model can be interpreted as the probability that the actual value is less or equal to the numerical value of a specific class. A potential drawback of the ordinal encoding is that a cut-off value needs to be set in order to pick the predicted class. Alternatively, a subsequent regression model can be trained.

Selection of the predicted Class. The transformation to a classification problem has several advantages: The standard regression approach directly predicts only one value (e.g. mean,

Class	Encoding		
g_1	1	0	0
g_2	1	1	0
g_3	1	1	1

Table 4.3: The *ordinal* encoding allows to predict the probability that the target is greater or equal to the value of a specific class ($g_1 < g_2 < g_3$).

quantile) depending on the loss function $\mathcal{L}$. In order to obtain the probability distribution of the forecast, one has to make distributional assumptions or rely on historical data (e.g. empirical distribution) (Kolassa, 2016). A survey on density and probabilistic forecasting is given by Tay and Wallis (2000) and Gneiting and Katzfuss (2014). In contrast, classification models can explicitly predict a complete (discrete) probability distribution over all classes which is basically a density forecast and does allow to obtain not only the mean of the distribution but also different quantiles or the mode.

If *1-hot* encoding is used, the class having the highest probability can be selected. But it is also possible to exploit the ordering of the classes in the post-processing step and transform the prediction to a discretized density forecast which allows to pick different quantiles. If *ordinal* encoding is used, it is necessary to define thresholds as the prediction for each class can be interpreted as the probability that the actual value is smaller or equal to the value of the class. For instance, the class having the highest assigned value and still a predicted value greater than or equal to 0.5 can be a suitable (median) point forecast.

4.3.3 Additional Remarks

The presented demand forecasting framework already illustrates flexibility by providing many options to model time series forecasting in order to be processed by ML methods. However, there are still open questions that need to be considered before applying ML:

- First, a number of ML methods are available. Thus, the selection of the method is an important decision. Depending on the method, model specific parameters need to be set and optimized. This includes the architecture of the model (e.g. size of the trees, the number of hidden layers + nodes, activation functions) but also hyper-parameters (e.g. learning rate). In order to optimize the hyper-parameters of a ML model, grid search, random search (Bergstra and Bengio, 2012), tree of parzen estimators (Bergstra et al., 2011, 2013), or Bayesian optimization (Snoek et al., 2012) have been proposed.

- Second, the dataset D can be sliced along two dimensions: samples and features. By selecting a subset of samples, it is possible to adjust the scope of the model. While it would be preferable to obtain a global model, it might be beneficial to specify models for different subsets. We also want to point out that the scope of the model can change between the training phase and the application phase. For instance, for some time series the best option might be to use a global model while other time series may be more accurately forecast when the model is only trained on its own data while the data is also used to train the global model. While we assume that identical features are provided for each time series in $\mathcal{S}$, it might be beneficial to only use a subset of the provided features (feature selection).

  There is already a research stream that focuses on automatizing the whole ML process and partially addresses the first two issues (Feurer et al., 2015; Bischl et al., 2016). It

can be expected that at least some time series forecasting applications will benefit from those approaches.

- Third, when it comes to the application of a model, it needs to be decided how frequently it needs to be re-trained. This has to be carefully decided because the training process is quite time consuming. If the model is trained on a large data basis and is able to learn a representation of the existing patterns, regular training is often not required as it does not add much value. During the training phase, the ordering of the observations is not preserved, which means that no special emphasis is put on the most recent observations. However, it also possible to update a pre-trained model and adjust its parameters θ by fine-tuning a model with the most recent observations.

- Fourth, the current formulation of the forecasting problem allows only one-step prediction. For multi-step forecasts, it is possible to iteratively apply the model. However, there are also alternatives that can be considered. For instance, a model can be trained for each forecasting step or a model having multiple outs can be constructed. Different possibilities have been proposed and evaluated by Ahmed et al. (2010), Bontempi et al. (2012), and Ben Taieb et al. (2012).

In the following sections, we apply ML to illustrate its viability for large-scale forecasting and highlight criteria that affect the performance of ML methods. Section 4.4 is concerned with forecasts at the store-category level with emphasis on calendric special days. We study forecasts at the store-article level in Section 4.5. Moreover, we investigate forecasts at other levels and possibilities to exploit the hierarchies in Section 4.6.

4.4 Application: Store-Category Level and Special Days

In this section, we study demand forecasting at the store-category level using dataset v1 (see Section 3.2). All operational figures are driven from observations on the lowest organizational level, which is the store-article level (see Section 3.3.1). Hence, the primary patterns that are present at this level get also propagated to aggregated levels. A category comprises a group of products having comparable characteristics. We consider the most common product categories including buns, breads, viennoiseries, cakes, and snacks. Those categories are typically provided by companies in the bakery industry. We evaluate our approach on aggregated data as it is cleaner, e.g., demand distortion due to stock-outs is reduced because of high substitution rates (van Woensel et al., 2007), and we do not have to deal with challenges arising from a changing assortment. Studies indicate that the willingness to substitute is higher for perishable items (84%) than for other product categories (50% (Gruen et al., 2002)) which is caused by an immediacy effect, i.e., the item is needed on the day it is bought (van Woensel et al., 2007). van Woensel et al. (2007) argue that a high service level for all items leads to plenty of left overs and van Donselaar et al. (2006) emphasize that it is important to monitor

the demand at category level if the articles are substitutable as this is an indicator for expected waste.

The demand in the bakery domain is subject to a strong weekly seasonal pattern. However, this is not entirely true for special days (SDs) that are subject to vastly different demand, which makes forecasting for such days a key challenge. A large share of the stores in dataset v1 also opens on public holidays. However, even stores that are not open on public holidays are affected on preceding and succeeding days. Thus, our definition of special days comprises public holidays, neighboring days of public holidays, and other calendric events (e.g. Carnival). The daily demand on such days differs from regular days as customers change their daily routine. We introduce a classification of special days in Section 4.4.1 and specific features for special days in Section 4.4.2. We outline the experimental design of the empirical evaluation in Section 4.4.3 and present the results in Section 4.4.4.

4.4.1 Day Classification

Calendric special days (SD) are days on which the demand vastly differs from the regular pattern due to a calendric event. Special days are primarily triggered by public holidays. Public holidays do often fall on working days but the typical working schedule of most people (i.e. customers) and the offered assortment in the stores as well as the opening times are rather comparable with Sundays. The German constitution states that public holidays are equal to Sundays with respect to the permission to open a store or work. Moreover, the neighboring days of public holidays are also affected as they could fall between the public holiday and the weekend which means that people typically take an extra day off. Neighboring days of public holidays can also be used to compensate the demand from public holidays if the store was closed. Hence, depending on the location of the store, the demand on special days can be lower or higher compared to regular days. However, other festivities that are not related to official public holidays like carnival can be considered as special days. In the case of public holidays, not only the actual day but also the surrounding days are affected. Hence, we consider the following special day classification that we will use for the evaluation:

- **SD1: special day (t):** The day of the public holiday or another significant event.
- **SD2: day before (t-1):** The day before a public holiday is affected as people tend to stockpile. If a public holiday falls on a Monday, the previous Saturday ($t - 2$) also belongs to this class.
- **SD3: day after (t+1):** The day after a public holiday. It can also be a bridge day, e.g., a day between a public holiday and the weekend.
- **SD4: following week (t+7):** The demand reverts back to the normal pattern in the following week. As the demand has a strong weekly seasonality, we use this day type as a sanity check. Autoregressive models might under- or overestimate the demand if the demand pattern is not accordingly learned by the model.

The type of days are ordered according to their precedence: $t > t-1 > t+1 > t+7$. A date is only assigned to the class having the highest precedence if multiple classes apply. The neighboring days $(t-1, t+1)$ are only of interest for public holidays as no noticeable effects are observed for other special days. We refer to days that are not part of the aforementioned classes as SD0 (regular days).

An additional challenge is that some public holidays are not always on the same weekday, which makes it difficult to quantify the actual effect of the public holiday. It is in particular difficult to forecast the demand on special days as the number of historic observations is limited, i.e., only one observation per time series and year, and different special days are subject to different changes. For instance, the sales on Christmas are not comparable to any other special day. Hence, there are also long-term relationships that need to be considered. However, it is also not only sufficient to consider the observed sales of the previous year as the weekday changes for some special days and the demand can be subject to a general trend or level shift. Moreover, some public holidays are on a fixed date while others are on a fixed weekday.

Date	Description	Comment
25.12.	1. Christmas Day	
26.12.	2. Christmas Day	
01.01.	New Year	
06.01.*	Epiphany	
Friday	Good Friday	Easter Sunday-2
Monday	Easter Monday	Easter Sunday+1
Thursday	Ascension Day	Easter Sunday+39
Monday	Whitmonday	Easter Sunday+50
Thursday*	Corpus Christi	Easter Sunday+60
01.05.	Labor Day	
03.10.	Day of German Unity	
01.11.*	All Hallows	

Table 4.4: Public holidays in Germany and Baden-Württemberg(*). The weekday changes if a date is stated. Public holidays that are aligned with Easter Sunday have a fixed weekday. Easter Sunday is the first Sunday after full moon in spring (22.03. - 25.04.).

Date	Description	Comment
24.12.	Christmas Eve	
31.12.	New Year's Eve	
7 weeks before Easter	(Thu) Women's Carnival Day (Fri) Carnival Friday (Sat) Carnival Saturday (Sun) Carnival Sunday (Mon) Carnival Monday (Tue) Carnival (Wed) Ash Wednesday	Carnival
Sunday	Easter Sunday	see Table 4.4
Sunday	Whitsunday	Easter Sunday+49

Table 4.5: Special days that are not a public holiday but are also subject to different demand patterns.

In Germany there are nine state-wide public holidays as well as additional public holidays that are set by each federal state, which are three for Baden-Württemberg (see Table 4.4). The

public holidays that are related to Easter are on a fixed weekday but can shift by up to one month over the years. The other public holidays have a fixed date but a changing weekday. So, not only the number of observations but also the comparability is limited. The same is true for the special days given in Table 4.5.

4.4.2 Feature Engineering

In general, we rely on the features outlined in Section 3.1 and Table 3.1 (excluding weather features). With respect to special days (see Tables 4.4 + 4.5), we can either include them in the model or replace and adjust the prediction in a post-processing step. As our goal is to build global models, we introduce specific features that are based on the proceeding of a domain expert who typically identifies reference days in previous years in order to predict the expected demand. Hence, we also compute features that are based on the history of each specific special day. The features cover the effect of the special day compared to a regular day, i.e., we measure the absolute and relative change of a special day compared to a rolling median of each weekday. This is necessary as the weekday might change, but the total demand might be comparable. In order to cover level changes, we also include the historic rolling median as a feature. We do not consider including the concrete value as a feature as this would put too much emphasis on it and causes the model to overfit, i.e., the future forecast accuracy would heavily depend on the feature. More precisely, we create the following features:

- **historic demand level:** The level is defined as the rolling seasonal median for a specific weekday and is an estimation for the day if it would not be a special day. Public holidays are always compared to Sundays while the other special days are compared to their actual weekday.
- **absolute change:** The absolute change on a special day compared to the level.
- **relative change:** The relative change on a special day compared to the level.
- **relative change (store class):** Additionally, we compute the average relative effect over all stores having the same store class. This effect is more robust as it is based on a larger data basis.

As we distinguish between public holidays and the remaining special days, we define a total of eight special day specific features. In order to obtain the values of the features for the forecast horizon, we apply a weighted rolling mean over the history such that the feature values do not only depend on the previous year (see Table 4.6). If a special day can fall on different weekdays, we consider the historic comparison with the weekday that is relevant within the forecast horizon.

Overall, we consider more than 250 features. The high dimensionality makes it necessary to increase the scope of the model to have more training data available and to reduce the risk of overfitting (e.g. curse of dimensionality). We apply a log transformation and subsequently

Year	Target	Level	Abs. Change	Rel. Change
2015 (train)	130	100	30	0.3
2016 (train)	150	125	25	0.2
2017 (test)	-	116.67	26.67	0.233

Table 4.6: For each special day, we compute the level (i.e. rolling seasonal median forecast) as well as the absolute and relative change between the level and the target (i.e. sales on the special day). In the test period, we compute a weighted rolling mean (weights: 2015: 1; 2016: 2) of historic observations in order to obtain the feature values for the absolute and relative change.

mean	sd	0.05	0.10	0.25	0.50	0.75	1.00
727.21	251.38	239	362	510	815	979	992

Table 4.7: Mean, standard deviation (sd) and quantiles of the number of observations per time series.

linearly scale the target variable and features directly obtained from it (i.e. autoregressive features) to the range [-0.5, 0.5] as this is beneficial for backpropagation (LeCun et al., 2012). When we transform the regression problem to a classification problem, we create a bin for each percentile or more frequently if the relative increase between neighboring bins exceeds 10%. In total, we end up with 124 bins as the data gets sparser for higher target values. We set the numerical value of a class to the mean of the interval endpoints. For the recurrent neural networks, we create short sequences covering the lags as provided to the direct approaches (lags: 2-7, 14) and only vary the dynamic features like sales and weekday. Hence, the same information is provided to all ML methods.

4.4.3 Experimental Setup

The evaluation aims to assess the performance of ML methods and provides a comparison with state-of-the-art time series methods. In particular, we investigate how well the models predict the sales for different types of special days. By reporting the empirical forecast performance, we also want to stress the importance of considering special days during the model building process. We outline the experimental setting including a brief description of the dataset, the setup of the ML models, and a description of the reference methods with adjustments for special days. The results of the evaluation are reported and discussed in Section 4.4.4.

4.4.3.1 Dataset & Setup of ML Methods

We use dataset v1 (see Section 3.2) for the evaluation. The dataset contains daily sales at the store-category level for 8 product categories in 141 stores, i.e., 1128 time series. The number of observations per time series varies as some stores are always closed on certain weekdays. Moreover, the available sales history of stores varies due to new openings. For over 90% of the time series, the sales history covers at least one year (see Table 4.7).

	Training Dataset		Test Dataset	
SD	**N**	**N [%]**	**N**	**N [%]**
0	568,653	83.88	107,427	75.48
1	36,463	5.38	10,369	7.29
2	15,651	2.31	6,869	4.83
3	16,004	2.36	4,461	3.13
4	41,192	6.08	13,200	9.27
Σ	677,963	100.00	142,326	100.00

Table 4.8: Number of observations of each special day type (SD), i.e., sales greater than zero, in the training dataset and the test dataset. The training dataset comprises data from 2014-10-01 to 2017-01-31 while the test set comprises data from 2017-02-01 to 2017-06-30.

We split the data into a training period and a test period (see Table 4.8). The training period comprises all observations from October 2014 to January 2017. The test period comprises 150 days between 2017-02-01 and 2017-06-30. This is the most interesting period of the year with respect to our motivation because most calendric special days fall in this time span. In fact, we classify 39 (26%) days in the test period as special days. This includes 15 (10%) days of type SD1, 7 (4.67%) days of type SD2, 5 (3.33%) days of type SD3, and 14 (9.22%) days of type SD4. The number of neighboring days (SD2 + SD3) is smaller than the number of special days (SD1) as they are only considered for public holidays. The distribution of day types is not exactly matched by the number of observations in the test dataset (see Table 4.8) as some stores are closed on public holidays.

With respect to the ML methods, we only rely on data from the training dataset to select the parameters and to train the models. Thus, we employ cross-validation within the training period as we also need validation data in order to select the best (hyper-)parameters and architectures of the models (see Tables 4.9 & 4.10). We also rely on the validation dataset to apply early stopping. For this purpose, we create 10 stratified samples based on the day type and product category whereby 80% of the full training dataset is used for training and the remaining 20% serve as validation data.

We select the best configuration for each method based on the forecast accuracy on the validation datasets of the 10 samples. In more detail, we compute the sum of the day type specific RMSEs for each sample, i.e., the ranking criteria is $RMSE_{SD0} + RMSE_{SD1} + RMSE_{SD2} + RMSE_{SD3} + RMSE_{SD4}$. Subsequently, we exclude the two best and the two worst results and compute the average RMSE over the remaining samples per model configuration. We pick the configuration that performs best on the validation datasets and train 40 models on 40 additional samples. This is necessary as especially neural network approaches require an ensemble of models in order to produce more reliable results. We employ the median ensemble operator to combine the predictions of the 50 trained models.

For the classification approaches, we evaluate *1-hot* encoding and pick the class having the highest probability (*max*) as well as the *median* of the predicted distribution. As we rely on an ensemble of models, we re-scale the sum of the probabilities to 1.0 after computing the median class probability over the samples and before we determine the predicted class, i.e.,

Parameter	Values
learning rate	0.001
batch size	128
early stopping patience	3
hidden layers	(50), (100), (200), (300), (400), (500), (50, 50), (100, 50), (100, 100), (150, 100), (150, 150), (200, 200), (300, 300), (50, 50, 50), (100, 50, 50), (100, 100, 50), (100, 100, 100), (200, 100, 50), (300, 200, 100)
activation functions (hidden layer)	relu, elu, LSTM-cell
activation functions (output layer)	linear (regression), softmax (classification)
loss	mean squared error (regression), categorical cross entropy (classification)

Table 4.9: Evaluated architectures and hyper-parameters for the neural network approaches (38 architectures per regression and classification). For the recurrent neural networks (LSTM), we considered only architectures with up to two layers (13 architectures).

Parameter	Values
num_iterations	10000
learning_rate	0.01
max_depth	-1
num_leaves	64, 128, **256**, 384, 512, 768
min_data_in_leaf	5, 10, 15, **20**
max_bin	100, 200, 400, 600, **800**
min_data_in_bin	1, **5**, 10, 15
early_stopping_rounds	10
boosting	gbdt
objective	regression_l2
metric	l2

Table 4.10: Parameter grid for LightGBM resulting in 480 settings. The selected parameters based on the performance on the validation dataset are highlighted in bold.

the class having the highest probability, and resolve the numeric value of the class. The classification approach is only evaluated for the ANNs as preliminary experiments with *LightGBM* did not terminate.

The selected neural network architectures are listed in Table 4.11, and the parameter configuration for the *LightGBM* (*LGBM*) models is shown in Table 4.10. With respect to the neural network architectures, we observe that the capacity of the hidden layers is smaller for the classification approach compared to the regression approach. An explanation for this is that the output layer has not just one node but 124 nodes, i.e., the number of predefined classes. The larger number of output nodes implies more trainable weights between the last hidden layer and the output layer. For all ML approaches, we train global models that forecast every time series.

Parameter	MLP-REG	MLP-CL	LSTM-REG	LSTM-CL
task	regression	classification	regression	classification
hidden layers	(300, 200, 100)	(100, 50, 50)	(300, 300)	(50, 50)
activation functions (hidden layer)	relu	elu	LSTM	LSTM

Table 4.11: Selected neural network architectures based on the performance on the validation dataset. The remaining parameters are given in Table 4.9.

To measure the accuracy of the predictions, we rely on the symmetric mean absolute percentage error (*SMAPE*), the seasonal mean absolute scaled error (*MASE*) (Hyndman and Koehler, 2006), the mean absolute error (*MAE*), and the root mean squared error (*RMSE*) as described in Section 2.3. However, the scaling factor of *MASE* is determined within the training set for each time series by only comparing the *seasonal naïve forecast* to the actual values for days of type SD0. Moreover, we only report relative error measures for better interpretability of performance gains and for reasons of confidentiality, e.g., $MAE_{rel} = \frac{MAE_{M2}}{MAE_{M1}}$ is the relative performance of method M2 compared to method M1 with respect to *MAE*.

4.4.3.2 Reference Methods

We compare the ML methods (see Section 2.2) to baseline methods and state-of-the-art time series models (see Section 2.1) to illustrate the competitiveness of the data-driven approaches. Additionally, we also introduce the method *S-Naïve-Std* which omits the sales on days of type SD1-SD3 and replaces them with the last observation on a regular day (SD0). Hence, the predictions of *S-Naïve-Std* for the days of type SD0 and SD4 are not by distorted by the sales on the previous special day.

We also evaluate the performance of the popular exponential smoothing model family (Hyndman et al., 2002, 2008). For the evaluation, we rely on the training period to select the model per time series. An identified model is used for the complete test period, but the model coefficients are updated in a rolling-origin fashion. As exponential smoothing is a univariate forecasting method that does not consider external effects, we replace the sales on special days with the rolling seasonal median (2.3) as we select and fit the models.

For univariate approaches, it is necessary to make adjustments for special days as sales deviate from regular days. The adjustment strategies of the forecasts are based on the special day features introduced in Section 4.4.2. In particular, we consider relative adjustments (*pct*, *pct-cl*) that allow a multiplicative effect as well as absolute adjustments (*abs*) that allow an additive effect. The strategies *pct* and *abs* are calculated per time series while *pct-cl* is the average effect over all stores of the respective store type for each product category. All adjustment values are calculated on the training period by comparing the sales on the special day to the rolling median of the weekday that has to be forecast. However, the reference weekday for public holidays is Sunday. Beside the adjustment strategies, we also evaluate the method *ETS [Sun]* where the forecast on public holidays is replaced by the last prediction for a Sunday.

Moreover, we evaluate a multiple regression model (*LIN-REG*) using LASSO regression (Tibshirani, 1996) which is inspired by the *ADL-own* model proposed by Ma et al. (2016). The model incorporates the same information that is also provided to the ML models which includes log-transformed lagged sales and *S-Median*, a coupon period indicator, binary dummy variables for the different day types and school holidays as well as the special day features. We employ cross-validation within the training period to determine the value for the

regularization hyper-parameter. The model *LIN-REG* is an alternative to the other baseline approaches as it does not require adjustments in a post-processing step. The linear regression models are fitted per time series as pooling did not improve the results.

4.4.4 Results & Discussion

In the first part of the evaluation (see Section 4.4.4.1), we analyze single-step predictions that are most important with respect to order decision optimization. The second part is concerned with multi-step forecasts (see Section 4.4.4.2) as longer planning horizons (e.g. three weeks) are also of interest for operational decisions.

4.4.4.1 Single-step Forecasts

We focus on the performance of the second forecasting step as this is the most crucial prediction when it comes to ordering decisions in an agile supply chain that is typical for bakeries. Point-of-sales data of the previous day is often only available after the planning and production for the following day starts. Thus, we specify the input features of the ML models such that they directly predict the second step by excluding lagged sales data of the previous day. We highlight the challenge of predicting sales on special days by elaborating on the results of the baseline methods (see Section 4.4.4.1.1) before we provide the comparison of the ML methods (see Sections 4.4.4.1.2 + 4.4.4.1.3).

4.4.4.1.1 Special Day Forecasting Challenge

The challenge of forecasting demand on special days is depicted in Figure 4.1. While the error level is rather stable on regular days, this is not the case for special days (SD1) and their neighboring days (SD2+SD3). Hence, there is a need for designing models that provide more accurate forecasts for special days.

The analysis of the results of the baseline and reference methods, presented in Table 4.12, leads to three central findings: First, all *ETS*-based approaches clearly outperform the simple baseline approaches for all day types. Hence, there are indeed patterns encoded in the time series that justify the application of sophisticated prediction models. Second, the forecast errors on special days (SD1-SD3) are higher than on regular days, which suggests that special days are more difficult to forecast as the demand patterns differ. Moreover, the severity of the forecast errors on special days is hidden on aggregated key figures. By splitting days in groups, it is possible to identify the origin of the error and improve the prediction model. Third, the adjustment strategies for special days significantly improve the accuracy of all methods, which indicates the existence of underlying demand patterns for such days. This is also supported by the fact that the model *LIN-REG* has lower errors on special days than the models with unadjusted forecasts.

With respect to the errors on the different types of special days, we notice that SD1 has the highest errors which are without adjustments more than 60% (300%) higher than on regular days with respect to SMAPE (MAE). However, also the neighboring days (SD2

SD	Method	SMAPE	MASE	MAE	RMSE	Rank
all	ETS	1.06	1.15	1.28	1.90	6.15
	ETS (abs)	1.03	1.05	**1.08**	**1.10**	6.05
	ETS (pct)	1.02	1.06	1.08	1.10	6.06
	ETS (pct-cl)	1.02	**1.05**	**1.08**	**1.11**	6.04
	ETS [Sun]	1.03	1.08	1.14	1.24	6.10
	LIN-REG	**1.00**	**1.06**	1.17	1.32	**6.02**
	S-Median	1.18	1.33	1.51	2.04	6.94
	S-Median (abs)	1.14	1.23	1.31	1.33	6.83
	S-Median (pct)	1.14	1.23	1.31	1.34	6.83
	S-Median (pct-cl)	1.14	1.23	1.31	1.35	6.81
	S-Naïve	1.28	1.47	1.75	2.62	7.18
	S-Naïve-Std	1.21	1.31	1.48	2.05	6.98
0	ETS	1.00	**1.00**	**1.00**	**1.00**	2.80
	LIN-REG	**0.98**	**1.01**	1.06	1.12	**2.79**
	S-Median	1.12	1.19	1.22	1.20	3.12
	S-Naïve	1.17	1.22	1.26	1.31	3.17
	S-Naïve-Std	1.14	1.16	1.18	1.22	3.12
1	ETS	1.63	2.49	4.06	5.88	6.46
	ETS (abs)	1.20	1.37	**1.67**	**1.73**	5.61
	ETS (pct)	1.17	1.41	1.72	1.77	5.70
	ETS (pct-cl)	**1.13**	**1.36**	**1.73**	**1.85**	**5.52**
	ETS [Sun]	1.22	1.61	2.20	2.53	5.79
	LIN-REG	1.15	1.46	2.11	2.65	5.55
	S-Median	1.71	2.60	4.23	5.90	6.80
	S-Median (abs)	1.26	1.46	1.84	1.91	5.92
	S-Median (pct)	1.23	1.48	1.85	1.93	5.95
	S-Median (pct-cl)	1.19	1.44	1.86	2.04	5.76
	S-Naïve	1.75	2.61	4.24	6.00	6.94
2	ETS	1.28	1.68	1.66	1.68	5.50
	ETS (abs)	1.13	1.46	**1.36**	**1.36**	5.01
	ETS (pct)	1.13	1.48	1.37	**1.36**	5.03
	ETS (pct-cl)	1.11	1.46	**1.35**	**1.33**	4.97
	LIN-REG	**1.07**	**1.40**	**1.46**	**1.55**	**4.92**
	S-Median	1.41	1.87	1.83	1.78	6.11
	S-Median (abs)	1.23	1.66	1.56	1.46	5.67
	S-Median (pct)	1.24	1.68	1.57	1.47	5.72
	S-Median (pct-cl)	1.23	1.66	1.56	1.45	5.70
	S-Naïve	1.52	1.96	1.89	1.90	6.36
3	ETS	1.06	1.32	1.74	1.71	5.53
	ETS (abs)	1.13	1.26	1.42	1.32	5.35
	ETS (pct)	1.11	1.29	1.42	1.32	5.44
	ETS (pct-cl)	1.09	1.25	1.42	1.32	5.30
	LIN-REG	**0.98**	**1.13**	**1.33**	**1.27**	**4.82**
	S-Median	1.14	1.43	1.87	1.87	5.91
	S-Median (abs)	1.18	1.30	1.48	1.39	5.52
	S-Median (pct)	1.14	1.31	1.47	1.39	5.53
	S-Median (pct-cl)	1.11	1.27	1.47	1.41	5.35
	S-Naïve	1.22	1.55	2.10	2.09	6.25
4	ETS	**1.01**	**0.94**	**1.01**	**1.00**	**2.69**
	LIN-REG	1.02	0.96	1.09	1.08	2.71
	S-Median	1.14	1.17	1.42	1.68	2.95
	S-Naïve	1.67	2.32	3.53	5.34	3.55
	S-Naïve-Std	1.18	1.14	1.26	1.20	3.10

Table 4.12: Results of the evaluation of the baseline and reference methods. The reported forecast errors are relative to the performance of the method ETS for SD0. For each day type, we underline the lowest error and mark results that are not significantly different to the method having the lowest at 0.05 significance level according to the Wilcoxon signed-rank test in bold face. We also provide the average rank based on the absolute error for every method per day type.

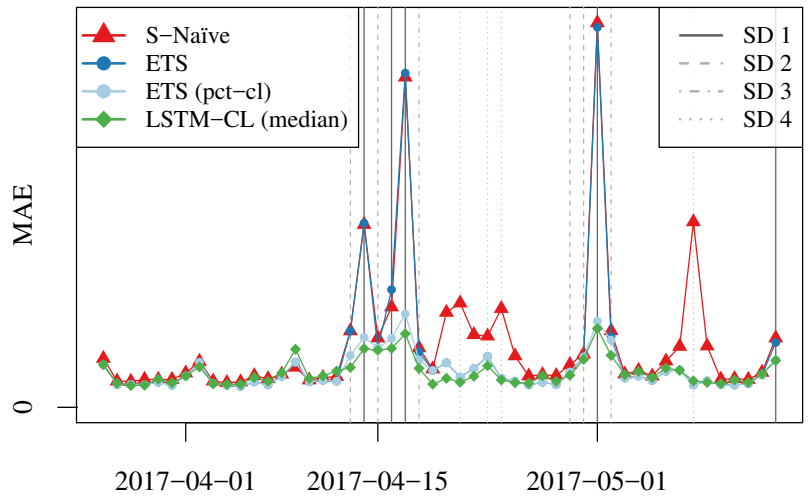


Figure 4.1: The MAE per day from 2017-03-26 to 2017-05-14 (7 weeks). The different day types are highlighted with vertical lines. The error level is quite constant on regular days while dramatic error peaks are observable on special days if they are not considered by the model. The scale of the y-axis is not provided for reasons of confidentiality.

and SD3) have approximately 25% (SMAPE) higher errors than normal days. In contrast, SD0 and SD4 yield rather comparable errors with exception of the method *S-Naïve* whose errors for SD4 are closer to the error level of SD1. This supports the assumption that SD4 is suitable as a sanity check in order to test if forecasts are affected by previous special days, which is possible for autoregressive models. Hence, for univariate methods, it is important to preprocess the observations on special days and make the required adjustment in a post-processing step. To some degree, preprocessing is also beneficial for sales on days of type SD2 and SD3 as the errors of *S-Naïve-Std* for SD0 are slightly lower than the errors of *S-Naïve*. *S-Median* is more robust to special days as its performance for SD4 is only slightly worse than for SD0 even though special days are not excluded from the sales history.

In general, the adjustment strategies work well for all day types and methods as forecast errors are significantly reduced. The strategies *abs* and *pct* should only yield different forecasts if the demand level changed between the training period and the test period. While *pct* leads to slightly lower errors, the differences to *abs* are not statistically significant. This observation suggests that the demand level is rather stable for most time series. More surprising is the good performance of *pct-cl* which seems to be the most reliable adjustment strategy but it is also not statistically significantly different from the other strategies. A possible explanation for this is that the special day effect is comparable within a group of stores. Thus, computing the average effect over multiple stores makes *pct-cl* slightly more robust. For SD1, we also notice that *ETS [Sun]* is much more accurate than *ETS* which supports the assumption that at least public holidays have much in common with Sundays.

However, while the adjustments significantly reduce the errors for SD1 and SD2, this only holds to a limited degree for SD3. While the error levels of SD3 are comparable with SD2, there are no significant differences among all *ETS*-based approaches and *S-Median-*

based approaches with adjustments. It is even the case that the standard *ETS* models have the lowest SMAPE. However, scale-dependent errors of unadjusted forecasts are still higher than the errors of adjusted forecasts. Thus, adjustment strategies still provide practically relevant improvements for those days while the relative benefits are reduced compared to SD1.

The method *LIN-REG* does not rely on adjustments as the special days are explicitly incorporated which works well for the neighboring days of public holidays while it is worse on main special days (SD1). On days of type SD0 and SD4, the differences between *LIN-REG* and *ETS*-based models are mostly noticeable for the error measures RMSE and MAE while SMAPE and MASE are rather comparable. Nevertheless, special day patterns can be learned by a linear model but relying on adjustments in a post-processing step seems to be the better option in some cases. *ETS* is a suitable model for the regular demand patterns that follow a very strong weekly seasonality while other external influences with the exception of special days are rather negligible or hard to separate from the noise of daily data. Hence, it is reasonable that *LIN-REG* struggles to outperform *ETS* (with adjustments) even though it incorporates more information.

Overall, the relative advantage of *ETS*-based models compared to baseline methods slightly diminishes for special days as all methods heavily depend on the adjustments which are identical for all methods. More fine-grained adjustments might be necessary to further reduce the forecast errors. As such adjustments rules and values are hard to specify manually, we rely on ML methods that are able to learn them from data.

4.4.4.1.2 Baseline vs. Machine Learning Methods

We compare the performance of the ML methods to the best reference methods, i.e., *LIN-REG* and *ETS (pct-cl)* (see Table 4.13). Overall, the ML methods statistically significantly outperform the reference methods by a noticeable error margin and also have lower average ranks (see Figure 4.2). The performance gains are the largest for the neighboring days of public holidays (SD2+SD3) where the errors drop by more than 10%. Those are the days where the adjustment strategies did not provide very much benefit compared to unadjusted forecasts, and the performance of *LIN-REG* suggested that improvements are possible. An explanation for this is that the neighboring days of public holidays share similarities among the different public holidays. Hence, the ML methods are able to detect and learn more precise patterns that translate to lower forecast errors. Due to those improvements, the forecast errors for SD2 and SD3 are noticeably closer to the performance on regular days. For SD1, the differences are not as significant. The approaches based on ANNs have a 5% lower SMAPE but relative improvements with respect to scale-depend error measures are rather limited. A possible explanation for this is that the demand on special days is very volatile as the weekday or the time of the year varies over the years. This makes it generally hard to estimate the demand for such days. Nevertheless, the ML approaches are still the preferred option as they provide more accurate predictions.

SD	Method	SMAPE	MASE	MAE	RMSE	Rank
all	LIN-REG	0.98	1.01	1.08	1.18	5.21
	ETS (pct-cl)	1.00	1.00	1.00	1.00	5.18
	LSTM-CL (median)	**0.91**	**0.91**	0.96	0.97	**4.72**
	MLP-CL (median)	0.92	0.94	0.98	0.98	4.85
	MLP-REG	0.94	0.95	1.01	1.02	4.93
	LSTM-CL (max)	0.94	0.94	0.98	**0.99**	4.97
	LSTM-REG	0.94	0.94	0.98	1.02	4.94
	MLP-CL (max)	0.96	0.97	1.00	1.01	5.08
	LGBM	1.01	0.98	**0.95**	**0.93**	5.12
0	LIN-REG	0.98	1.01	1.06	1.12	5.20
	ETS (pct-cl)	1.00	1.00	1.00	1.00	5.16
	LSTM-CL (median)	**0.91**	**0.92**	0.97	0.98	**4.74**
	MLP-CL (median)	0.93	0.94	0.98	**0.98**	4.86
	MLP-REG	0.94	0.95	0.99	1.00	4.92
	LSTM-CL (max)	0.94	0.95	**1.00**	1.01	5.00
	LSTM-REG	0.94	0.94	0.98	**0.97**	4.95
	MLP-CL (max)	0.96	0.97	1.01	1.01	5.09
	LGBM	1.00	0.97	**0.95**	**0.92**	5.08
1	LIN-REG	1.02	1.08	1.22	1.43	5.26
	ETS (pct-cl)	1.00	1.00	1.00	1.00	5.17
	LSTM-CL (median)	**0.90**	**0.91**	**0.94**	**0.96**	**4.61**
	MLP-CL (median)	0.92	0.96	1.01	1.02	4.92
	MLP-REG	0.94	1.00	1.06	1.08	4.95
	LSTM-CL (max)	0.94	0.94	0.95	0.97	4.86
	LSTM-REG	0.96	1.02	1.07	1.19	4.92
	MLP-CL (max)	0.96	0.99	1.03	1.05	5.12
	LGBM	1.08	1.00	0.98	0.98	5.19
2	LIN-REG	0.96	0.96	1.09	1.17	5.25
	ETS (pct-cl)	1.00	1.00	1.00	1.00	5.35
	LSTM-CL (median)	**0.86**	**0.84**	**0.93**	**0.95**	**4.54**
	MLP-CL (median)	0.88	0.86	0.94	0.96	4.69
	MLP-REG	0.93	0.95	1.09	1.13	5.14
	LSTM-CL (max)	0.90	0.88	0.96	1.00	4.81
	LSTM-REG	0.88	0.87	0.97	1.00	4.79
	MLP-CL (max)	0.93	0.90	0.97	1.00	4.98
	LGBM	1.03	0.99	1.00	0.97	5.44
3	LIN-REG	0.90	0.91	0.94	0.96	5.21
	ETS (pct-cl)	1.00	1.00	1.00	1.00	5.61
	LSTM-CL (median)	**0.82**	**0.82**	**0.82**	**0.84**	**4.59**
	MLP-CL (median)	0.84	0.85	0.91	0.95	4.79
	MLP-REG	0.86	0.86	0.92	0.97	4.95
	LSTM-CL (max)	0.86	0.84	0.84	0.85	4.83
	LSTM-REG	0.84	0.84	**0.85**	**0.87**	4.74
	MLP-CL (max)	0.88	0.87	0.93	0.97	5.04
	LGBM	0.95	0.91	0.89	0.90	5.24
4	LIN-REG	1.01	1.03	1.08	1.08	5.23
	ETS (pct-cl)	1.00	1.00	**1.00**	**1.00**	5.12
	LSTM-CL (median)	**0.93**	**0.94**	0.98	0.99	**4.73**
	MLP-CL (median)	0.94	**0.94**	0.97	0.95	4.80
	MLP-REG	0.96	0.96	1.01	1.01	4.91
	LSTM-CL (max)	0.96	0.96	1.00	1.01	4.97
	LSTM-REG	0.99	0.97	0.99	1.00	5.03
	MLP-CL (max)	0.98	0.97	0.99	0.98	5.02
	LGBM	1.06	1.04	**0.97**	**0.91**	5.20

Table 4.13: Results of the evaluation of the ML methods. The reported forecast errors are relative to the performance of the method ETS (pct-cl) for each day type. For each day type, we underline the lowest error and mark results that are not significantly different to the method having the lowest at 0.05 significance level according to the Wilcoxon signed-rank test in bold face. We also provide the average rank based on the absolute error for every method per day type.

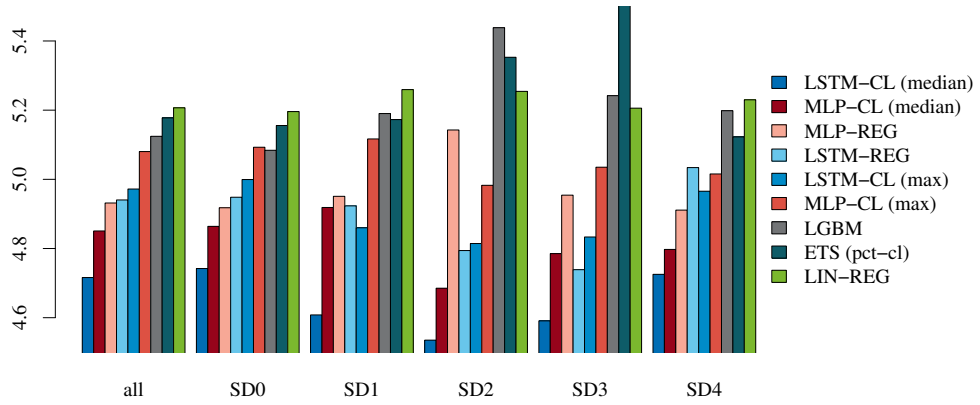


Figure 4.2: Comparison of the ML methods: The average rank of the evaluated methods for the different day types based on the absolute deviation.

4.4.4.1.3 Regression vs. Classification

A comparison of the ML methods (see Table 4.13) reveals that the methods based on ANNs outperform *LGBM* according to SMAPE and MASE. However, *LGBM* is especially for regular days (SD0) the most accurate method with respect to scale-depend error measures (RMSE, MAE). The reason for this is that *LGBM* provides more accurate predictions for larger values. The preprocessing of the target variables of the ANNs might negatively affect the forecast accuracy for larger target values. For regression-based approaches, we perform a log transformation followed by linearly scaling the target values to the range $[-0.5, 0.5]$. For classification-based approaches, we perform binning that benefits higher absolute errors as the bins comprise larger intervals as the target values grow. However, preliminary experiments with other transformations led to worse results. In contrast, transforming the target values for the *LGBM* models did not lead to performance gains. *LGBM* implicitly creates data bins during the learning phase while constructing the decision trees, which apparently leads to a better grouping of the target values.

With respect to the ANN-based methods, we notice that recurrent ANNs (LSTM) outperform their feed-forward (MLP) counterparts. Hence, it seems to be beneficial to process the time series data in a sequential fashion. Moreover, transforming the regression problem to a classification problem also seems to be beneficial even though the range and the number of possible target values is limited by the number of classes and their underlying values. However, this does not seem to be a problem in the present use case because the classification approaches are at least as good as their regression-based counterparts.

The prediction of a classification model is a probability distribution over the target classes which can be interpreted as a density forecast. In order to select the predicted values of the classification, we investigate two approaches: By selecting the class with the highest probability (*CL (max)*), we pick the mode of the distribution. However, it is also possible to pick the median (*CL (median)*) at no additional costs which might be better suited for the used performance measures. Generally, we observe that the accuracy of *CL (max)* is rather comparable or only slightly better than the accuracy of the standard regression approach.

Taking the median of a probability distribution of the target classes leads to significantly better results. We observe the largest performance gain for feed-forward ANNs as the recurrent ANNs are already on a high accuracy level. Both methods that employ *CL (median)* are mostly more accurate than any other method. An explanation for this is that predicting and exploiting the probability distribution over the target classes provides an additional level to address the uncertainty associated with a prediction. Gneiting (2011) and Kolassa (2016) highlight that different functionals optimize different loss functions, e.g., the mean squared error (mean absolute error) is optimized by the mean (median) of the probability distribution which leads to vastly different outcomes for asymmetric distributions.

In fact, in more than 50% of the cases, the predictions between *CL (median)* and *CL (max)* are different (see Table 4.14). For nearly 69% of the forecasts, the selected classes are neighboring classes. The relative change of the predicted values is only 0.05% for roughly 50% of the forecasts. The direction of changes is only slightly favored to an increase of the forecasts using *CL (median)*. Hence, we conclude that selecting a different class does not serve as bias correction. The comparison also reveals that *CL (median)* is for roughly 55% of the forecasts more accurate than *CL (max)* which translates to a reduction of the forecast errors by 6% to 9% (see Table 4.15). The aforementioned statements hold for *MLP-CL* and *LSTM-CL*.

	Direction			**Class Steps**					**Relative Change**				
Method	$<$	$==$	$>$	**0**	**25**	**50**	**75**	**100**	**0**	**25**	**50**	**75**	**100**
FNN-CL	0.24	0.46	0.31	-12	-1	-1	1	14	-0.49	-0.05	0.05	0.12	10
LSTM-CL	0.25	0.47	0.28	-14	-1	1	1	17	-0.42	-0.05	0.05	0.12	9

Table 4.14: Comparison of *CL (median)* and *CL (max)*. The table shows the direction of the change and the degree of change with respect to the target class and target value relative to *CL (max)* if the predictions of both approaches do not match. We provide the quantiles for the latter two key figures.

Method	**SMAPE**	**MASE**	**MAE**	**RMSE**	**Rank**
MLP-CL (max)	1.00	1.00	1.00	1.00	1.55 (0.49)
MLP-CL (median)	**0.94**	**0.94**	**0.93**	**0.91**	**1.45 (0.49)**
LSTM-CL (max)	1.00	1.00	1.00	1.00	1.56 (0.49)
LSTM-CL (median)	**0.94**	**0.94**	**0.93**	**0.92**	**1.44 (0.49)**

Table 4.15: Accuracy of the classification approaches when *CL (max)* and *CL (median)* provide different results. *CL (median)* has lower forecast errors and a lower average rank compared to *CL (max)*. We also provide the average rank and in brackets its standard deviation.

4.4.4.1.4 Effect of Re-training

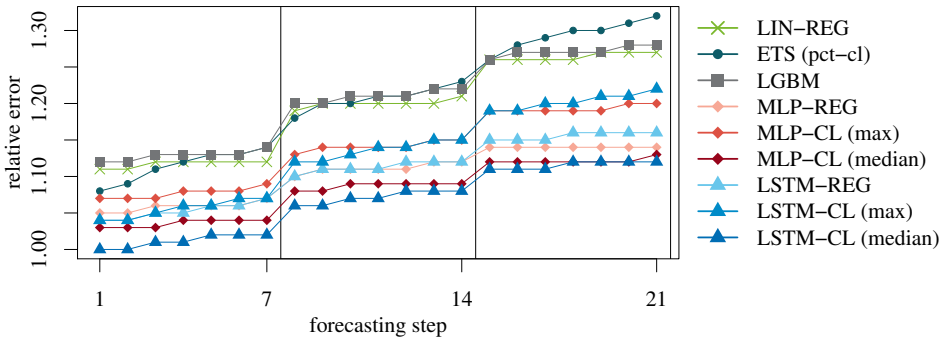
In forecasting literature, it is common practice to perform a rolling-origin evaluation as this typically leads to better forecasts. Rolling-origin evaluation means that at least the model coefficients are re-fitted for every forecast origin. In this study, we only re-fitted the *ETS* models and *LIN-REG* for each of the $142,326$ forecasts (see Table 4.8). This is already

challenging in a productive setting as the time frame to calculate and deliver daily forecasts is limited to only a few hours. For the evaluated ML methods, it is certainly not possible to train them on a daily basis. However, we provide empirical evidence that ML models that are not re-trained during the test period outperform the reference methods (see Section 4.4.4.1.2). Applying the trained models is quite efficient, which makes them suitable for productive usage. Nevertheless, we also investigate if more frequent training leads to improved results as the test period is relatively long and more regular re-training is possible. Having an additional week of data available for training only increases the dataset by 0.98% which accumulates to approximately 20% over the 150 days of the test period.

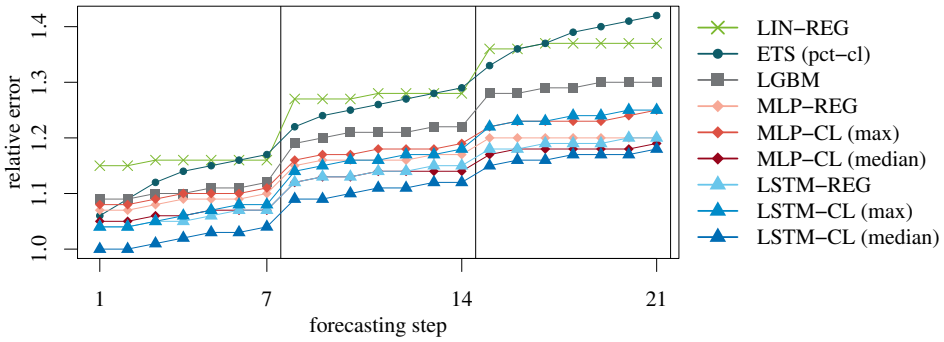
Thus, we bi-weekly re-train the ML models from scratch, i.e., using random initial weights or no tree, with the same hyper-parameters as before. The only exception are the *LSTM* models whose weights of the previously trained models are only fine-tuned with the extended datasets in order to save training time. The drawback of this approach is that it is more likely to overfit the "old" training data. The trained models are then used for a limited number of upcoming weeks before they are replaced with re-fitted models. We report the accuracies for various fitting frequencies in Table 4.16.

In general, we observe that bi-weekly re-training leads to the best results while no re-training, i.e., models are used for 22 weeks, leads to the highest forecast errors. Due to bi-weekly re-training, the errors drop by 2-3% compared to no re-training depending on the error measure. The relative improvements are larger for the scale-dependent error measures (RMSE, MAE) and MASE than for SMAPE. Re-training frequencies between 10 and 4 weeks offer hardly (statistically) significant differences and the additional performance gains due to bi-weekly re-training are only around one percentage point. We also notice that the ANNs tend to benefit more from re-training than *LGBM* even though they already outperformed *LGBM* with no re-training. The relative error decrease of *LGBM* is not larger than 1.7% for any re-training frequency. Thus, in particular ANNs have the capability to adapt and incorporate additional observations which helps to reduce the larger absolute forecast errors. A reason for this is that the number of available observations decreases as the demand grows, which makes additional data valuable.

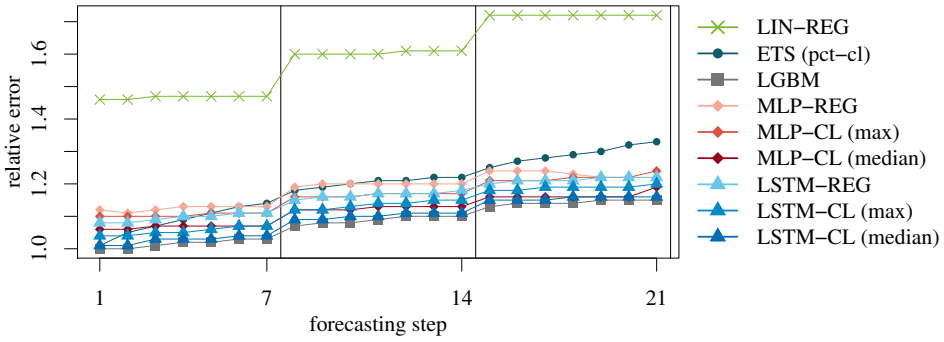
To summarize, despite the fact that the models are trained on a very large dataset, forecast accuracy improvements are possible if the models are more frequently re-trained. Hence, in a productive setting it has to be evaluated how often it is feasible to re-train the models. There are different opportunities to limit the computational costs. As we rely on an ensemble of 50 models, it is possible to continuously replace a subset of the models. It is also possible to only fine-tune the weights of the ANNs that are only trained on rather old data. However, even if re-training is not workable in a productive setting, the ML methods are still very competitive over a long time period as shown in Section 4.4.4.1.2.



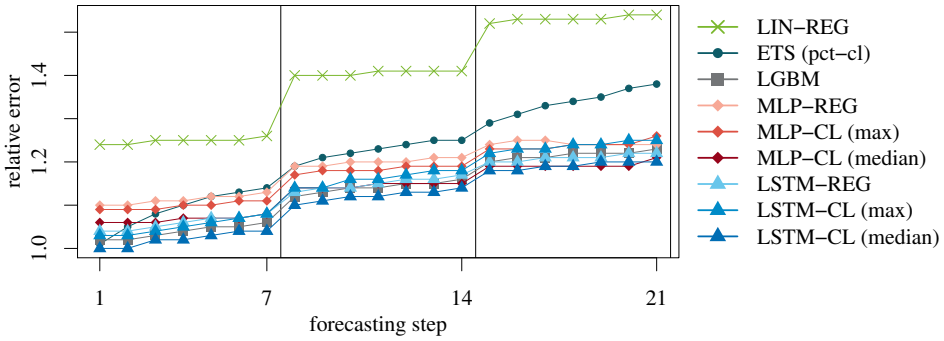
(a) SMAPE



(b) MASE



(c) RMSE



(d) MAE

Figure 4.3: The multi-step forecast errors for the next 21 days. The provided errors are relative to the best-performing method at step 1.

		Model Re-training Frequency in Weeks					
Measure	**Method**	**22**	**10**	**8**	**6**	**4**	**2**
SMAPE	LGBM	1.000	0.999	1.000	0.996	0.997	**0.993**
	LSTM-CL (max)	1.000	0.995	0.994	0.996	0.991	**0.988**
	LSTM-CL (median)	1.000	0.994	0.992	0.992	0.989	**0.983**
	LSTM-REG	1.000	0.992	0.992	0.990	0.988	**0.980**
	MLP-CL (max)	1.000	0.991	0.992	0.993	0.992	**0.984**
	MLP-CL (median)	1.000	0.989	0.991	0.992	0.989	**0.981**
	MLP-REG	1.000	0.993	0.994	0.994	0.994	**0.987**
MAE	LGBM	1.000	0.995	0.996	0.988	0.993	**0.983**
	LSTM-CL (max)	1.000	0.991	0.988	0.983	0.978	**0.967**
	LSTM-CL (median)	1.000	0.991	0.987	0.986	0.981	**0.969**
	LSTM-REG	1.000	0.988	0.996	0.981	0.986	**0.965**
	MLP-CL (max)	1.000	0.986	0.992	0.991	0.992	**0.971**
	MLP-CL (median)	1.000	0.987	0.993	0.995	0.994	**0.972**
	MLP-REG	1.000	0.989	1.001	0.994	1.001	**0.983**
MASE	LGBM	1.000	0.998	1.002	0.995	0.999	**0.993**
	LSTM-CL (max)	1.000	0.993	0.993	0.991	0.987	**0.980**
	LSTM-CL (median)	1.000	0.993	0.992	0.990	0.988	**0.978**
	LSTM-REG	1.000	0.991	0.992	0.986	0.985	**0.971**
	MLP-CL (max)	1.000	0.988	0.991	0.990	0.988	**0.976**
	MLP-CL (median)	1.000	0.986	0.990	0.991	0.988	**0.973**
	MLP-REG	1.000	0.990	0.994	0.993	0.994	**0.983**
RMSE	LGBM	1.000	0.996	0.995	0.988	0.993	**0.983**
	LSTM-CL (max)	1.000	0.991	0.982	0.978	0.972	**0.965**
	LSTM-CL (median)	1.000	0.991	0.983	0.985	0.978	**0.968**
	LSTM-REG	1.000	0.984	0.989	0.977	0.983	**0.965**
	MLP-CL (max)	1.000	0.986	0.989	0.992	0.989	**0.966**
	MLP-CL (median)	1.000	0.992	0.994	1.002	0.998	**0.974**
	MLP-REG	1.000	0.987	1.009	0.995	1.009	**0.982**

Table 4.16: The effect of model re-fitting during the test phase for different frequencies. The relative error compared to no re-training, i.e., re-fitting every 22 weeks, is given for each method. For each method, we underline the lowest error and mark frequencies that are not significantly different at 0.05 significance level according to the Wilcoxon signed-rank test in bold face.

4.4.4.2 Multi-step Forecasts

The previous part of the evaluation was concerned with single-step predictions. However, longer planning horizons can also be a requirement for certain operational decisions. For example, public holidays are partially planned in advance. Multi-step predictions also allow us to check if the full demand patterns of special days are actually learned as multiple special days fall within the forecast horizon. We compute forecasts for 21 days (3 weeks) in order to evaluate if certain methods perform better for shorter or longer horizons. We excluded the predictions for the first 20 days of the test period in order to have 21 predictions for each observation which enables the comparison of the different forecasting steps. The resulting test set comprises $122,730$ observations per step. For the ML methods, we re-use the trained models from the experiments in Section 4.4.4.1.2 and apply them iteratively. The results of the evaluation are presented in Figure 4.3.

We observe that the ordering of the methods is stable for the different forecasting steps. The only exception are *ETS*-based approaches whose errors increase more quickly than the errors of the ML methods. *ETS* is partially competitive for the first three steps and for scale-

dependent error metrics but the performance gap to the ML approaches increases with the forecast horizon. For the ML methods, we also notice that the results are rather stable within a season (i.e. a week) which matches the strong weekly seasonality of the time series. Stable forecast errors within a season are not only observable for regular days but also for special days. Hence, it can be assumed that the special day patterns are correctly represented by the models, which makes post-processing obsolete. Finally, the conclusions drawn in Section 4.4.4.1.2 are not only valid for single-step predictions but do also hold for the other forecasting steps. This makes it reasonable to initially optimize a model for single-step predictions before options for multi-step forecasts are explored.

4.5 Application: Store-Article Level

In this section, we study forecasting at the store-article level, which is of interest as ordering decisions are ultimately made at this level. We investigate whether ML methods are a viable alternative to time series models and identify criteria that influence their performance. To this end, we assess the effect of the three levels of data usage on the performance:

- What is the value of explanatory feature data?
- What is the impact of the scope of the model, e.g., learning across stores and products?
- How does the length of the time series history (sample size) influence the performance?

Moreover, we compare regression and various other approaches in the context of the formulation of the forecasting problem as a classification task in order to address the following questions:

- Is the problem formulation as a classification problem instead of a regression problem a viable alternative?
- Which classification approach works best for demand forecasting?

We outline the experimental setup and considered methods in Section 4.5.1 and present the results in Section 4.5.2.

4.5.1 Experimental Setup

The evaluation in this section is based on dataset v2 (see Section 3.2) and additional explanatory feature data (see Table 3.1). The dataset contains daily sales over 88 weeks (1.7 years) at the store-article level for 11 products in 5 stores, i.e., 55 time series which contain up to 528 observations. All stores are closed on Sundays and public holidays, and the contained articles are not strongly affected by special days. Moreover, the size of the dataset also allows a more exhaustive evaluation of different aspects that influence the forecasting performance when ML methods are employed.

We split the dataset into a training set containing up to 63 weeks and a test set containing the remaining 25 weeks (see Table 4.17). We perform a rolling 1-step-ahead prediction evaluation on the test set to assess the performance of the methods. We fit the models and distribution parameters every 10 days on a rolling training dataset with a constant size. Due to computational constraints, we fit the parameters of the ANNs every 50 days only. Alternatively, we also do not re-fit the models during the test phase in order to assess the robustness of the models.

Sample	**1.0**	**0.8**	**0.6**	**0.4**	**0.2**	**0.1**
train length (days)	378	300	228	150	78	36
test length (days)	150	150	150	150	150	150

Table 4.17: Training & test periods for different sample sizes.

To evaluate the effect of the amount of available data, we vary the three levels: sample size (demand history), features, and model scope. Hence, we use different sample sizes for the training set. The full training set (sample size 1.0) covers 63 weeks, while the smallest training set (sample size 0.1) contains only 6 weeks (see Table 5.1). Moreover, we consider different feature sets for the ML methods:

- **Feature set: v1** comprises only information contained in the time series, i.e., autoregressive features and features describing the seasonality.
- **Feature set: v2** contains the same features as the feature set v1 and is additionally enhanced with explanatory information (see Table 3.1).

Finally, we vary the scope of the model in order to assess the effect of training with data from multiple time series:

- **Scope: ts** indicates that we fit a model for each time series.
- **Scope: all** indicates that we fit a model with data from every time series.

In order to validate the performance of the ML methods, we consider the reference methods *S-Naïve*, *S-Median* (last four observations), *ETS*, and *S-ARIMA* (see Section 2.1). With respect to the ML methods, we evaluate ANNs (ANN-MLP REG, ANN-LSTM REG) and gradient boosted regression trees (DT-LGBM) (see Section 2.2) for regression tasks. For the ANNs, we also discretize the output and transform the problem into a classification problem. A bin is created for each percentile or more frequently if the range or the relative increase from the lower bound to the upper bound exceeds 10%. The value of each bin is the average of the interval endpoints. We denote the 1-hot encoding *CL-1Hot* and the ordinal encoding *CL-Ord.* followed by the value that is taken from the predicted distribution (e.g. median). Each time series is linearly scaled to the range [0, 0.75] and the hyper-parameters are optimized by a random search (Bergstra and Bengio, 2012) using cross-validation on the training set. We always train an ensemble of 50 models that are combined with the median ensemble operator.

4.5.2 Results & Discussion

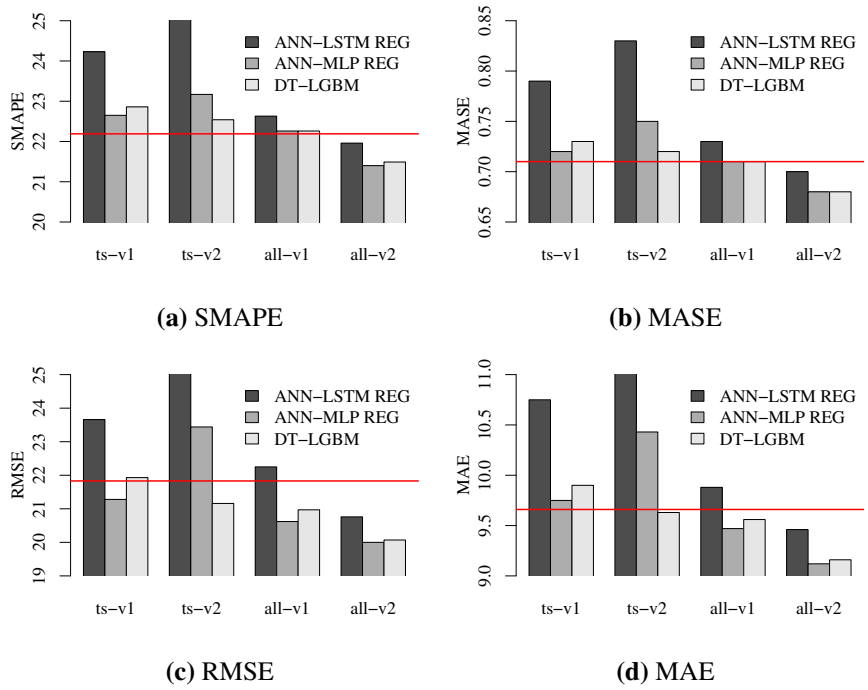
The presentation of the results falls in multiple parts. First, we generally compare the results of all methods. Subsequently, we take a closer look at data usage of the ML methods and the effect of model re-training during the test phase. Finally, we analyze the different approaches for the formulation of the classification problem.

Scope	Features	Method	SMAPE	MASE	MAE	RMSE
ts	-	ETS	22.19	0.71	9.66	21.83
		S-ARIMA	22.87	0.73	9.87	21.40
		S-Median	**23.89**	0.76	10.21	22.48
		S-Naïve	28.71	0.92	12.56	27.80
	v1	ANN-LSTM REG	24.23	0.79	10.75	23.66
		ANN-MLP REG	22.65	0.72	9.75	21.28
		DT-LGBM	22.86	0.73	9.90	21.93
	v2	ANN-LSTM REG	25.12	0.83	12.04	28.38
		ANN-MLP REG	23.17	0.75	10.43	23.44
		DT-LGBM	22.54	0.72	9.63	21.16
all	v1	ANN-LSTM CL-1Hot (median)	22.64	0.72	9.80	21.88
		ANN-LSTM CL-Ord. (median)	22.57	0.72	9.88	22.15
		ANN-LSTM REG	22.63	0.73	9.88	22.25
		ANN-MLP CL-1Hot (median)	22.25	0.71	9.46	20.70
		ANN-MLP CL-Ord. (median)	22.14	0.70	9.43	20.70
		ANN-MLP REG	22.26	0.71	9.47	20.62
		DT-LGBM	22.26	0.71	9.56	20.97
	v2	ANN-LSTM CL-1Hot (median)	22.64	0.72	9.82	22.22
		ANN-LSTM CL-Ord. (median)	22.13	0.70	9.52	21.09
		ANN-LSTM REG	21.96	0.70	9.46	20.76
		ANN-MLP CL-1Hot (median)	**21.44**	0.68	9.15	20.01
		ANN-MLP CL-Ord. (median)	**21.34**	**0.68**	**9.09**	**19.97**
		ANN-MLP REG	21.40	**0.68**	**9.12**	**20.00**
		DT-LGBM	21.49	0.68	9.16	20.07

Table 4.18: Forecast accuracy at the store-article level.

4.5.2.1 Overview on the Results

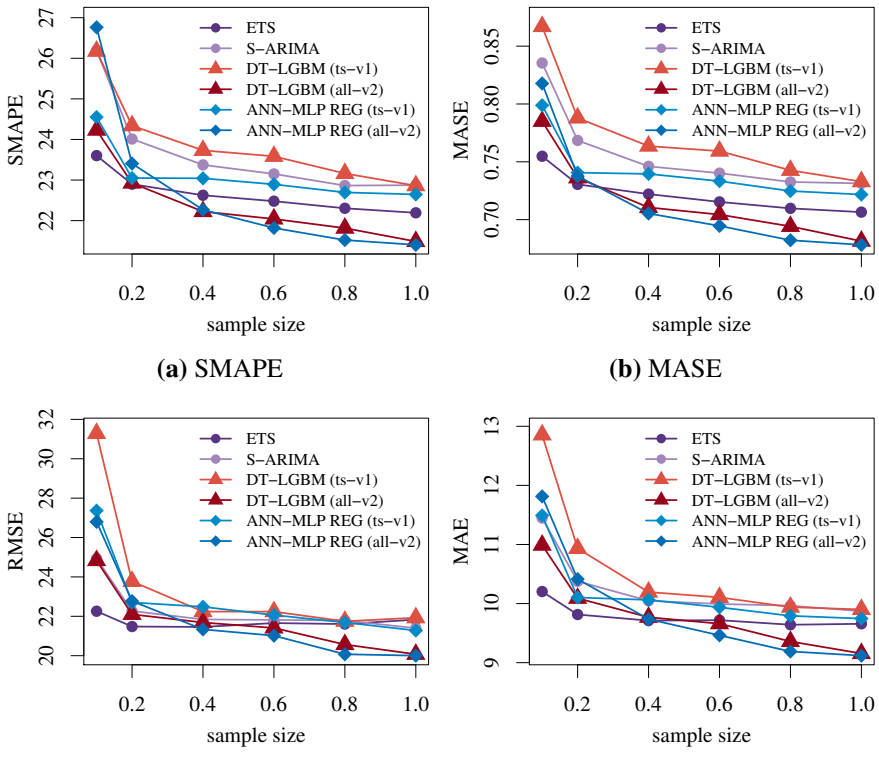
The main results of the evaluation are presented in Table 4.18. A comparison of the baseline methods reveals that more advanced methods lead to a significantly higher forecast accuracy, i.e., *S-Median* is better than *S-Naïve* while *ETS* and *S-ARIMA* outperform *S-Median*. Overall, *ETS* is the best reference method because it is only beaten by *S-ARIMA* according to RMSE. *ETS* is also superior to the ML methods when they are trained per time series. Overall, the method *ANN-MLP CL-Ord. (median) (all-v2)* is superior to all other approaches. Generally, the characteristics of the data enable the application of more sophisticated prediction models. Another observation is that the presented classification-based approaches perform reasonably well compared to their regression-based counterparts. We also notice that the performance of all LSTM-based models is relatively poor compared to the other ML methods. LSTM is the most sophisticated ML method and may require a different configuration (e.g. features, hyper-parameters) or even more data as it overfits the training data.



(a) SMAPE **(b)** MASE

(c) RMSE **(d)** MAE

Figure 4.4: Effect of data usage on the forecast accuracy. The horizontal line indicates the performance of ETS. The bars depict the performance for the combinations of model scope (ts, all) and used features (v1, v2).



(a) SMAPE **(b)** MASE

(c) RMSE **(d)** MAE

Figure 4.5: Effect of the sample size at the store-article level for different error measures.

4.5.2.2 Analysis of Data Usage

While a ML method is able to significantly outperform the reference method, we are also interested how the use of data affects the performance. We consider three levels of data usage: model scope, features, and sample size (demand history). The effect of the first two levers is shown in Figure 4.4. First of all, we notice that it is important to train models across stores and products (model scope: all). The substantially larger training set (factor: 55) can be exploited by ML methods and leads to smaller prediction errors. This is a convenient result as this also means that a smaller number of models needs to be optimized. *ETS* is very competitive if ML methods are trained for each time series independently. When features are added (feature set: v2) the performance of *ANN-LSTM REG* and *ANN-MLP REG* even decreases, which is caused by overfitting the training data. However, if data from every time series and additional explanatory data (all-v2) is used, the performance of all ML methods improves considerably for all error measures and is superior to the reference methods.

The sample size (see Figure 4.5) also has a significant impact on the performance. If very little data is available (sample size ≤ 0.2), the ML methods perform very poor, while especially *ETS* already yields a comparably good performance. However, the performance of *ETS* and *S-ARIMA* stagnates in particular with respect to RMSE and MAE rather quickly. This is to some degree also true for *ANN-MLP REG (ts-v1)* and *DT-LGBM (ts-v1)*. In contrast, the performance of *ANN-MLP REG (all-v2)* and *DT-LGBM (all-v2)* constantly improves as more data becomes available. We also notice that a rather short demand history of only 150 days (sample size = 0.4) is required to outperform the reference methods. In summary, training across multiple time series with features (all-v2) seems to be the only successful way to exploit all available data and also benefit from a growing demand history.

4.5.2.3 Analysis of Model Re-training

The previous results indicate that only a small number of models need to be optimized as data pooling significantly improves the forecast accuracy. Another important factor for the applicability of ML in a productive setting is the frequency of training the models. Due to computational constraints, the ANNs were already only trained every 50 days during the 150 days of the test phase. We analyze how no training during the test phase affects the performance (see Table 4.19). If the training data is pooled (all-v1, all-2), the error metrics change less than 1% for all methods. Even the accuracy of the *DT-LGBM* approaches, which were initially more frequently trained, hardly diminishes. However, if the training data is not pooled (ts-v1, ts-v2), we notice that the forecasts are significantly less accurate when models are only trained once. The effect is most prevalent for *DT-LGBM* whose predictions yield $5 - 15\%$ higher forecast errors. The observations underline an additional advantage of data pooling: As more demand patterns (i.e. more data) are provided to the models during the training phase, the models are better equipped for slight changes in the demand patterns and are more robust to noise in the training data. Hence, ML models can be used for a long time span without re-training.

Method	SMAPE	Δ [%]	MASE	Δ [%]	RMSE	Δ [%]	MAE	Δ [%]
ETS	22.62	1.90	0.72	1.40	23.27	6.60	10.05	4.00
S-ARIMA	22.85	-0.10	0.73	0.00	21.22	-0.80	9.90	0.30
LSTM REG (ts-v1)	25.00	3.20	0.82	3.80	25.36	7.20	11.40	6.00
MLP REG (ts-v1)	22.85	0.90	0.73	1.40	21.55	1.30	9.87	1.20
DT-LGBM (ts-v1)	24.12	5.50	0.78	6.80	24.42	11.40	10.91	10.20
LSTM REG (ts-v2)	25.53	1.60	0.85	2.40	29.44	3.70	12.41	3.10
MLP REG (ts-v2)	23.74	2.50	0.77	2.70	24.68	5.30	10.85	4.00
DT-LGBM (ts-v2)	23.90	6.00	0.78	8.30	24.40	15.30	10.77	11.80
LSTM CL-1Hot (median) (all-v1)	22.63	0.00	0.72	0.00	21.88	0.00	9.80	0.00
LSTM CL-Ord. (median) (all-v1)	22.63	0.30	0.72	0.00	22.15	0.00	9.87	-0.10
LSTM REG (all-v1)	22.65	0.10	0.73	0.00	22.46	0.90	9.92	0.40
MLP CL-1Hot (median) (all-v1)	22.23	-0.10	0.71	0.00	20.66	-0.20	9.45	-0.10
MLP CL-Ord. (median) (all-v1)	22.16	0.10	0.70	0.00	20.52	-0.90	9.40	-0.30
MLP REG (all-v1)	22.31	0.20	0.71	0.00	20.60	-0.10	9.48	0.10
DT-LGBM (all-v1)	22.27	0.00	0.71	0.00	20.82	-0.70	9.54	-0.20
LSTM CL-1Hot (median) (all-v2)	22.57	-0.30	0.72	0.00	22.20	-0.10	9.83	0.10
LSTM CL-Ord. (median) (all-v2)	22.09	-0.20	0.70	0.00	20.86	-1.10	9.49	-0.30
LSTM REG (all-v2)	21.91	-0.20	0.70	0.00	20.73	-0.10	9.41	-0.50
MLP CL-1Hot (median) (all-v2)	21.42	-0.10	0.68	0.00	**19.94**	-0.30	9.13	-0.20
MLP CL-Ord. (median) (all-v2)	**21.37**	0.10	**0.68**	0.00	20.02	0.30	**9.10**	0.10
MLP REG (all-v2)	21.48	0.40	0.68	0.00	**20.11**	0.60	9.18	0.70
DT-LGBM (all-v2)	21.44	-0.20	**0.68**	0.00	**19.98**	-0.40	**9.12**	-0.40

Table 4.19: Effect of re-fitting during the test phase. The accuracy measures indicate the performance when the models are not re-fitted during the test phase and the relative performance change (Δ [%]) compared to re-fitting during the test phase (see Table 4.18).

Method	SMAPE	MASE	MAE	RMSE	SL
ANN-MLP CL-1Hot (max)	28.05	0.84	11.63	25.82	0.44
ANN-LSTM CL-1Hot (max)	27.37	0.84	11.89	27.28	0.43
ANN-MLP CL-1Hot (median)	21.44	0.68	9.15	20.01	0.50
ANN-LSTM CL-1Hot (median)	22.64	0.72	9.82	22.22	0.51
ANN-MLP CL-Ord. (median)	**21.34**	**0.68**	**9.09**	**19.97**	0.51
ANN-LSTM CL-Ord. (median)	22.13	0.70	9.52	21.09	0.51

Table 4.20: Forecast accuracy using different formulations for the classification problem.

4.5.2.4 Analysis of Classification

The results presented in Table 4.18 show that the transformation of the regression problem to a classification problem can lead to lower forecast errors. Thus, we analyze the results in more detail (see Table 4.20). In typical classification problems, the class with the highest probability (max) is picked. Unfortunately, this approach leads to unsatisfactory results on this dataset as the demand is systematically underestimated, i.e., the service level is far below 50%. However, if 1-hot encoding is employed, we can exploit the ordering of the classes and compute a cumulative distribution function that allows selecting specific quantiles (e.g. median). This approach works reasonably well as the forecasts are less biased and achieve a higher service level that is better suited for the considered forecast accuracy measures. The differences between the two encoding strategies are rather small, but the ordinal encoding leads to slightly better results for both types of ANNs.

Threshold	Method	SL	SL Δ	Fill Rate	Loss Rate
0.5	ANN-LSTM CL-1Hot	0.52	-0.01	0.92	0.21
	ANN-LSTM CL-Ord.	0.51	-0.01	0.92	0.20
	ANN-MLP CL-1Hot	0.50	-0.00	0.92	0.20
	ANN-MLP CL-Ord.	0.51	-0.01	0.92	0.19
0.6	ANN-LSTM CL-1Hot	0.62	-0.02	0.94	0.26
	ANN-LSTM CL-Ord.	0.61	-0.01	0.94	0.26
	ANN-MLP CL-1Hot	0.62	-0.02	0.94	0.25
	ANN-MLP CL-Ord.	0.62	-0.02	0.94	0.25
0.7	ANN-LSTM CL-1Hot	0.72	-0.02	0.96	0.33
	ANN-LSTM CL-Ord.	0.71	-0.01	0.96	0.32
	ANN-MLP CL-1Hot	0.73	-0.03	0.96	0.31
	ANN-MLP CL-Ord.	0.73	-0.03	0.96	0.31
0.8	ANN-LSTM CL-1Hot	0.81	-0.01	0.98	0.42
	ANN-LSTM CL-Ord.	0.81	-0.01	0.98	0.40
	ANN-MLP CL-1Hot	0.82	-0.02	0.98	0.40
	ANN-MLP CL-Ord.	0.82	-0.02	0.98	0.40
0.9	ANN-LSTM CL-1Hot	0.90	0.00	0.99	0.56
	ANN-LSTM CL-Ord.	0.90	0.00	0.99	0.53
	ANN-MLP CL-1Hot	0.91	-0.01	0.99	0.54
	ANN-MLP CL-Ord.	0.91	-0.01	0.99	0.53
0.95	ANN-LSTM CL-1Hot	0.95	0.00	0.99	0.70
	ANN-LSTM CL-Ord.	0.95	0.00	0.99	0.66
	ANN-MLP CL-1Hot	0.95	-0.00	0.99	0.66
	ANN-MLP CL-Ord.	0.95	0.00	0.99	0.66

Table 4.21: Effect of the threshold on the achieved service level (SL). The column SL Δ refers to the absolute deviation from the service level to the threshold. A higher service level also leads to a higher fill rate as well as a higher loss rate.

As the classification approach leads to good results for median predictions, we also check if other quantiles can be accurately predicted. The results presented in Table 4.21 suggest that this is indeed the case as the absolute deviation from the expected service level is frequently below 2%. It is a very interesting but expectable observation that the quantiles are actually learned by the models without providing a specific loss function. Hence, another benefit from the transformation to a classification task is that essentially the probability distribution over all discretized target values is accurately computed. In this context, we can point out that there is a trade-off between a higher fill rate and a lower loss rate (see Figure 4.6). The ranking of the methods remains stable with respect to this trade-off for all service levels, i.e., the method having the lowest forecast errors (*ANN-MLP CL-Ord.*) is the best choice. We study methods to deal with this trade-off by maximizing profits in Part III.

4.6 Application: Hierarchical Forecasts

In this section, we study forecasts at different levels of the organizational hierarchy and product hierarchy (see Section 3.3.1 and Figure 3.1). We evaluate the forecast accuracy at different levels and analyze if the hierarchy can be exploited in order to reduce computational costs. It can be expected that a higher accuracy can be achieved at higher levels in the hierarchy as uncertainty is pooled which makes the time series less volatile and the demand patterns more

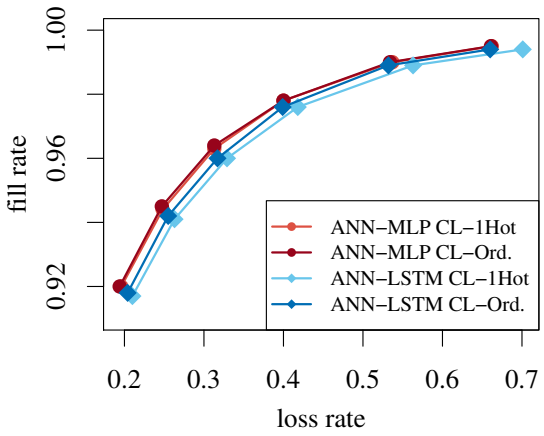


Figure 4.6: Trade-off: Fill rate vs. loss rate.

prevalent. Hence, forecasts at aggregated levels can serve as sanity checks for predictions at lower levels. For instance, van Donselaar et al. (2006) suggest that an inventory control system for perishable items should monitor the total orders for a group of substitutable items as this is an indicator for expected waste. The levels of interest, as outlined in Section 3.3.1, include region (RX), region-category (RC), region-article (RA), store-all (SX), store-category (SC), and store-article (SA). For the used dataset, the region (RX) is equivalent to the sum of the demand over all stores because the considered stores belong to the same region.

4.6.1 Experimental Setup

The evaluation in this section is an extension of Section 4.5 and the experimental setup is vastly identical (see Section 4.5.1). The evaluation is also based on dataset v2 (see Section 3.2) but we do additionally consider different levels of the hierarchy. The dataset contains substitutable products from category C1 (6 buns) and from category C2 (5 breads). With respect to the ML methods, we consider the approach *ANN-MLP REG (all-v2)*, subsequently denoted *ANN-MLP*, as it performed reasonably well at the store-article level. We consider only one ML method as our focus is the effect of demand forecasting at different levels in the hierarchy rather than an extensive comparison of different approaches. However, we consider the benchmark methods as they provide a reference and allow us to assess the quality of the predictions of *ANN-MLP*. For the training of *ANN-MLP*, we rely not only on data from all stores and all articles but also on data from all levels in the hierarchy in order to build a global model.

The bottom-up forecasts are the sum of the forecasts at the lower level. For the top-down forecasts, we compute the allocation proportions that are necessary to allocate the predictions from higher levels to lower levels. The allocation schemes are based on historic proportions, as suggested by Gross and Sohl (1990). For each weekday w, the demand proportion PR of

	SA	SC	SX	RA	RC	RX
SA	1	↓ 7 / 9	↓ 16	↓ 5	↓ 25 / 30	↓ 55
RA	↑ 5	-	-	1	↓ 5 / 6	↓ 11

Table 4.22: (Dis-)aggregation ratio between the layers (top-down: ↓, bottom-up: ↑).

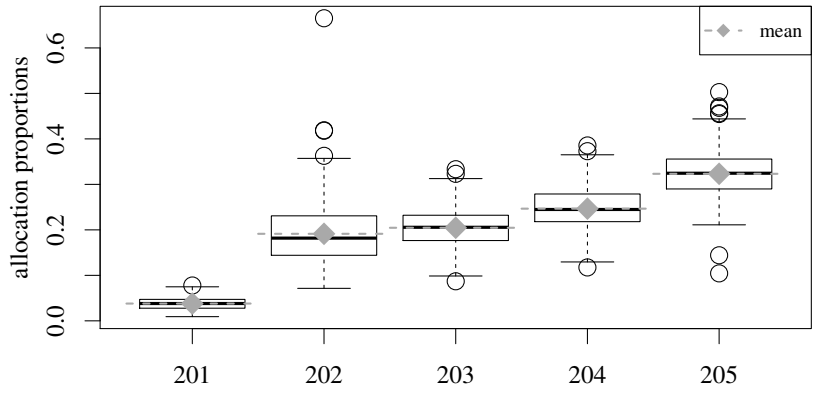


Figure 4.7: Demand proportions at the store-article level in relation to the demand at the store-category level at a single store.

an article a at layer l_1 depends on the total demand at the upper level l_2 based on past sales observations:

$$PR_{l_1,l_2,a,w} = \frac{1}{T} \sum_t \frac{y_{l_1,a,w,t}}{y_{l_2,w,t}} \tag{4.1}$$

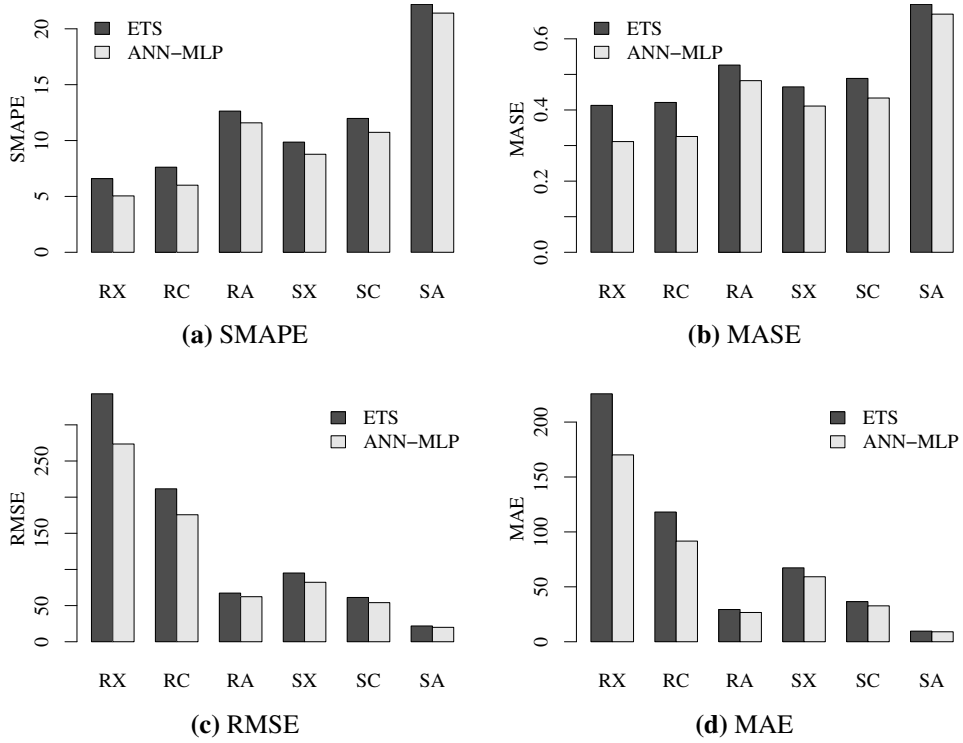
We evaluate two allocation strategies: For the first strategy (allocation: *fixed*), we compute the average allocation proportions only once over the complete training set for each combination of two levels. For the second strategy (allocation: *fc*), we compute the allocation proportions using a rolling seasonal median of the last four proportions. The advantage of the former strategy is that it is based on a larger number of observations, which seems to be reasonable as the proportions are rather stable, while the latter strategy considers more recent information concerning the demand distributions. As an example, the empirical proportions between the SC level and the SA level are depicted in Figure 4.7 for category C2 and a specific store.

The (dis-)aggregation ratio, e.g., the number of time series to which a single forecast needs to be distributed (top-down) and the number of forecasts that are aggregated (bottom-up), is given in Table 4.22. For instance, a prediction at the highest level (RX) needs to be allocated to 55 time series at the SA level.

4.6.2 Results & Discussion

The analysis of the results of the evaluation falls into two parts. First, we investigate the performance of the direct forecasts at each level of the hierarchy (see Section 4.6.2.1). Subsequently, we assess if hierarchical forecasts allow reducing the computational costs without sacrificing the performance (see Section 4.6.2.2). Thereby, we focus on the prediction accu-

racy for the daily demand at the store-article (SA) level and the region-article (RA) level as these are the most relevant levels in the present application scenario. Decisions concerning production and delivery quantities are made on the article level.



(a) SMAPE **(b)** MASE

(c) RMSE **(d)** MAE

Figure 4.8: Accuracy at different levels of the hierarchy.

4.6.2.1 Direct Forecasts

The results of forecasts at different levels of the hierarchy are presented in Table 4.23 and Figure 4.8. A general observation is that the relative errors (SMAPE) and scaled errors (MASE) decrease while the scale-dependent errors (MAE, RMSE) increase for higher levels in the hierarchy. The increase of scale-dependent errors has to be expected as aggregating the demand from multiple time series increases the volumes substantially. For instance, the level RX is based on the sum of 55 time series from the SA level. The decrease of relative errors suggests that uncertainty from multiple time series is pooled. From the SA level to the SC level, the relative errors are roughly halved, which is favorable as the goods within each category are substitutable. Hence, forecasts at the SC level can be used to validate the forecasts at the SA level. Between those two levels, the fill rate increases from 92% to 95% and the loss rate drops from 20% to 6% using method *ANN-MLP*. Hence, by relying on substitution, i.e., substitution rates for baked goods are as high as 84% (van Woensel et al., 2007), it is possible to limit the amount of discarded goods and to avoid costly turnover losses that result from maintaining a high service level for each article. The key figures are similar at the RA level (fill rate: 96%, loss rate: 8%), which indicates that accurate planning of the production

Level	Method	SMAPE	MASE	RMSE	MAE	SL	FR	LR
RX	S-Naïve	13.47	0.66	717.21	363.47	0.49	0.94	0.05
	S-Median	**6.95**	0.42	381.18	231.45	0.56	0.97	0.04
	ETS	6.59	0.41	342.98	225.76	0.51	0.97	0.04
	S-ARIMA	**6.40**	0.40	324.29	219.64	0.58	0.98	0.05
	ANN-MLP	**5.05**	**0.31**	**273.53**	**170.13**	0.62	0.98	0.04
RC	S-Naïve	**15.33**	0.67	448.44	187.16	0.52	0.93	0.05
	S-Median	**8.18**	0.43	233.93	120.40	0.55	0.96	0.04
	ETS	**7.62**	0.42	211.55	118.11	0.49	0.96	0.04
	S-ARIMA	7.41	0.41	200.86	115.44	0.58	0.97	0.05
	ANN-MLP	**6.01**	**0.33**	**175.66**	**91.65**	0.54	0.98	0.04
RA	S-Naïve	21.33	0.76	138.84	44.16	0.49	0.90	0.10
	S-Median	13.92	0.58	75.71	31.49	0.50	0.94	0.08
	ETS	**12.64**	0.53	67.31	29.34	0.52	0.95	0.09
	S-ARIMA	12.66	0.53	64.60	29.08	0.57	0.95	0.09
	ANN-MLP	**11.59**	**0.48**	**62.30**	**26.65**	0.53	0.96	0.08
SX	S-Naïve	18.06	0.72	165.94	103.46	0.50	0.92	0.08
	S-Median	**11.00**	0.51	103.05	73.12	0.50	0.95	0.07
	ETS	**9.86**	0.46	95.08	67.30	0.52	0.96	0.06
	S-ARIMA	9.96	0.47	92.69	67.58	0.55	0.96	0.07
	ANN-MLP	**8.77**	**0.41**	**82.26**	**59.17**	0.53	0.97	0.06
SC	S-Naïve	**20.74**	0.72	106.21	55.17	0.49	0.91	0.09
	S-Median	13.05	0.52	66.37	39.21	0.50	0.94	0.07
	ETS	11.98	0.49	61.26	36.50	0.52	0.95	0.07
	S-ARIMA	12.00	0.49	60.04	36.84	0.56	0.96	0.08
	ANN-MLP	**10.74**	**0.43**	**54.09**	**32.69**	0.50	0.95	0.06
SA	S-Naïve	28.71	0.91	27.80	12.56	0.51	0.88	0.23
	S-Median	23.89	0.75	22.48	10.21	0.51	0.91	0.21
	ETS	22.19	0.70	21.83	9.66	0.53	0.92	0.20
	S-ARIMA	22.87	0.72	21.40	9.87	0.55	0.92	0.22
	ANN-MLP	**21.41**	**0.67**	**19.94**	**9.08**	0.53	0.92	0.20

Table 4.23: Forecast accuracy and operational key figures (SL = service level, FR = fill rate, LR = loss) at different levels of the hierarchy.

quantities is possible but the allocation to the stores remains a challenge. However, it is likely that the key figures with respect to fill rate and loss rate are slightly better at the SA level than reported as substitution is not considered and the achieved service level is only at 53%. The measured service level roughly matches the expectation for unbiased forecasts, but it also means that in almost 50% of the cases the full demand could not be fulfilled, i.e., 8% of the demand could not be served. We study methods to increase the service level by maximizing the profits in Part III. The scale-independent error metrics are lowest at the RX level and still relatively low at the SX level, which implies that accurate short-term revenue predictions, which are required for staffing decisions, should be feasible.

The model *ANN-MLP* outperforms all reference methods on all levels of the hierarchy with respect to all error metrics (see Figure 4.8). It is particularly noteworthy that relative performance gains due to the employment of *ANN-MLP* over simpler reference methods tend to increase at the aggregated levels. This observation is also true for the comparison between the time series models (*ETS*, *S-ARIMA*) and *S-Naïve*. Hence, despite the fact that the data is less volatile at aggregated levels, it is beneficial to rely on more complex models that are able to separate noise from the actual demand patterns. *ANN-MLP* offers not only the lowest prediction errors but also the highest fill rate and lowest loss rate across all levels which

Target	Method	Mode	Allocation	Base	SMAPE	MASE	RMSE	MAE
RA	ETS	top-down	fixed	RX	14.83	0.63	76.78	34.75
				RC	14.89	0.63	77.41	34.86
			fc	RX	13.18	0.55	71.95	31.08
				RC	13.18	0.56	72.68	31.27
		direct	-	RA	**12.64**	0.53	67.31	29.34
		bottom-up	-	SA	11.87	0.50	66.52	28.23
	ANN-MLP	top-down	fixed	RX	**14.13**	0.59	67.60	31.91
				RC	13.82	0.58	69.50	32.13
			fc	RX	12.26	0.51	61.93	27.47
				RC	12.11	0.51	65.39	27.95
		direct	-	RA	11.59	0.48	62.30	26.65
		bottom-up	-	SA	**11.18**	**0.46**	**58.67**	**25.56**
SA	ETS	top-down	fixed	RX	24.26	0.77	24.04	10.81
				RC	24.23	0.77	24.16	10.81
				RA	23.10	0.74	22.63	10.17
				SX	24.31	0.77	22.89	10.51
				SC	24.12	0.77	22.98	10.49
			fc	RX	23.95	0.76	22.39	10.32
				RC	24.00	0.76	22.43	10.34
				RA	23.75	0.75	21.90	10.18
				SX	23.89	0.75	22.10	10.16
				SC	23.88	0.75	22.19	10.17
		direct	-	SA	22.19	0.70	21.83	9.66
	ANN-MLP	top-down	fixed	RX	**23.80**	0.76	22.74	10.44
				RC	23.56	0.75	22.88	10.42
				RA	22.48	0.71	21.77	9.83
				SX	23.65	0.75	20.97	9.90
				SC	23.31	0.74	20.91	9.88
			fc	RX	23.43	0.74	21.01	9.88
				RC	23.33	0.73	21.41	9.92
				RA	23.08	0.73	21.13	9.81
				SX	23.20	0.73	20.22	9.53
				SC	23.09	0.72	20.42	9.56
		direct	fixed	SA	**21.41**	**0.67**	**19.94**	**9.08**

Table 4.24: The forecast accuracy at different levels using different approaches to exploit the hierarchy.

supports the assumptions that ML methods are suitable for this application scenario. At the SA level, we observe slightly lower prediction errors than reported in the previous section (see *MLP REG (all-v2)* in Table 4.18). Thus, augmenting the training data with additional samples that are not directly relevant, e.g., demand patterns at aggregated levels, can be beneficial.

4.6.2.2 Hierarchical Forecasts

We investigate if the hierarchy can be exploited in order to obtain more accurate results or to reduce computational costs. We focus on the demand predictions for articles on store level (SA) and region level (RA) as those are relevant levels with respect to production planning and deliveries.

Overall, the top-down forecasts do not improve the accuracy of the direct forecasts (see Table 4.24). However, some hierarchical forecasts achieve a comparable accuracy with the direct forecasts. For predictions at the SA level, the article categories (SC, RC) are more suitable for top-down forecasting as they lead to better results than the aggregation of all articles (SX, RX). At the SA level, it is also apparent that deriving the demand estimation based on forecasts at region level (RA, RC, RX) is worse than relying on the aggregated forecasts from the same organizational level (SC, SX). A reason for this might be that the number of the required allocation proportions is rather high (see Table 4.22). Moreover, the sales dynamics of the different stores may vary slightly and are not sufficiently covered in the allocation proportions. Hence, a trade-off between forecast accuracy and computational costs is to forecast the demand at the SC level and thereby exploit the substitutability of the goods. Based on these forecasts, the predictions at the SA level can be calculated with top-down forecasts. The derived predictions at the SA level are then used to compute the RA level predictions bottom-up as the RA level seems to benefit from predictions at lower levels.

The comparison of both allocation strategies reveals that it is in the vast majority of the cases better to rely on a rolling forecast instead of the average of many historic and partially outdated observations. A possible explanation is that the general demand distribution slightly changes or that the time of the year (e.g. summer vs. winter) has a slight impact. However, by obtaining the allocation proportion using a seasonal median, the top-down predictions of *ANN-MLP* are more accurate than the best forecast approach (e.g. direct) of the reference method *ETS*. Hence, even if it is not feasible to use a ML method at the SA level, they still offer value for top-down predictions that are less computationally demanding.

The qualitative results are compatible with those reported in the literature (see Section 4.2). Top-down predictions can be applied if the forecasts at the top-level and the allocation proportions are sufficiently accurate. In the present use case, the allocation proportions between the SC level and the SA level are rather stable while the prediction accuracy at the top-level is higher than at the bottom-level. We also observe that it is important to group time series with similar demand patterns. For instance, the SC level seems to be slightly better suited to predict the demand at the SA level than top-down forecasts from the RA level even though the numbers of allocation proportions are comparable (see Table 4.22). Thus, it is also important to consider differences among the stores. In general, the forecasts at aggregated levels are quite accurate as the patterns of the lower level time series are similar. Hence, the information loss can be neglected while the noise is reduced. Nevertheless, direct forecasts are also possible in this use case because the items are fast-moving goods. This means that enough information is available at any level as the items are sold in high volumes on a daily basis. Thus, bottom-up forecasts are at least comparable to direct forecasts at an aggregated level. However, the advantages of top-down forecasts based on meaningful groups comprise reduced computational costs and increased scalability.

4.7 Conclusion

In this chapter, we highlight challenges and opportunities for daily large-scale demand forecasting in the retail industry. To this end, we present solution approaches for large-scale demand forecasting based on ML that are able to leverage large datasets. The considered datasets contain time series that are enriched with explanatory data. We also conduct a comprehensive empirical evaluation that covers different aspects of the methods and focuses on specific characteristics of the use case.

The most important result is that ML methods are indeed a viable alternative to established approaches for large-scale retail forecasting. Our empirical evaluation offers evidence that a ML method is the best approach at every level of the organizational hierarchy as well as the article hierarchy. However, an analysis of different levels of data usage reveals that the performance of ML methods heavily depends on the data that is used during the training phase. Statistical time series models are the better option if not enough data is available. This is the case if the demand history covers less than 150 days. Moreover, if a ML model is fitted for each time series separately, it performs usually worse than the reference methods. If the models are fitted per time series, adding explanatory feature data can even lead to worse results as overfitting is more likely (e.g. curse of dimensionality). Hence, we conclude that a few hundred training samples are hardly enough to build a good prediction model based on ML. Those results partially support the observations of comparative studies that emphasize the competitiveness of time series models (e.g. Makridakis et al. (2018a,b)).

However, the flexibility of ML methods allows to pool data from multiple time series in order to increase the number of training samples substantially. Data pooling is reasonable in the considered application scenario because it can be expected that stores and articles share similar demand patterns but it is not certain that a method benefits from data pooling as each time series has also unique characteristics. ML methods perform at least as good as the reference methods if pooled training data without features is used. The inclusion of explanatory feature data provides an additional performance boost and makes ML methods superior. More data allows the models to separate noise from actual demand patterns which is in particular useful for special days where only a small number of observations is available per time series and it is difficult to define suitable adjustment rules. A rather short demand history of 150 days is sufficient to outperform the reference methods if data is pooled. While the performance of the reference methods stagnates rather quickly, the performance of ML methods continuously improves as more training data becomes available. Therefore, we come to the conclusion that only ML methods are able to gain more information from a growing amount of training data.

The fact that data pooling works very well is also favorable for the application of ML models because this means that only a smaller number of ML models needs to be optimized and maintained which makes the deployment of ML models in real-world applications much more feasible. In this context, our results also suggest that frequent re-fitting of the ML models that are trained with data across stores and articles might not be necessary as the

performance gains are limited, i.e., less than 3%. Hence, ML models need only to be trained every couple of months or even less frequent, which reduces the computational requirements dramatically.

In this context, we need to point out that leveraging the hierarchies of the application domain is also a possibility to reduce computational costs. For certain combinations of top-level (e.g. store-category level) and bottom level (e.g. store-article level), it is reasonable to rely on top-down forecasts if other approaches are not feasible. In some cases, top-down predictions of ML models are even more accurate than direct predictions of the reference methods at the lower level. Forecasts at different levels of the hierarchy can also serve as a sanity check of the predictions at lower levels as uncertainty is pooled which makes aggregated predictions partially more reliable. This is especially useful as articles in certain categories of baked goods are substitutable and subject to high substitution rates. For ML methods, it can also be beneficial to augment the training set with samples from aggregated time series, which is another advantage for considering the hierarchies.

With respect to the evaluated methods, we notice that artificial neural networks (ANNs) outperform gradient boosted decision trees. However, the results are rather mixed with respect to the different types of ANNs, i.e., recurrent neural networks (e.g. LSTMs) and feed-forward neural networks (e.g. MLPs). The LSTMs perform best at the store-category level on the biggest dataset (dataset v1) while the results are comparatively poor on the store-article level using a much smaller dataset (dataset v2). The structure of a LSTM is rather complex, which makes overfitting more likely. Hence, LSTMs need either more training data to prevent this problem or the input sequences and features need to be designed more carefully.

Moreover, for both types of neural networks, it is beneficial to model the learning problem as a classification task instead of a regression problem. Transforming the forecasting problem to a classification task offers an additional level to address the uncertainty of a prediction. The prediction of a classification model, i.e., a probability distribution over the target classes, can be interpreted as a density forecast. For the typical encoding of a classification problem (i.e. 1-hot encoding), it is in particular recommendable to select the median of the distribution instead of the class having the highest probability, i.e., the mode of the distribution. However, an ordinal encoding of the target classes works slightly better and also allows to pick different quantiles of the predicted distribution.

The vast majority of the evaluation is concerned with single-step predictions because they are most relevant for the determination of ordering decisions. However, the experiments related to multi-step predictions indicate that the trained ML models are quite robust, i.e., the forecast errors are rather stable within a season (i.e. a week) and the ranking of the evaluated methods is constant over different horizons. Hence, it is reasonable to initially focus on single-step prediction before other forecasting steps are explored.

Part III

Decision Support

5 Daily Decision Support

The research presented in this chapter is based on joint work with Sebastian Müller, Moritz Fleischmann, and Heiner Stuckenschmidt. In particular, Section 5.3 is based on a paper titled *"A data-driven newsvendor: From data to decision"* (Huber et al., 2019) and Section 5.4 is based on a paper titled *"Data-driven Inventory Management under Customer Substitution"*. For both projects, I was particularly responsible for the aspects concerning forecasting and Machine Learning as well as the empirical evaluation.

A point forecast (e.g. expected value) does not reflect an optimal decision because forecasts are stochastic, i.e., a point forecast is only a functional of a distribution, and costs for overages and underages are not always symmetric. Newsvendor models (Silver et al., 2017) are inventory models for determining the optimal quantities of perishable goods, taking into account demand uncertainty and costs. Hence, we study and propose data-driven solution approaches for newsvendor problems in this chapter.

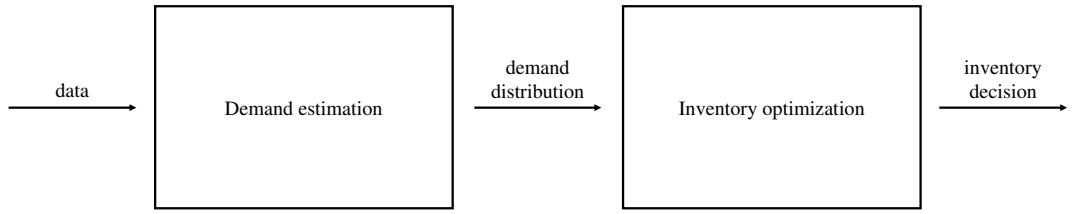
5.1 Introduction

Demand uncertainty is a major challenge in supply chain management practice and research. An important remedy for demand risk is the deployment of safety stock. In order to set appropriate stock levels, many inventory models assume a specific demand distribution (Silver et al., 2017). Despite the theoretical insights generated, the distribution assumption is problematic in real-world applications, as the actual demand distribution and its parameters are not known to the decision maker in reality and may even change over time (Scarf, 1958).

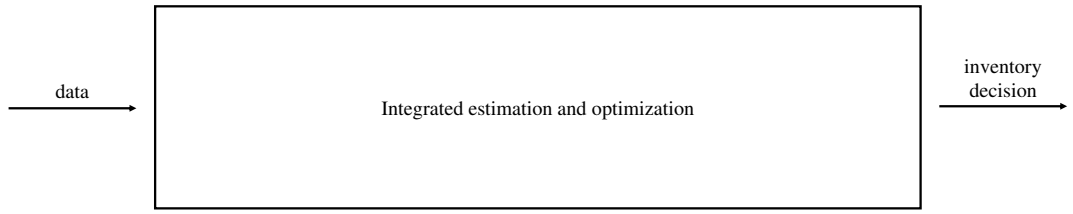
Recent advances in data storage and processing technologies have led companies to collect more and more data due to the promises of "big data". The growing availability of large datasets may help overcome the issue of unknown demand distributions and improve the performance of inventory models in real-world situations. Data that are indicative of future demand provide an opportunity to make better-informed decisions. These data include external information that is available through the Internet and data from internal IT systems. While this potential is widely recognized (see e.g. Bertsimas and Kallus (2018)) and the adaption of data-driven decision making increases, many companies are still struggling to turn data into better decisions (Brynjolfsson and McElheran, 2016).

Extant literature is rather fragmented in that regard and proposes multiple alternative directions. We intend to contribute to a more holistic understanding of the potential of data-

driven inventory management. Hence, our main research question is: *How to get from data to decision in newsvendor settings?* To this end, we distinguish three levels on which data can be used to revise the traditional decision process (see Figure 5.1). We discuss how these levels are interrelated, and we quantify their respective impact in a real-life application.



(a) Separate estimation and optimization



(b) Integrated estimation and optimization

Figure 5.1: The three levels of data-driven inventory management.

Traditionally, the process of inventory optimization contains the two steps *demand estimation* and *inventory optimization* (see Figure 5.1a). The demand estimation problem and the inventory optimization problem are usually addressed separately in the literature. The objective in the estimation problem is to minimize some estimation error (e.g. mean squared error). The problem of optimal inventory levels is usually not addressed by the forecasting and ML literature. Recently, the strict separation between estimation and optimization (and more generally between Machine Learning (ML) and Operations Research (OR)) has been questioned by some authors (Prak et al., 2017; Bertsimas and Kallus, 2018; Huber et al., 2019). Instead of separately estimating the demand distribution and optimizing the inventory decision, the authors integrate both into a single optimization problem. Hence, we distinguish three levels of data-driven approaches in inventory management:

- The first level on which data can be exploited is demand estimation. The available data may contain information about future demand that can be extracted by suitable forecasting methods. These methods use historical demand data and other feature data (e.g. weekdays, prices, weather, and product ratings) to estimate future demand. The output of these models is a demand estimate together with historical forecast errors. If additional information can be extracted, the reduced demand risk results in more accurate decisions. There is a large set of established and refined forecasting methods that use time series data in order to predict demand distributions. More recently, Machine Learning (ML) approaches have been successfully applied to numerous fore-

casting problems (Thomassey and Fiordaliso, 2006; Carbonneau et al., 2008; Crone et al., 2011; Barrow and Kourentzes, 2018; Huber et al., 2019).

- On the second level, the inventory decision is optimized based on the demand forecast and the historical forecast errors. To this end, it is necessary to incorporate the remaining uncertainty associated with the forecast. Traditionally, uncertainty is modeled through a demand distribution assumption (Silver et al., 2017). We call this approach *model-based* since it explicitly models a demand distribution. However, this assumption might be misspecified and lead to suboptimal inventory policies (Ban and Rudin, 2018). Instead of speculating about a parametric demand distribution, the assumption can be replaced by empirical data that are now available on large scale. This approach is called Sample Average Approximation (SAA) (Kleywegt et al., 2002; Shapiro, 2003) and we call it *data-driven* as it does not rely on a distribution assumption. In a multi-product case, this level also includes the consideration of substitution rates among the products.

- On the third level, demand estimation and optimization are integrated into a single model that directly predicts the optimal decision from historical demand data and feature data, as depicted in Figure 5.1b (Beutel and Minner, 2012; Sachs and Minner, 2014; Bertsimas and Kallus, 2018; Ban and Rudin, 2018; Huber et al., 2019). This approach is also *data-driven*, as it does not require the assumption of a demand distribution and works directly with data. We propose novel integrated estimation and optimization (IEO) approaches (see Figure 5.1b) that directly optimize the inventory decision based on data and do not rely on demand distribution assumptions.

From the existing literature, it is not yet clear whether and under which circumstances data-driven approaches are preferred to model-based approaches. Furthermore, the question of the conditions under which separate or integrated estimation and optimization is superior remains open.

To shed light on these questions, we focus on the newsvendor problem as the basic inventory problem with stochastic demand. We empirically analyze the effects of data-driven approaches on overall costs on the three levels. Moreover, we develop novel data-driven solution methods that combine modern ML approaches with optimization and empirically compare them to well-established methods.

In our approaches, we integrate Artificial Neural Networks (ANNs) and Decision Trees (DTs) into an optimization model. Most previous work on integrated estimation and optimization assumed the inventory decision to be linear in the explanatory features (Beutel and Minner, 2012; Sachs and Minner, 2014; Ban and Rudin, 2018). This assumption poses many restrictions on the underlying functional relationships. We extend this literature by integrating multiple alternative ML methods and optimization in order to avoid these strong assumptions and incorporate unknown seasonality, breaks, thresholds, and other non-linear relationships.

Recently, Oroojlooyjadid et al. (2016) and Zhang and Gao (2017) also used ANNs in this context.

We evaluate our solution approaches with real-world data from a large bakery chain in Germany. The company produces and sells a variety of baked goods. It operates a central production facility and over 150 retail stores. Every evening, each store must order products that are delivered the next morning. Reordering during the day is not possible. Most of the goods have a shelf life of only one day. Thus, leftover product at the end of the day is wasted, while stock-outs lead to lost sales and unsatisfied customers. Moreover, when customers cannot find their preferred product in stock, they might choose a similar product instead (Gruen et al., 2002; van Woensel et al., 2007).

From an optimization perspective, the problem can be represented by a newsvendor model, and the available point-of-sales data can be used to calculate forecasts. We consider two types of newsvendor problems: First, we study a single-product newsvendor problem where the order quantity of each product is optimized independently. Second, we study a multi-product newsvendor problem with substitution where the order decisions of substitutable products are interdependent. We apply our data-driven methods to the problems and compare their performance to the performance of well-established approaches. To summarize, our key contributions include the following:

- We investigate the process from data to decision in a single-product and a multi-product newsvendor setting.
- We identify and conceptualize three levels of data-driven approaches in inventory management.
- We propose novel data-driven integrated estimation and optimization (IEO) approaches for both newsvendor problems.
- We investigate the impact of the three levels on the overall performance in newsvendor problems.
- We compare our methods to well-established approaches on the three levels and show that data-driven methods outperform their model-based counterparts on our real-world dataset in most cases.
- We quantify the value of ML methods that exploit additional feature data.
- We quantify the value of considering substitution and uncertainty.

The remainder of this chapter is organized as follows. In the next section, we provide an overview of related literature (see Section 5.2). We propose and evaluate solution approaches for the single-product newsvendor in Section 5.3 and for the multi-product newsvendor with substitution in Section 5.4. We summarize our findings in Section 5.5.

5.2 Related Work

In this section, we review the OR literature related to demand uncertainty and demand substitution in inventory models. We refer to Section 4.2 for a review of literature concerning demand estimation using ML.

5.2.1 Demand Uncertainty

Most inventory management textbooks assume that the relevant demand distribution and its parameters are exogenously given and known (Silver et al., 2017). For a review of newsvendor-type problems, see Qin et al. (2011). In this section, we review the literature on inventory problems in which the demand distribution is unknown. More specifically, we focus on Robust Optimization, Sample Average Approximation (SAA), and Quantile Regression (QR).

One approach that needs only partial information on demand distributions is robust optimization (Ben-Tal et al., 2009). Scarf (1958) studies a single period problem in which only the mean and the standard deviation of the demand distribution are known. He then optimized for the maximum minimum (max-min) profit for all distributions with this property. Gallego and Moon (1993) further analyzed and extended it to a setting where reordering is possible. Bertsimas and Thiele (2006) and Perakis and Roels (2008) provide more insights into the structure of robust inventory problems. The main drawback of robust optimization is its limitation to settings with very risk-averse decision makers. For most real-world applications, robust optimization is overly conservative. For our analysis, we focus on methods that minimize expected costs instead of the max-min objective.

A data-driven method with a wider range of applications is Sample Average Approximation (SAA) (Kleywegt et al., 2002; Shapiro, 2003). Here, the demand distribution assumptions are replaced by empirical data. Levi et al. (2007) analyze the SAA solution of a newsvendor model and its multi-period extensions. The authors calculate bounds on the number of observations that are needed to achieve similar results compared to the case with full knowledge of the true demand distribution. These bounds are independent of the actual demand distribution. More recently, Levi et al. (2015) showed that the established bound is overly conservative and does not match the accuracy of SAA obtained in simulation studies. Therefore, they develop a tighter bound that is distribution specific. In this work, we provide empirical support for the good performance of SAA and compare the results of diverse methods.

Instead of using sequential estimation and optimization, integrating both steps into a single optimization model has been suggested (Bertsimas and Kallus, 2018). Beutel and Minner (2012) incorporate a linear regression function for demand into their newsvendor model. The authors test their approach on simulated data and actual retail data. The model was later extended to situations with censored demand observations (Sachs and Minner, 2014). Ban and Rudin (2018) propose an algorithm that is equivalent to the one in Beutel and Minner (2012),

in addition to a kernel optimization method. Furthermore, the authors show several properties of the algorithm and test it with empirical data in a newsvendor-type nurse staffing problem. Oroojlooyjadid et al. (2016) and Zhang and Gao (2017) integrate a neural network into a newsvendor model and compare it to several other approaches from the literature. However, they do not distinguish the effects of estimation, optimization, and integrated estimation and optimization. A drawback of extant research on integrated estimation and optimization is that non-linear relationships between inventory decision and feature data remain understudied. By using ML instead of a linear decision rule, our approaches can detect a priori unknown non-linear relationships between the optimal decision and the input features. Furthermore, we disentangle the effects of the three different levels of data usage highlighted in Figure 5.1.

It is well known that the optimal solution to the standard newsvendor model corresponds with a certain quantile of the demand distribution (Silver et al., 2017). Estimating a certain quantile of a distribution is known as Quantile Regression (QR) in the statistics and ML literature (Koenker, 2005). A very general approach to QR is presented by Takeuchi et al. (2006). The authors derive a quadratic programming problem and provide bounds and convergence statements of the estimator. Taylor (2000) use an ANN for QR in order to estimate conditional densities of financial returns. Similarly, Cannon (2011) describes an implementation of ANNs for QR and gives recommendations on solution approaches with gradient algorithms. More related to our application, Taylor (2007a) applies QR to forecast daily supermarket sales. The proposed method can be interpreted as an adaption of exponential smoothing to QR. In the empirical evaluation, the author tests three implementations of the method: one with no regressors, one with a linear trend term, and one with sinusoidal terms to account for seasonality. None of the papers on QR we found uses QR to evaluate the costs of an inventory decision. For our solution approach, we build on the existing literature on QR by integrating ML methods into the optimization model and evaluate the resulting costs of the newsvendor decision.

The challenge of incorporating demand uncertainty in inventory models without demand distribution assumptions is most recently also discussed by Trapero et al. (2019). They argue that the typical assumption of normal i.i.d. forecast errors should be questioned and suggest using a non-parametric kernel density approach for short lead times. Prak and Teunter (2019) propose a framework for incorporating demand uncertainty in inventory models that mitigates the parameter estimation uncertainty.

5.2.2 Demand Substitution

We review literature on the multi-product newsvendor problem under customer substitution and literature on the estimation of substitution probabilities.

There are many studies on the structural properties of the multi-product newsvendor problem under customer substitution and solution algorithms in the OR literature. Kök et al. (2015) provide a broad review on the topic. Two main modeling approaches for substitution can be distinguished. Models that rely on the first approach assume a specific customer choice

model (e.g. Multinomial Logit) (van Ryzin and Mahajan, 1999; Topaloglu, 2013; Farahat and Lee, 2018). In the second approach, substitution is represented by exogenous substitution rates. We follow the literature with exogenous substitution rates, which is prevalent in the inventory management literature. For the two-product case, Parlar and Goyal (1984) show that the objective function is concave under mild conditions and provide necessary optimality conditions. This work is later extended for more than two products by Netessine and Rudi (2003), who derive necessary optimality conditions for the more general case. The authors study the centralized case as well as competition. Schlapp and Fleischmann (2018) include capacity restrictions in addition to substitution.

Although these theoretical insights are important, in this study, we are more interested in methodologies that effectively solve real-world inventory problems. To this end, Zhang et al. (2018) develop two mixed integer linear programs that are able to solve problems with realistic sizes for many applications. For very large problems, they provide approximation algorithms. Closely related to our work is Kök and Fisher (2007), who describe a step-by-step approach from the estimation of substitution rates and demand from sales data to the final inventory decision. They develop a heuristic for the problem and apply their approach at a large supermarket chain and are able to gain a large increase in profit. Similarly, Hübner et al. (2016) develop a heuristic procedure to solve a multi-product newsvendor problem and can increase the solution quality and speed. While the extant work relies on separate estimation of demand distributions and optimization of inventory levels, we integrate both problems into a single optimization problem. While this has been done for the single-product problem (Beutel and Minner, 2012; Ban and Rudin, 2018; Huber et al., 2019) and for the two-product case (Sachs, 2015), we are not aware of any approach for more than two products.

All the above models need substitution rates as input. There are basically two ways to measure stock-out based substitution rates. Either directly ask customers for their response or infer their behavior from data. The first stream of literature collects responses to stock-outs with questionnaires. For ground coffee, orange juice, peanut butter tomato sauce, and tooth-paste, Emmelhainz et al. (1991) found that between 65% and 83% of customers substitute in response to a stock-out. Campo et al. (2000) find substitution rates of 44% and 51% for cereals and margarine, respectively. The most extensive study by Gruen et al. (2002) found that substitution rates vary significantly by category and are around 45% on average. Most related to our research is the work of van Woensel et al. (2007). The authors investigate the consumer responses to stock-outs of bakery bread and find that around 82% of customers are willing to substitute to another product if their first choice is not available.

Studies in the second stream of research estimate substitution from sales and inventory data instead of asking customers for their response. Some papers in this stream assume a specific customer choice model and estimate its parameters (Talluri and van Ryzin, 2004; Vulcano et al., 2010, 2012; Musalem et al., 2010), while we are more interested in methods that estimate exogenous substitution rates. To this end, Anupindi et al. (1998) propose an approach based on Maximum Likelihood Estimation (MLE) that works with inventory trans-

action data and test it with data from vending machines. Kök and Fisher (2007) generalize this approach to dynamic choice processes. The estimation method in Fisher and Vaidyanathan (2014) is also based on MLE. Wan et al. (2018) compare the accuracy of a customer choice model (Nested Logit) to exogenous substitution rates in a multi-store environment and find that the Nested Logit model outperforms the exogenous model. For the estimation of substitution rates in our empirical evaluation, we adapt the methodology of Anupindi et al. (1998) as all the necessary data is available.

To summarize, the literature on ML and OR is still relatively separate. More specific to our problem, there is only little research on how to get from data to decision. This is particularly true for the multi-product newsvendor problem with substitution. To address this gap, we propose novel data-driven approaches to the newsvendor problems that integrate demand estimation and inventory optimization into a single optimization problem. Our solution approaches are based on ML and leverage existing big data and computation power for inventory optimization. We empirically evaluate the impact of data-driven approaches on the three levels (1) estimation, (2) optimization, and (3) integrated estimation and optimization. To illustrate the viability of ML for inventory optimization, we also compare the data-driven methods to their model-based counterparts and other more traditional separate approaches.

5.3 Single-Product Newsvendor

In this section, we study how data can be exploited for decision optimization while the daily order quantity of each product is optimized independently. The remainder of this section is structured as follows: In Section 5.3.1, we describe the problem and introduce the methodology, including the data-driven ML approaches. Section 5.3.2 contains an introduction to the reference models, an empirical evaluation, and a discussion of the results.

5.3.1 Methodology

5.3.1.1 Problem Description

We consider a classical newsvendor problem with an unknown demand distribution: a company sells perishable products over a finite selling season with uncertain demand. The company must choose the number of products to order prior to the selling season. If the order is too high and not all products can be sold the company bears a cost of c_o for each unit of overage. If the order is too low and more units could have been sold, the company bears costs of c_u for each unit of underage. Thus, the objective is to minimize the total expected costs according to

$$\min_{q \geq 0} \mathbb{E}\left[c_u(D-q)^+ + c_o(q-D)^+\right], \tag{5.1}$$

where q is the order quantity and D is the random demand. The well-known optimal solution to this problem is to choose as the order quantity the quantile of the cumulative demand distribution function F that satisfies

$$q^* = \inf\left\{p : F(p) \geq \frac{c_u}{c_u + c_o}\right\}, \tag{5.2}$$

where $\frac{c_u}{c_u+c_o}$ is the optimal service level. The service level represents the probability of satisfying demand in a given period.

The problem that we address is that in most real-world cases, the actual demand distribution F is unknown. However, historical data $S_n = \{(d_1, \mathbf{x_1}), ..., (d_n, \mathbf{x_n})\}$ are available, where d_i is the demand and $\mathbf{x_i}$ is a vector of covariates or features (e.g. weekday, historical demand, and price) in period i. These data can be leveraged in different ways to reduce demand risk.

In the following sections, we present approaches that use the data on the three levels introduced in Section 5.1. First, we remark forecasting models based on ML that we use throughout our analysis. Next, we describe a data-driven optimization approach that leverages the empirical distribution of forecast errors. Finally, we present novel data-driven models that integrate ML and the optimization model.

5.3.1.2 Demand Estimation

If the underlying structure of the demand data is unknown, it is reasonable to consider very general forecasting models. ML methods have been applied to numerous forecasting tasks. Compared to traditional forecasting methods, ML is able to "learn" non-linear relationships between inputs and outputs. The most widely and successfully used methods are Artificial Neural Networks (ANNs) and Gradient Boosted Decision Trees (DTs). We refer to Section 2.2.2 for a brief description of the methods.

5.3.1.3 Optimization

Recall that the true demand distribution F is unknown to the decision maker. In the following sections, we present two different ways to deal with this problem: traditional model-based optimization and data-driven optimization based on SAA. Both approaches use the point forecast and the historical estimation errors as inputs to determine an inventory decision.

Model-based Optimization

The model-based approach assumes a certain forecast error distribution $\bar{F}$ (e.g. normal distribution) whose parameters θ (e.g. mean and standard deviation) are estimated based on historical forecast errors. The order quantity is then optimized by evaluating the function at the service level quantile and adding it to the forecast:

$$q(\mathbf{x}) = \hat{y}(\mathbf{x}) + \inf\left\{p : \bar{F}(p, \hat{\theta}) \geq \frac{c_u}{c_u + c_o}\right\}, \tag{5.3}$$

where $\hat{y}(\mathbf{x})$ is the mean forecast, given that the features $\mathbf{x}$, and $\hat{\theta}$ are the parameters of the error distribution estimated from the resulting forecast errors. In our evaluation, we adopt normally distributed errors for the model-based approaches.

Of course, this approach yields the optimal decision if the distribution assumption is true. However, in reality, the distribution is unknown and may even change over time. The observed forecast errors depend on the model chosen to produce the forecast. A misspecified model leads to errors that are not distributed as assumed. If the demand distribution is misspecified, highly distorted decisions may result. Ban and Rudin (2018) show this for the example of a normal distribution assumption where the actual demand is exponentially distributed.

Data-driven Optimization with Sample Average Approximation

A data-driven method to optimize the inventory decision is SAA. Here, the error distribution $\bar{F}$ is determined by the empirical forecast errors $\epsilon_1, ..., \epsilon_n$. A distribution assumption is not needed. Thus,

$$\bar{F}(p) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\epsilon_i \leq p). \tag{5.4}$$

To optimize the order quantity, the service level quantile of the empirical distribution is selected and added to the point forecast. Thus, the resulting order quantity given the features $\mathbf{x}$ is

$$q(\mathbf{x}) = \hat{y}(\mathbf{x}) + \inf \left\{ p : \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\epsilon_i \leq p) \geq \frac{c_u}{c_u + c_o} \right\}. \tag{5.5}$$

The performance of the optimization highly depends on the quality of the forecast, the number of available data points, and the target service level. Levi et al. (2007, 2015) provide worst-case bounds for a given number of observations. An important and intuitive result is that if the optimal service level is close to 0 or 1, i.e., extreme quantiles need to be estimated, the required sample size is much higher than for service levels close to 0.5, as extreme observations are rare.

5.3.1.4 Integrated Estimation and Optimization with Quantile Regression

Instead of sequentially forecasting demand and optimizing inventory levels, one can also directly optimize the order quantity by integrating the forecasting model into the optimization problem. The optimal order quantity q of the standard newsvendor model (5.1) is then a function of the feature data $\mathbf{x}$. Instead of first estimating the mean demand and the error distribution and then solving the newsvendor problem, we can now directly estimate the optimal order quantity from the feature data. Beutel and Minner (2012) and Ban and Rudin (2018) formulate this problem as a linear program. This implies that the optimal order quantity is

a linear function of the features. We extend these approaches by incorporating ML and thus also allowing for non-linear relationships:

$$\min_{\mathbf{\Phi}} \frac{1}{n} \sum_{i=1}^{n} \left[c_u (d_i - q_i(\mathbf{\Phi}, \mathbf{x_i}))^+ + c_o (q_i(\mathbf{\Phi}, \mathbf{x_i}) - d_i)^+ \right], \tag{5.6}$$

where $q_i(\mathbf{\Phi}, \mathbf{x_i})$ is the output of the ML method in period i with parameters $\mathbf{\Phi}$ (e.g. weight matrix of an ANN) and input variables $\mathbf{x_i}$.

By introducing dummy variables u_i and o_i for the underage and overage in period i, the problem can be reformulated as a non-linear program:

$$\min_{\mathbf{\Phi}} \frac{1}{n} \sum_{i=1}^{n} (c_u u_i + c_o o_i) \tag{5.7}$$

subject to:

$$u_i \geq d_i - q_i(\mathbf{\Phi}, \mathbf{x_i}) \qquad \forall i = \{1, ..., n\}, \tag{5.8}$$

$$o_i \geq q_i(\mathbf{\Phi}, \mathbf{x_i}) - d_i \qquad \forall i = \{1, ..., n\}, \tag{5.9}$$

$$u_i, o_i \geq 0 \qquad \forall i = \{1, ..., n\}. \tag{5.10}$$

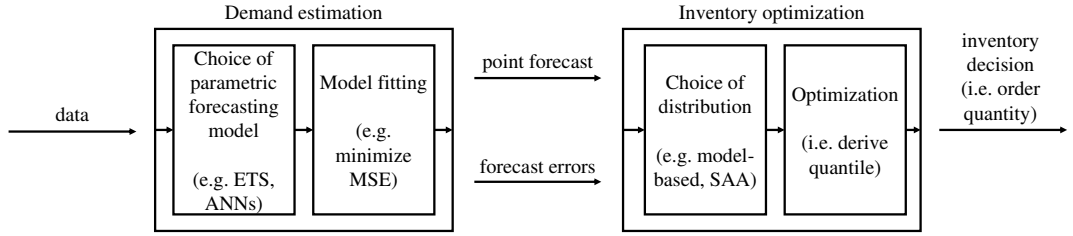
The objective function (5.7) minimizes the empirical underage and overage costs, while the constraints (5.8) to (5.10) ensure that deviations of the estimate from the actual demand are correctly assigned to underages and overages. By solving the problem for the empirical data $S_n = \{(d_1, \mathbf{x_1}), ..., (d_n, \mathbf{x_n})\}$, we obtain parameters $\mathbf{\Phi}^*$ for the ML method that minimize the empirical costs with respect to these data. Once the model has been trained, the resulting order quantity for period p is the quantile forecast with $q_p(\mathbf{\Phi}^*, \mathbf{x}_p)$.

Bertsimas and Kallus (2018) and Ban and Rudin (2018) showed that integrating forecasting in the optimization model is equivalent to the more general QR problem in Takeuchi et al. (2006). For a better understanding, we elaborate on this relation in more detail. The basic idea of QR is to estimate the unobservable quantile by modifying the loss function of a standard regression model. Minimizing the sum of squared errors $\sum_{i=1}^{n} (y_i - \hat{y_i})^2$ yields the mean, while minimizing the sum of absolute errors $\sum_{i=1}^{n} |y_i - \hat{y_i}|$ yields the median. By weighting the underages with the quantile $\tau \in (0, 1)$ and overages with $(1 - \tau)$, thus $\sum_{i=1}^{n} \tau (y_i - \hat{y_i})^+ + (1 - \tau)(\hat{y_i} - y_i)^+$, we obtain an estimate for the quantile (Koenker, 2005). The optimal solution of the newsvendor model is the quantile $\tau = \frac{c_u}{c_u + c_o}$ of the demand distribution; thus, $(1 - \tau) = \frac{c_o}{c_u + c_o}$. Inserting these values of τ and $(1 - \tau)$ into the objective function of the quantile regression yields the optimization problem (5.7).

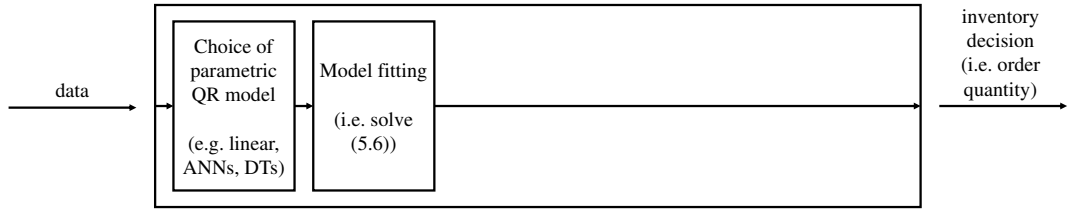
The main advantage of QR over the model-based approach and SAA is its ability to model conditional quantiles under heteroscedasticity and for unknown error distributions. However, the performance of the approach depends crucially on the underlying model q. On the one hand, if q is too simplistic (e.g. linear), the model might not be able to capture the structure in the training data. On the other hand, if q is too complex, there is a risk of overfitting the model.

5.3.1.5 Summarizing the three Levels of data-driven Inventory Management

We conclude this section by linking our methodology explained in Subsections 5.3.1.2 - 5.3.1.4 to our framework of data-driven inventory management introduced in Figure 5.1. To this end, Figure 5.2 positions each piece of our methodology in the framework.



(a) Separate estimation and optimization



(b) Integrated estimation and optimization

Figure 5.2: Relating our methodology to the three levels of data-driven inventory management.

- On the first level (demand estimation), we choose a parametric forecasting model (e.g. ETS or ANN). For the ML models, this includes the selection and optimization of hyper-parameters (e.g. number of layers of ANNs). We then use the data to fit the model by optimizing its parameters in order to minimize a certain objective function (i.e. MSE). The outputs of the first level of data-driven inventory management are a point demand forecast and the resulting empirical error distribution.

- On the second level (inventory optimization), we operationalize a model-based approach by fitting a normal distribution and distinguish it from a data-driven (SAA) approach. We then optimize by selecting a certain quantile of the respective demand distribution. This gives us the resulting order quantity.

- On the third level (integrated estimation and optimization), we choose a parametric QR model (e.g. ANNs) and fit its parameters by solving problem (5.6) instead of minimizing the MSE.

From the existing literature, it is not yet clear how the choices on each of the three levels affect performance. In the following, we investigate this question empirically.

5.3.2 Empirical Evaluation

Our empirical evaluation aims to assess the impact of data-driven approaches for the three levels – (1) demand estimation, (2) optimization, and (3) integrated estimation and optimization – on average costs for the newsvendor problem. To this end, we evaluate the performance of the methods with respect to costs by using a real-world dataset to compare it to various standard approaches.

5.3.2.1 Data

We evaluate the proposed approaches using dataset v2 (see Section 3.2). We need to highlight that observed sales are not necessarily equal to demand, as stock-outs occur and lead to censored demand information (Conrad, 1976). In order to estimate the daily demand in the case of a stock-out, intra-day sales patterns of point-of-sales data are leveraged (Lau and Lau, 1996).

The dataset comprises eleven stock-keeping units, namely, six buns and five breads, for five stores over a period of 88 weeks, where each store is open from Monday to Saturday. This configuration amounts to 55 ordering decisions per day. Additionally, we enrich the dataset with external explanatory features related to calendar, weather, and location of the store (see Table 3.1). We split the dataset into a training set containing up to 63 weeks and a test set containing the remaining 25 weeks (see Table 5.1). We perform a rolling 1-step-ahead prediction evaluation on the test set in order to assess the performance of the methods. We fit the models and distribution parameters every 10 days on a rolling training dataset with a constant size. Due to computational constraints, we fit the parameters of the ANNs every 50 days only. To evaluate the effect of the amount of available data, we use different sample sizes for the training set. The full training set (sample size 1.0) covers 63 weeks, while the smallest training set (sample size 0.1) contains only 6 weeks (see Table 5.1).

Sample	**1.0**	**0.8**	**0.6**	**0.4**	**0.2**	**0.1**
train length (days)	378	300	228	150	78	36
test length (days)	150	150	150	150	150	150

Table 5.1: Training & test periods for different sample sizes.

While traditional time series methods such as exponential smoothing or ARIMA are able to process only a single times series at a time, a major advantage of the ML methods is their ability to deal with a large number and variety of features. In order to leverage this advantage, we do not only train them with a single time series per product but alternatively also across products and stores. In the latter case, we also include the features listed in Table 3.1.

5.3.2.2 Experimental Design

In our experiment, we evaluate the impact of different (1) estimation, (2) optimization, and (3) integrated estimation and optimization approaches on the costs of the newsvendor model.

We start by assessing the impact of forecast performance. In addition to the ANNs and DTs introduced in the previous section, we evaluate six different reference forecasting methods, which we outline in the next section. For each forecasting method, we measure the forecast accuracy (Section 5.3.2.4) and then investigate its impact on costs (Section 5.3.2.5.1). Second, we compare the model-based optimization assuming a normal distribution (*Norm*) with the data-driven optimization using *SAA*. To this end, we calculate the average costs for different target service levels (Section 5.3.2.5.2). Third, we assess the performance of the integrated estimation and optimization approach with QR and compare it to the separate approaches (Section 5.3.2.5.3). Fourth, we evaluate the sensitivity to the sample size in order to assess the value of a large training set (Section 5.3.2.5.5). Overall, the database of the evaluation results comprises more than 9.1 million entries, i.e., close to 0.6 million point forecasts and approximately 8.6 million order quantities. We employ the Wilcoxon signed-rank test to test the statistical significance of our results at the 5% significance level.

5.3.2.3 Reference Methods and ML Setup

In order to evaluate the ML approaches, we compare them to well-established forecasting methods. With the exception of the first approach (*Median*), we rely on methods that are explicitly able to model seasonal time series because the demand for baked goods exhibits a strong weekly seasonality (see Section 3.3).

5.3.2.3.1 Reference Methods

The evaluated reference methods comprise *Median*, *S-Median*, *S-Naïve*, *S-MA*, *ETS*, and *S-ARIMA*. A brief description of the methods is contained in Section 2.1. However, *Median*, which is included in our comparison in order to evaluate the benefit of seasonal demand models, and *S-Median* consider the entire training set instead of only the few last observations. For the method *S-MA*, we determine k in the range from 3 to 12 based on the last 20% of the training set for each time series. We choose the value of k that minimizes the sum of squared errors.

5.3.2.3.2 ML Setup

In this subsection, we describe the setup of methods that take multiple time series and additional features (see Table 3.1) into account. For these methods, we also evaluate the integrated estimation and optimization approach introduced in Section 5.3.1.4.

Linear regression The linear regression model uses lagged demand data (lags: 1, 2, ..., 6, 12, 18) which are linearly scaled between 0 and 0.75 as input. The weekly seasonality is modeled through binary variables. When all time series across stores and products and the extended feature set are used for the prediction, further variables are introduced. In order to avoid overfitting, we include a regularization term in the objective function. The integrated linear approach is equivalent to the models in Beutel and Minner (2012) and Ban and Rudin (2018).

ANNs We employ an ensemble of 50 ANNs with the *median* ensemble operator, as this approach is robust to the initial weights and provides reliable results (Barrow et al., 2010; Kourentzes et al., 2014). Several hyper-parameters (learning rate, batch size, number of hidden nodes, activation function of hidden layer) are optimized by a random search (Bergstra and Bengio, 2012) in combination with cross-validation on the training set. As activation function for the output layer, we use a linear function, which is reasonable for regression with ANNs (Zhang et al., 1998). The input consists of lagged demand information (lags: 1, 2, ..., 6, 12, 18), which is linearly scaled between 0 and 0.75, and trigonometric functions of the weekday in order to enforce the weekly seasonality. Hence, the provided information is similar to what other seasonal methods consider. When all time series across products and stores are considered, we enrich the dataset with further explanatory features (see Table 3.1).

DTs We use Microsoft's LightGBM implementation (Ke et al., 2017a) of gradient boost decision trees. Similar to the ANNs, several hyper-parameters (learning rate, number of leaves, minimum amount of data in one leaf, maximum number of bins, maximum depth of each tree) are selected based on a random search within the training data (Bergstra and Bengio, 2012). The number of trees is controlled by early stopping, which also reduces the risk of overfitting. We consider the same features as in the other ML methods.

5.3.2.4 Point Forecast Analysis

The relevant performance measure of the newsvendor model is overall costs (overage and underage). Before evaluating the impact of the different estimation and optimization approaches on cost in Section 5.3.2.5, we separately measure the accuracy of the point forecasts in order to relate it to overall costs in the subsequent analysis.

For each forecasting method introduced in the previous section, we compute a set of common accuracy measures (see Section 2.3), including the Mean Percentage Error (MPE), Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE) (Hyndman and Koehler, 2006), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Relative Absolute Error (RAE). Table 5.2 shows the average forecast accuracy over all time series by method.

Not surprisingly, the worst accuracy is achieved by the *Median* forecast, which is the only method that does not incorporate the weekly seasonality pattern. The results improve noticeably (more than 5 percentage points in MAPE) when the weekly seasonality is considered (*S-Median*). *S-Median* is also more robust against sudden changes in demand and provides more reliable results than *S-Naïve*. *S-MA* outperforms all baseline methods (*Median*, *S-Median*, *S-Naïve*) and its accuracy is even competitive to more sophisticated approaches. It is not as prone to outliers but follows minor level shifts. Overall, *ETS* is the best method compared to models that are trained on a single time series as it captures the main characteristics of the time series by computing the weighted average of past observations. Even the more complex

Method	MPE	SMAPE	MAPE	MASE	RMSE	MAE	RAE
Median	-22.34	29.71	39.43	1.01	39.89	15.70	1.72
S-Median	-21.45	24.74	33.73	0.82	28.42	11.99	1.31
S-Naïve	**-11.84**	28.71	34.86	0.92	27.80	12.56	1.37
S-MA	-14.61	23.32	30.15	0.75	22.27	10.14	1.11
ETS	-12.47	22.19	28.47	0.71	21.83	9.66	1.06
S-ARIMA	-14.35	22.88	29.71	0.73	21.40	9.87	1.08
Linear	-18.73	23.75	32.07	0.77	23.43	10.54	1.15
DT-LGBM	-18.80	22.88	31.13	0.73	21.98	9.92	1.08
ANN-MLP	-14.73	22.63	29.59	0.72	21.28	9.75	1.07
Linear (all)	-14.33	22.14	29.18	0.71	21.23	9.63	1.05
DT-LGBM (all)	-13.44	21.51	28.34	**0.68**	**20.06**	**9.15**	**1.00**
ANN-MLP (all)	**-12.62**	**21.42**	**27.87**	**0.68**	**20.09**	**9.16**	**1.00**

Table 5.2: Forecast performance of the point predictions (sample size: 1.0). The best performance for each metric is underlined. Results that do not differ from the one of the best method at a significance level of 5% for each metric are printed in bold face.

ML approaches cannot improve the forecast. However, when trained across stores and products with additional features, the ML methods further improve significantly. *ANN-MLP* and *DT-LGBM* also outperform ETS. The information contained in the features and supplementary time series has additional explanatory potential that is effectively extracted by all three ML approaches.

We note that the negative MPE throughout all methods indicates that in the test data, there are low-demand events that cannot be foreseen by the models based on historical demand. These low-demand events are more frequent, more extreme, or both during the test period than events of unexpectedly high demand. This observation might be due to the fact that situations with very low demand (e.g. supply disruption, partial shop closing, and construction) are more likely than situations with extremely high demand.

5.3.2.5 Inventory Performance Analysis

The purpose of the newsvendor model is to determine the cost-minimal order quantity by considering demand uncertainty and underage and overage costs. In order to perform a comprehensive analysis of the introduced methods, we calculate the order quantities and compute the resulting average costs for each approach. As underage and overage cost may vary among products and stores, we analyze multiple target service levels. The target service level $c_u/(c_u + c_o)$ is the optimal probability of having no stock-out during the day. In the repeated newsvendor model, this corresponds to the long-run fraction of periods in which demand is fully satisfied. By setting the unit price and the sum of underage and overage costs (c_u + c_o) to 1.00 and varying their relative share, we obtain six different target service levels. This process allows us to interpret c_u as the profit margin and c_o as the unit costs (e.g. material and production costs) of an item. In order to compare the different methods, we measure the performance relative to the best method for each target service level. Additionally, we report

the realized average service level for each approach. We calculate the realized service level as the relative share of days on which total demand was met. A large deviation of the realized service level from the target service level indicates that a method tends to overestimate or underestimate the optimal order quantity. Note that the reported service level just serves to characterize the solution by relating it to the newsvendor solution. It does not reflect a cost-service trade-off since costs include both overage and underage costs. The results are reported in Table 5.3.

In the following sections, we analyze the effects of (1) demand estimation, (2) optimization, and (3) integrated estimation and optimization on average costs and observed service levels. Furthermore, we evaluate the sensitivity of the results to the size of the available sample.

5.3.2.5.1 The Effect of Demand Estimation

To evaluate the effect of demand estimation on costs, we compare the average cost of the different estimation approaches for each target service level in Table 5.3. The best approach for each target service level is underlined. We see that the approaches based on the ML forecasts that use data across stores and products and additional features (*all*) provide the lowest average costs for all target service levels. The performance of *ANN-MLP* and *DT-LGBM* is very similar, while methods based on the *Linear* forecast yield higher costs. An interesting result is that *ETS* performs best when training is restricted to single time series. This is particularly noteworthy when considering its computational efficiency compared to the ML methods. Overall, we observe that approaches based on accurate estimation methods achieve significantly lower costs, independent of the optimization approach. Thus, the level of demand estimation has a substantial impact on overall performance.

In order to further substantiate this statement, we conduct a correlation analysis. We compute the Spearman's rank correlation coefficient ρ between costs and forecast accuracy (SMAPE and RMSE) for each store-article-service level combination. The results are depicted in Table 5.4.

The analysis supports the claim that the general ranking of methods with respect to costs is similar to the ranking with respect to forecast accuracy, with a median ρ of 0.8799 for the rank correlation of costs and SMAPE and 0.9406 for the median rank correlation of costs and RMSE. The reason for this observation is that more accurate point predictions lead to more precise demand distribution estimates, which make the succeeding optimization phase less crucial.

We complement the above cost analysis by looking at the realized service levels which provide further insights into the order quantities obtained from the different methods. Table 5.4 also shows the Spearman Correlations between the absolute service level deviations (i.e. difference between average observed service level and the newsvendor target service level) and costs and forecast errors, respectively. From Table 5.3, we can see that all methods overachieve the target service level on average. This matches our observation of Section

	Method		TSL = 0.5		TSL = 0.6		TSL = 0.7		TSL = 0.8		TSL = 0.9		TSL = 0.95	
	Estimation	**Optimization**	**Δ Cost**	**SL**	**Δ Cost**	**SL**	**Δ Cost**	**SL**	**Δ Cost**	**SL**	**Δ Cost**	**SL**	**Δ Cost**	**SL**
Benchmarks	Median	Norm	72.5%	0.61	83.9%	0.72	93.2%	0.79	99.4%	0.86	97.8%	0.92	92.0%	0.95
		SAA	72.5%	0.61	79.4%	0.70	87.9%	0.79	99.6%	0.87	109.5%	0.94	101.9%	0.98
	S-Median	Norm	31.8%	0.64	33.5%	0.74	34.7%	0.83	34.9%	0.89	32.2%	0.95	29.6%	0.97
		SAA	31.8%	0.64	30.3%	0.72	30.4%	0.80	29.5%	0.88	27.4%	0.95	31.2%	0.98
	S-Naive	Norm	38.0%	0.51	37.5%	0.63	37.0%	0.75	37.3%	0.85	37.2%	0.93	37.2%	0.96
		SAA	38.4%	0.51	37.6%	0.61	35.6%	0.71	34.3%	0.81	32.2%	0.91	33.3%	0.96
	S-MA	Norm	11.5%	0.56	13.6%	0.68	16.0%	0.78	17.6%	0.86	18.3%	0.94	16.6%	0.97
		SAA	10.5%	0.52	11.0%	0.62	11.4%	0.73	11.7%	0.82	12.2%	0.92	13.9%	0.96
	ETS	Norm	6.1%	0.53	6.7%	0.64	7.0%	0.74	7.1%	0.83	5.6%	0.91	5.7%	0.95
		SAA	6.2%	0.50	6.5%	0.61	6.7%	0.71	6.7%	0.80	5.6%	0.90	5.9%	0.95
	S-ARIMA	Norm	8.5%	0.55	8.9%	0.65	8.8%	0.75	8.3%	0.84	7.5%	0.92	7.2%	0.95
		SAA	8.0%	0.52	8.1%	0.62	8.0%	0.71	7.7%	0.81	6.5%	0.91	7.2%	0.95
ML single time series	Linear	Norm	15.8%	0.58	17.7%	0.69	19.6%	0.78	20.8%	0.85	20.2%	0.93	20.9%	0.95
		SAA	15.6%	0.56	17.2%	0.66	18.9%	0.75	20.1%	0.84	20.7%	0.93	21.8%	0.96
		QR	10.6%	0.54	10.7%	0.64	11.4%	0.73	11.2%	0.82	11.8%	0.91	18.9%	0.96
	DT-LGBM	Norm	9.0%	0.60	8.6%	0.68	8.5%	0.76	8.8%	0.83	10.2%	0.89	**15.2%**	0.93
		SAA	7.9%	0.57	7.7%	0.65	7.8%	0.73	8.4%	0.81	10.0%	0.89	14.4%	0.94
		QR	11.1%	0.59	10.8%	0.68	12.1%	0.78	15.3%	0.85	20.8%	0.93	29.0%	0.96
	ANN-MLP	Norm	7.2%	0.55	8.4%	0.66	9.0%	0.75	9.6%	0.83	9.4%	0.91	10.5%	0.95
		SAA	6.6%	0.52	7.6%	0.63	8.2%	0.72	8.6%	0.82	8.6%	0.91	10.2%	0.95
		QR	7.5%	0.53	7.9%	0.64	8.6%	0.73	9.8%	0.82	13.0%	0.91	18.1%	0.95
ML pooled ts + features	Linear (all)	Norm	5.9%	0.53	5.5%	0.64	5.6%	0.75	6.1%	0.84	4.9%	0.91	4.0%	0.95
		SAA	5.4%	0.51	5.3%	0.62	5.0%	0.72	5.3%	0.82	5.2%	0.91	4.9%	0.95
		QR	5.1%	0.52	4.5%	0.62	5.2%	0.72	7.2%	0.81	10.0%	0.90	12.8%	0.95
	DT-LGBM (all)	Norm	**0.6%**	0.53	**0.4%**	0.62	0.0%	0.71	0.1%	0.80	0.4%	0.87	2.1%	0.92
		SAA	0.9%	0.51	**0.4%**	0.61	**0.0%**	0.69	**0.0%**	0.79	0.2%	0.88	1.7%	0.92
		QR	1.6%	0.52	1.7%	0.61	1.6%	0.71	3.1%	0.80	6.4%	0.90	11.4%	0.94
	ANN-MLP (all)	Norm	0.7%	0.52	0.7%	0.63	0.7%	0.73	0.7%	0.82	**0.0%**	0.90	**0.0%**	0.95
		SAA	**0.3%**	0.51	**0.2%**	0.61	0.3%	0.72	0.4%	0.81	0.4%	0.90	1.5%	0.95
		QR	**0.0%**	0.50	**0.0%**	0.61	0.9%	0.72	3.3%	0.82	6.8%	0.91	11.2%	0.95

Table 5.3: Inventory performance analysis: Average cost increase relative to the best approach and average service level (SL) for various target service levels (TSLs) and a sample size of 1.0. Methods denoted with *all* are trained on data across all products and stores. The best approach for each target service level is underlined. Results that do not differ from the one of the best method at a significance level of 5% for each service level are printed in bold face.

	Costs	SMAPE	RMSE
Costs	-	0.8799 ($\pm$ 0.1211)	0.9406 ($\pm$ 0.0481)
SL	0.4202 ($\pm$ 0.2879)	0.4253 ($\pm$ 0.2834)	0.3034 ($\pm$ 0.2678)

Table 5.4: Median of Spearman's Correlations ($\pm$ standard deviation) between absolute service level deviation (SL), costs, and forecast accuracy (SMAPE, RMSE).

5.3.2.4, that all forecasting methods overestimate the demand on average, due to events with unexpectedly low demand in the test data.

We further see that the correlation between the absolute service level deviation and costs is relatively low (0.4202). This shows that the ability of a method to achieve a desired service level on average is not a very good indicator for the cost performance of that method. The service level measures only whether or not there was a stock-out and thus indicates the direction of the deviation from the optimal order quantity on average. It does not take into account the order of magnitude of overages and underages. The low correlation between the forecast accuracy measures and the service level deviation confirms this conclusion.

5.3.2.5.2 The Effect of Optimization

To assess the impact of model-based vs. data-driven optimization on costs, we compare the average cost of *Norm* and *SAA* for each estimation method and target service level. We perform a Shapiro-Wilk test on the residuals of the forecasts of S-ARIMA and ETS and find that for approximately one quarter of the time series the residuals are normally distributed at 95% confidence level. Thus, the normal distribution assumption can be justified, although one cannot expect that all residuals follow the distribution assumption in a real-world data set. We observe that the performance differences between *SAA* and *Norm* are relatively small, and the effect of accurate demand estimation clearly outweighs the effect of data-driven optimization. However, for the majority of estimation methods, *SAA* leads to lower costs than *Norm* for target service levels up to 0.9, while the normal distribution assumption can be beneficial for higher service levels.

The good performance of *SAA* and its weaknesses for higher service levels are in line with the theoretical results of Levi et al. (2015). The authors provide a bound on the accuracy of *SAA* for the newsvendor model (Theorem 2 *Improved LRS Bound*) that does not rely on assumptions on the demand distribution. The bound has an exponential rate that is proportional to the sample size and $\min(c_u, c_o)/(c_u + c_o)$. In our case, the bound implies that using *SAA*, in order to obtain the same accuracy for a service level of 0.9 (0.95) as for a service level of 0.8, we would need 1.5 (4) times more data. However, in the bakery industry, such high service levels are not common, and our dataset is sufficient to let *SAA* outperform *Norm* for service levels up to 0.9 for most approaches.

5.3.2.5.3 The Effect of Integrated Estimation and Optimization

We also employ the QR approach that integrates the demand estimation into the optimization

model for the *linear* approach and the ML methods *DT-LGBM* and *ANN-MLP*. In order to focus on the effect of integrated estimation and optimization, we compare *QR* to *SAA* for the respective approaches. For *DT-LGBM* and *ANN-MLP* trained on single time series, *QR* performs worse than *SAA*, while *Linear QR* outperforms *SAA*. For high service levels *QR* generally performs relatively poor for all three estimation approaches. When trained on data across stores and products and including features, integration of estimation and optimization improves the performance of *Linear (all)* and *ANN-MLP (all)* for low service levels. However, for high target service levels, *SAA* and *Norm* perform better than *QR* for all estimation approaches.

The theoretical advantage of the *QR* approach is its ability to estimate *conditional* quantiles that depend on the features (see Figure 5.3). The observation that for the approaches trained only on single time series, *QR* is not beneficial, might be explained by the fact that too little features are available to leverage the feature-dependency of the quantile. The previous statement is supported by the fact that *Linear (all)* and *DT-LGBM (all)* improve through integration at low service levels as more data are available and feature-dependent variance can be estimated more accurately. However, this theoretical advantage cannot be observed for higher service levels. We suspect that more extensive hyper-parameter optimization in combination with alternative scaling of the input data for each individual target service level might improve the performance.

Our results for the single time series case are in line with the outcome of the empirical analysis of Ban and Rudin (2018) who also report that separate estimation and optimization outperforms the linear integrated approach on their relatively small dataset of one year. We observe that this effect gets smaller when the models are trained with pooled time series and features.

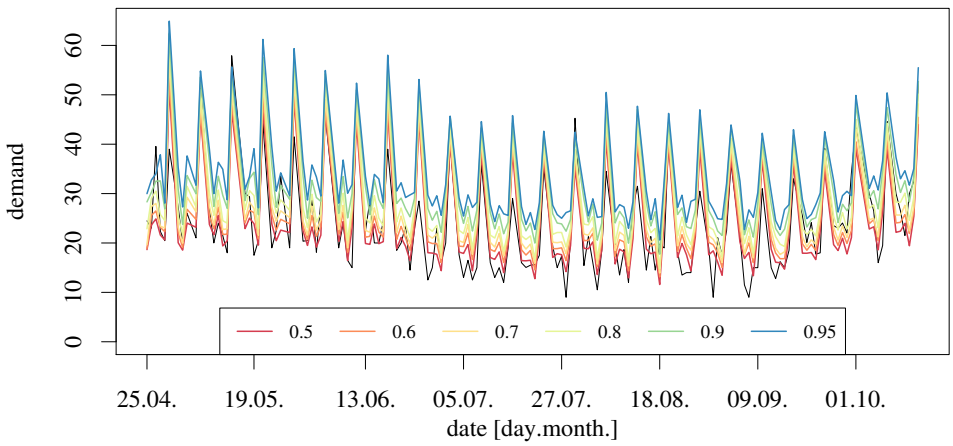


Figure 5.3: Forecasts for different service levels using ANN QR.

5.3.2.5.4 The Effect of Learning across Products and external Features

Our dataset comprises sales data of several breads and buns across multiple stores. These products are relatively similar to one another and therefore one time series might contain

information about the other. Univariate time series models can only consider a single product at a time, while ML methods are able to process a large number of inputs. Therefore, we train *linear (all)*, *DT-LGBM (all)*, and *ANN-MLP (all)* across all products and stores. The pooling of training data also makes it possible to enhance the data set with a large number of additional features that cannot be employed if the models are trained per time series.

From Table 5.3, we observe that indeed all ML methods benefit from the additional data and improve significantly. *DT-LGBM (all)* and *ANN-MLP (all)* perform similarly and outperform all other methods. We note that a similarity of time series is not specific to our case but can be found in many retail settings.

5.3.2.5.5 Sensitivity to Sample Size

The power of the data-driven approaches lies in their ability to leverage large amounts of available data, which makes them very flexible but may limit their deployability if not enough data is available. In order to determine the dependency of the different approaches on data availability, we vary the size of the training data and compare the results on a fixed test set (see Table 5.1). The results of this experiment are given in Table 5.5 and depicted in Figure 5.4 for the data-driven approaches. We present only the results for target service level 0.7, noting that the qualitative results also apply to the other service levels.

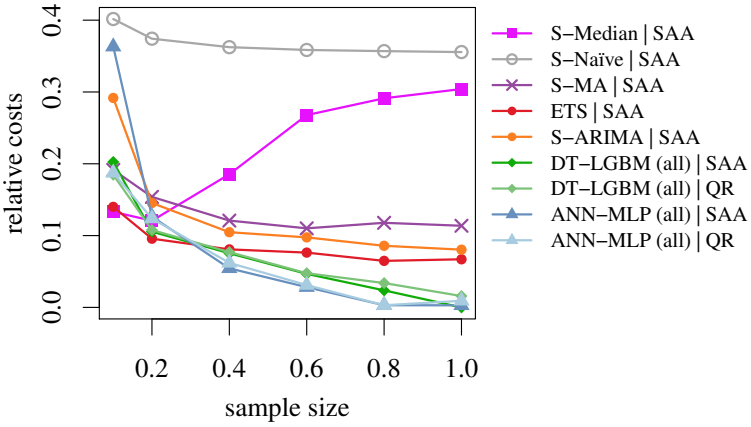


Figure 5.4: Effect of the sample size (TSL = 0.7).

Based on our results, the methods can be divided into three groups: The first group consists of methods whose performance hardly depends on the sample size. In our case, this includes methods based on the *S-Naïve* forecast. The *S-Naïve* approaches simply forecast the demand of the same weekday of the week before. Thus, it does not improve as more data becomes available. The second group consists of methods whose performance diminishes as more training data become available. The approaches with a *Median* (not depicted in Figure 5.4, see Table 5.5) and *S-Median* forecast are part of this group. The costs increase as more training data are available and as more "outdated" data are included. In our real-world case, this observation implies that, for example, demand data from Winter is used to estimate the median forecast for Summer although these data are not representative of this season. The

	Method		S = 0.1		S = 0.2		S = 0.4		S = 0.6		S = 0.8		S = 1.0	
	Estimation	**Optimization**	**△ Cost**	**SL**	**△ Cost**	**SL**	**△ Cost**	**SL**	**△ Cost**	**SL**	**△ Cost**	**SL**	**△ Cost**	**SL**
Benchmarks	Median	Norm	79.5%	0.72	82.0%	0.75	87.1%	0.79	92.7%	0.80	92.7%	0.80	93.2%	0.79
		SAA	77.2%	0.70	78.2%	0.74	82.6%	0.78	88.0%	0.79	87.7%	0.79	87.9%	0.79
	S-Median	Norm	13.5%	0.70	15.5%	0.76	23.2%	0.81	32.5%	0.84	33.6%	0.83	34.7%	0.83
		SAA	13.3%	0.65	12.1%	0.72	18.5%	0.79	26.8%	0.81	29.1%	0.80	30.4%	0.80
	S-Naive	Norm	36.7%	0.72	36.5%	0.73	36.8%	0.75	37.5%	0.75	37.1%	0.75	37.0%	0.75
		SAA	40.2%	0.68	37.4%	0.69	36.2%	0.70	35.9%	0.71	35.7%	0.71	35.6%	0.71
	S-MA	Norm	15.8%	0.73	14.6%	0.75	15.1%	0.77	15.9%	0.78	15.6%	0.78	16.0%	0.78
		SAA	19.2%	0.67	15.4%	0.67	12.1%	0.70	11.0%	0.72	11.8%	0.72	11.4%	0.73
	ETS	Norm	**12.1%**	0.70	**8.8%**	0.73	7.9%	0.74	7.7%	0.74	6.6%	0.74	7.0%	0.74
		SAA	14.0%	0.66	9.6%	0.68	8.1%	0.69	7.6%	0.70	6.5%	0.71	6.7%	0.71
	S-ARIMA	Norm	25.8%	0.69	13.8%	0.72	10.7%	0.74	10.3%	0.75	9.2%	0.75	8.8%	0.75
		SAA	29.2%	0.64	**14.5%**	0.68	10.5%	0.68	9.8%	0.70	8.6%	0.72	8.0%	0.71
ML single time series	Linear	Norm	29.8%	0.70	18.0%	0.71	18.9%	0.76	21.1%	0.79	20.4%	0.78	19.6%	0.78
		SAA	30.2%	0.68	18.3%	0.70	18.1%	0.74	20.1%	0.76	19.3%	0.76	18.9%	0.75
		QR	32.1%	0.67	17.4%	0.68	14.3%	0.71	13.2%	0.73	12.0%	0.73	11.4%	0.73
	DT-LGBM	Norm	50.4%	0.72	22.6%	0.73	13.6%	0.77	12.5%	0.79	10.4%	0.78	8.5%	0.76
		SAA	48.7%	0.68	22.6%	0.68	11.1%	0.73	10.2%	0.75	8.5%	0.75	7.8%	0.73
		QR	52.4%	0.73	27.3%	0.75	17.7%	0.78	16.8%	0.80	13.6%	0.79	12.1%	0.78
	ANN-MLP	Norm	33.2%	0.69	14.3%	0.73	13.9%	0.78	12.0%	0.78	9.9%	0.78	9.0%	0.75
		SAA	33.5%	0.70	14.9%	0.72	12.4%	0.74	10.2%	0.75	8.4%	0.75	8.2%	0.72
		QR	33.1%	0.67	17.8%	0.71	12.5%	0.75	10.7%	0.75	9.2%	0.75	8.6%	0.73
ML pooled ts + features	Linear (all)	Norm	**16.0%**	0.69	**14.2%**	0.70	14.8%	0.73	8.8%	0.74	8.2%	0.75	5.6%	0.75
		SAA	**17.2%**	0.64	14.5%	0.66	13.5%	0.69	7.3%	0.71	7.6%	0.72	5.0%	0.72
		QR	**15.3%**	0.65	12.7%	0.67	13.3%	0.69	7.7%	0.71	7.1%	0.72	5.2%	0.72
	DT-LGBM (all)	Norm	**21.1%**	0.67	10.4%	0.69	**8.0%**	0.69	**5.5%**	0.71	2.5%	0.70	0.0%	0.71
		SAA	**20.3%**	0.65	10.6%	0.66	**7.5%**	0.67	**4.7%**	0.69	**2.4%**	0.68	**0.0%**	0.69
		QR	18.5%	0.71	10.8%	0.69	7.7%	0.70	**4.8%**	0.71	3.4%	0.70	1.6%	0.71
	ANN-MLP (all)	Norm	33.5%	0.60	12.4%	0.69	6.1%	0.69	3.6%	0.73	0.7%	0.71	0.7%	0.73
		SAA	36.3%	0.58	12.6%	0.66	**5.4%**	0.67	**2.8%**	0.71	**0.3%**	0.69	0.3%	0.72
		QR	**18.7%**	0.63	12.4%	0.65	6.2%	0.68	**3.1%**	0.70	0.3%	0.70	0.9%	0.72

Table 5.5: The effect of the sample size: Average cost increase relative to the best approach (over all sample sizes) and average service level (SL) for the target service level 0.7 and various sample sizes (S). Methods denoted with *all* are trained on data across all products and stores. The best approach for each sample size is underlined. Results that do not differ from the one of the best method at a significance level of 5% for each sample size are printed in bold face.

third group consists of methods whose performance improves as more data become available. This group comprises the ML methods proposed in this study. We also include methods based on *S-ARIMA*, *ETS*, and *linear* forecast in this group. However, the performance of *S-ARIMA* and *ETS* stagnates for sample sizes larger than 0.6. This effect might be due to the fact that we use a little over one year of training data and consequently some months are included twice. It seems that the ML approaches can account for this matter. Thus, in the present application, the purely data-driven approaches benefit most from a large training set.

Comparing the different optimization methods, we find that with a sample size of $S = 0.4$ (150 days) and larger, the data-driven *SAA* method yields lower costs than its model-based counterpart *Norm* for most forecasting methods at a service level of 0.7. This observation implies that a normal distribution assumption is beneficial in our case only if a very limited dataset is available or if the target service level is very high (see Section 5.3.2.5.2).

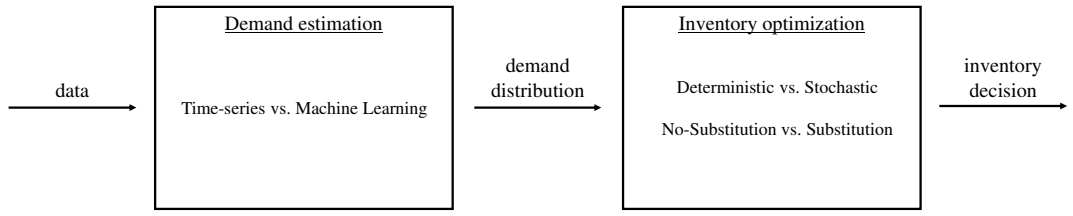
The performance and the ranking of the methods varies depending on the sample size. However, if more data are available, it is possible to employ a method that reduces the costs compared to the best method on the smaller dataset. For sample size 0.1, *ETS Norm* is the best approach, while costs can be reduced by 17.4% using *DT-LGBM Norm* with a sample size of 1.0.

5.4 Multi-Product Newsvendor with Substitution

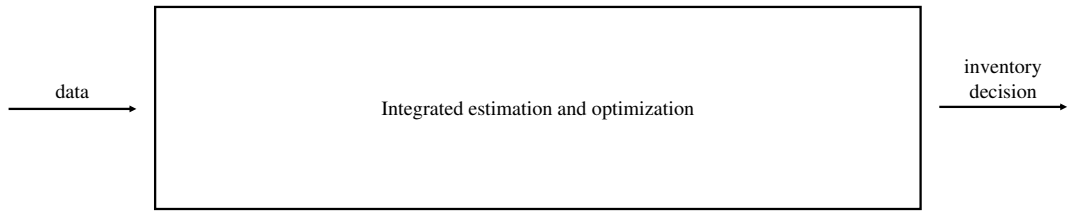
In this section, we propose and analyze methods for the joint optimization of order quantities of substitutable products. Stock-outs are common in many retail settings. When customers cannot find their preferred product in stock, they might choose a similar product instead (Gruen et al., 2002; van Woensel et al., 2007). This substitution behavior makes inventory optimization for multi-product portfolios especially challenging due to the resulting interdependencies of stocking decisions.

The objective of the inventory optimization problem is to maximize the overall profit by setting appropriate inventory levels for each product. While the classical single-product newsvendor problem is well-solved, the multi-product version is known to be notoriously harder (Netessine and Rudi, 2003; Schlapp and Fleischmann, 2018). The OR literature on this problem is mainly concerned with establishing theoretical properties (Parlar and Goyal, 1984; Netessine and Rudi, 2003; Schlapp and Fleischmann, 2018) and developing efficient solution algorithms for the problem (Farahat and Lee, 2018; Zhang et al., 2018). Due to the complex interactions of ordering decisions not many papers exist that study the value of considering the multi-product nature of the problem instead of treating each product independently in a real-world setting (Kök and Fisher, 2007; Sachs, 2015). By and large, the interaction with model parameter estimation and selection of estimation methods remains unmentioned. In the inventory optimization step, there are two important modeling questions (see Figure 5.5). First, should uncertainty be considered or is a deterministic approach sufficient? Second, should substitution be considered or is a single-product approach sufficient?

Additionally, we propose a novel integrated estimation and optimization (IEO) method for the multi-product newsvendor problem that builds on an ANN. Integrated approaches for the single-product newsvendor problem have been introduced (Beutel and Minner, 2012; Ban and Rudin, 2018; Huber et al., 2019). To the best of our knowledge, such an approach does not exist for the multi-product version for more than two products. Thus, the question remains open how such an approach could look like. Furthermore, it is unclear whether the potential benefits and drawbacks of integrated estimation and optimization of the single-product problem carry over to the multi-product case.



(a) Separate estimation and optimization (SEO)



(b) Integrated estimation and optimization (IEO)

Figure 5.5: The three levels of data-driven inventory management (Huber et al., 2019).

The remainder of this section structured as follows: Section 5.4.1 contains the formal problem description of the multi-product newsvendor problem with unknown demand distributions and an introduction of two solution approaches to the problem. In Section 5.4.2, we report and discuss the results of the empirical evaluation of the different methods.

5.4.1 Methodology

We study the multi-product newsvendor problem with stock-out based substitution where the demand distributions are unknown. To begin with, we follow the prevalent model formulation in the inventory management literature (Netessine and Rudi, 2003; Kök et al., 2015; Schlapp and Fleischmann, 2018).

5.4.1.1 Problem Formulation

Consider a retailer selling n partially substitutable products with uncertain demand D_i of product i over a finite selling season. The retailer must choose the order quantity q_i of product i before the selling season, such that expected profits Π are maximized. Each unit of product

i can be sold for price p_i. The unit cost is c_i. Products that could not be sold at the end of the season are left over and have a unit salvage value of s_i. Naturally, $p_i > c_i > s_i \geq 0$.

In order to model substitution, we assume that a fraction $\alpha_{ji} \in [0, 1]$ of customers that cannot find their preferred product j in stock (i.e. $D_j > q_j$) will substitute to product i, where $\sum_{i \neq j} \alpha_{ji} \leq 1$. This substitution behavior results in an inflation of the *initial* demand D_i of product i and the *substitution* demand of product i becomes $D_i^s = D_i + \sum_{j \neq i} \alpha_{ji}(D_j - q_j)^+$.

Thus, the retailer's objective is to maximize expected profits according to

$$\max_{q_i \geq 0} \Pi = \mathbb{E} \sum_i \left[u_i D_i^s - u_i (D_i^s - q_i)^+ - o_i (q_i - D_i^s)^+ \right], \tag{5.11}$$

$$= \sum_i \left(u_i q_i - (u_i + o_i) \mathbb{E} \left[q_i - D_i^s \right]^+ \right), \tag{5.12}$$

where $u_i = p_i - c_i$ and $o_i = c_i - s_i$ are product i's underage and overage costs, respectively.

In the problem that we consider, the probability distributions of the initial demand D_i for every product i is not known to the decision maker a priori. Instead, historical data $S_T = \{(\mathbf{d}_1, \mathbf{x}_1), ..., (\mathbf{d}_T, \mathbf{x}_T)\}$ are available, where $\mathbf{d}_t = [d_{1,t}, ..., d_{n,t}]$ is a vector of historical demand realizations of all n products and $\mathbf{x}_t = [x_{1,t}, ..., x_{m,t}]$ is a vector of m covariates or *features* (e.g. store location, opening hours, weather data) in period t. A straight forward approach would be to estimate the demand distribution D_i for every product i from this data and after that optimize problem (5.11). We introduce an alternative solution approach that integrates the estimation into the optimization problem.

Next, we describe the traditional separate estimation and optimization (SEO) approach and our novel integrated estimation and optimization (IEO) approach for the multi-product newsvendor problem.

5.4.1.2 Separate Estimation and Optimization (SEO)

By and large, the inventory management literature on the multi-product newsvendor problem neglects the fact that the demand distributions of products are unknown to the decision maker (Netessine and Rudi, 2003; Kök et al., 2015; Schlapp and Fleischmann, 2018). The extant papers that address both, the inventory problem and the demand estimation problem, use a two-step procedure. First, estimating the demand distributions. Second, optimizing the inventory decisions based on the demand distributions (Kök and Fisher, 2007).

Estimation

If only historical demand data $\mathbf{d}_1, ..., \mathbf{d}_T$ are available, one can approximate the actual demand distributions with their empirical counterpart. This approach is called Sample Average Approximation (SAA). However, if additional feature data $\mathbf{x}_1, ..., \mathbf{x}_T$ (e.g. weekdays, opening hours, weather data) are available, the estimate might be improved because the *conditional* forecast can be more accurate (Ban and Rudin, 2018; Huber et al., 2019). Note, that we use

the empirical residuals as an estimate for future uncertainty in order to avoid problematic distribution assumptions.

In ML, the data $S_T = \{(\mathbf{d}_1, \mathbf{x}_1), ..., (\mathbf{d}_T, \mathbf{x}_T)\}$ is called *training set*. Based on the training set, a ML algorithm approximates a function $f(\cdot)$ by minimizing a certain *loss function* $\mathcal{L}$. The most common loss function is the L_2 norm. The resulting optimization problem is

$$\min_{\mathbf{W}} \mathcal{L} = \sum_t \left(\mathbf{d}_t - f(\mathbf{W}, \mathbf{x}_t)\right)^2, \tag{5.13}$$

where $\mathbf{W}$ are the parameters of the function $f(\cdot)$ that have to be optimized (e.g. coefficients of a linear regression model or the weight matrices of an ANN).

The estimate for the demand distributions for product i in period $T+1$ is then

$$\hat{D}_{i,T+1}(\psi) = \frac{1}{T} \sum_t \mathbb{I}\left[f_i(\mathbf{W}^*, \mathbf{x}_{T+1}) + \epsilon_{i,t} \leq \psi\right], \tag{5.14}$$

where $f_i(\mathbf{W}^*, \mathbf{x}_{T+1})$ is the point forecast for product i in period $T+1$ with optimized parameters $\mathbf{W}^*$ and feature vector $\mathbf{x}_{T+1}$, $\mathbb{I}$ is the indicator function and $\epsilon_{i,t} = d_{i,t} - f_i(\mathbf{W}^*, \mathbf{x}_t)$ is the tth residual of the estimation model (5.13).

Optimization

Once there is an estimate $\hat{D}_i$ for the demand distribution of each product i, we can attempt to solve the original multi-product newsvendor problem (5.11). This problem is known to be notoriously hard to solve due to the non-convexity of the objective function (Netessine and Rudi, 2003). Obtaining optimal analytical solutions seems unlikely and there are only few efficient solution algorithms. Zhang et al. (2018) developed two mixed-integer linear program (MILP) formulations of the problem that we will use throughout the study for the optimization part of the SEO solution approach:

$$\max_{q_i \geq 0} \sum_i \left(u_i q_i - (u_i + o_i) \frac{1}{n} \sum_t y_{i,t} \right) \tag{5.15}$$

subject to:

$$y_{i,t} \geq q_i - \hat{D}_{i,t} - \sum_{j \neq i} \alpha_{ji} v_{i,t} \qquad \forall i, t \tag{5.16}$$

$$v_{i,t} \leq \hat{D}_{i,t} - q_i + M_i z_{i,t} \qquad \forall i, t \tag{5.17}$$

$$v_{i,t} \geq \hat{D}_{i,t} - q_i - M_i z_{i,t} \qquad \forall i, t \tag{5.18}$$

$$v_{i,t} \leq \hat{D}_{i,t}(1 - z_{i,t}) \qquad \forall i, t \tag{5.19}$$

$$v_{i,t}, y_{i,t} \geq 0 \qquad \forall i, t \tag{5.20}$$

$$z_{i,t} \in \{0, 1\} \qquad \forall i, t \tag{5.21}$$

For expositional purposes, we introduce only the first formulation and refer the reader to Zhang et al. (2018) for more details. The authors reformulate the expectation in the objective function of the original problem (5.11) as a finite summation over the discrete demand distribution estimate. $y_{i,t} = q_i - \hat{D}_{i,t} - \sum_{j \neq i} \alpha_{ji}(\hat{D}_{j,t} - q_j)^+$ and $v_{i,t} = (\hat{D}_{i,t} - q_i)^+$ represent overages and underages, respectively. Constraints (5.16) to (5.20) make sure that these equations hold. In order to linearize the $(\cdot)^+$ functions, the authors introduce binary variables $z_{i,t}$, where $z_{i,t} = 1$ if $q_i \geq \hat{D}_{i,t}$ (i.e., $v_{i,t} = 0$) and $z_{i,t} = 0$ if $q_i < \hat{D}_{i,t}$ (i.e., $v_{i,t} = \hat{D}_{i,t} - q_i$). M_i is an upper bound for the order quantity q_i of product i.

5.4.1.3 Integrated Estimation and Optimization (IEO)

Alternatively to solving two optimization problems consecutively, namely the loss minimization problem (5.13) and the profit maximization problem (5.15), we integrate the estimation and optimization into one optimization problem. To this end, we express the order quantities $q_{i,t}$ as a function of the feature vector $\mathbf{x}_t$ and the parameters $\mathbf{W}$ of a ML approach (e.g. the weight matrix of an ANN).

$$\max_{\mathbf{W}} \frac{1}{n} \sum_t \sum_i \left(u_i q_{i,t}(\mathbf{W}, \mathbf{x}_t) - (u_i + o_i) \left(q_{i,t}(\mathbf{W}, \mathbf{x}_t) - D^s_{i,t} \right)^+ \right), \tag{5.22}$$

where $D^s_{i,t} = D_{i,t} + \sum_{j \neq i} \alpha_{ji}(D_{j,t} - q_{j,t}(\mathbf{W}, \mathbf{x}_t))^+$.

In the IEO approach, the demand uncertainty is feature dependent, while in the SEO approach only the mean of the demand distribution is feature dependent. This leads to a theoretical advantage of the IEO approach in situations where the actual demand distribution is indeed feature dependent. Another advantage of IEO compared to SEO is that IEO requires less computational effort as the inventory optimization step is omitted and only the loss minimization problem needs to be solved.

5.4.2 Empirical Evaluation

5.4.2.1 Data Description and Preparation

We evaluate the proposed approaches using dataset v3a (see Section 3.2). In this section, we describe our dataset and the data preparation process. The dataset comprises the hourly sales data of six most frequently sold stock-keeping units from the product categories buns and breads for nine stores over a period of 987 days. We enrich the dataset with external explanatory features related to calendar, weather, and location of the store (see Table 3.1). In order to apply and compare our optimization approaches, we need price and cost parameters, substitution rates, and daily demand data.

First, we report price and cost parameters for both categories in Tables 5.6 and 5.7. Unit prices can be directly observed, whereas unambiguous cost parameters cannot be obtained

due to varying cost accounting methods and parameters. We assume uniform unit costs for each category that are based on expert judgment.

Buns	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
Unit price [EUR]	0.30	0.50	0.60	0.50	0.50	0.50
Unit cost [EUR]	0.06	0.06	0.06	0.06	0.06	0.06

Table 5.6: Price and cost parameters for buns.

Breads	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
Unit price [EUR]	1.75	2.45	2.45	1.70	2.85	2.95
Unit cost [EUR]	0.60	0.60	0.60	0.60	0.60	0.60

Table 5.7: Price and cost parameters for breads.

Second, we estimate the substitution probabilities based on the methodology of Anupindi et al. (1998). As the assumption of stationary demand during the day does not hold in our case due to a strong intraday sales pattern, we apply the approach to each hour of the day. We get an estimate for the substitution matrices for each hour of the day and compute the average. The results are shown in Tables 5.8 and 5.9.

↗	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**Total**
P1	-	0.11	0.15	0.14	0.14	0.15	0.69
P2	0.22	-	0.08	0.11	0.12	0.12	0.65
P3	0.24	0.07	-	0.07	0.08	0.07	0.53
P4	0.26	0.09	0.07	-	0.12	0.12	0.66
P5	0.19	0.10	0.10	0.12	-	0.14	0.65
P6	0.17	0.13	0.11	0.11	0.13	-	0.65

Table 5.8: Substitution rates of buns. Read as substitution from row to column.

We detect that 43 % to 75 % of customers are willing to substitute to another product if their first choice is not available. Earlier empirical work on substitution within bakery products found rates of 75% to 82% (van Woensel et al., 2007). However, the product portfolio in the study of van Woensel et al. (2007) was much larger (208 products) and therefore it is also more likely that a substitution to a more similar product takes place and substitution rates are higher. We note that in the category buns, the substitution rates *to* product 1 are relatively high. The same is true for product 4 in category breads. These products stand out as they have the lowest price within the respective category.

Finally, we *decensor* and *deflate* historical sales data. A main drawback of point-of-sales data (e.g. our dataset) is that historical demand is unobservable. Our sales data is distorted as there have been frequent stock-outs in the past. There are two possible distorting effects in the case of a stock-out. First, the sales data of the out-of-stock product is *censored* if customers cannot find their preferred product and choose to substitute to another product or

↗	P1	P2	P3	P4	P5	P6	Total
P1	-	0.14	0.09	0.25	0.12	0.12	0.72
P2	0.19	-	0.09	0.19	0.14	0.14	0.75
P3	0.07	0.09	-	0.07	0.10	0.10	0.43
P4	0.17	0.16	0.09	-	0.20	0.10	0.72
P5	0.18	0.12	0.09	0.24	-	0.12	0.75
P6	0.16	0.13	0.08	0.22	0.15	-	0.74

Table 5.9: Substitution rates of breads. Read as substitution from row to column.

leave the store without buying anything at all. Second, the sales data of the substitute products is *inflated* by the demand for the out-of-stock products.

The *decensoring* of sales in Step 4 in the following algorithm is based on Lau and Lau (1996). In particular, we calculate the average hourly share of demand for product $\tilde{P}$ in relation to the total demand on days on which the product was not sold out. Based on this averaged intraday demand pattern, we extrapolate the sales data when a stock-out occurs in order to estimate uncensored demand. To *deflate* the demand of the products in L, we subtract the sales in each hour of stock-out of product $\tilde{P}$ that are due to substitution demand.

We apply the following procedure to each product category on each day in order to *decensor* and *deflate* the hourly sales data.

1. Define a set of all products of the category $L = \{P_1, ..., P_n\}$.
2. Find product $\tilde{P}$ in L that goes out of stock first. If no stock-out, end.
3. Delete $\tilde{P}$ from L.
4. *Decensor* demand of product $\tilde{P}$ based on Lau and Lau (1996).
5. *Deflate* demand of all products in L based on substitution rates.
6. Go to 2.

Table 5.10 shows the average proportion of daily demand of each product within each category. While this proportion is relatively homogeneous for breads (8.5% to 23.8%), product 1 dominates the buns with a share of 64.3%.

Category	P1	P2	P3	P4	P5	P6
Buns	0.643	0.107	0.070	0.068	0.057	0.055
Breads	0.220	0.139	0.205	0.238	0.113	0.085

Table 5.10: Average proportion of demand of each product within each category.

We split the dataset into a training set containing 110 weeks and a test set containing the remaining 31 weeks and perform a rolling 1-step-ahead prediction evaluation on the test set in order to assess the performance of the methods. We fit the models and determine the error distributions every 14 days on a rolling training dataset with constant size.

5.4.2.2 Methods

In order to analyze the impact of the modeling decision in each step and to compare our IEO approach to the more traditional process of separately estimating demand and optimizing inventories, we introduce two different estimation methods and use the estimates as input for the optimization. In the optimization step, we differentiate four different modeling approaches that we describe below.

Separate Estimation and Optimization (SEO)
We briefly introduce the considered estimation and optimization methods.

Estimation. As a benchmark estimation method, we employ exponential smoothing that is widely used in practice and also implemented in most enterprise resource planning software. Exponential smoothing methods calculate the forecast by computing a weighted average of past observations. The weights decay as the observations get older. Hyndman et al. (2002, 2008) propose innovation space models that generalize exponential smoothing methods (*ETS*). For our use case, a model with additive seasonality, no trend, and additive errors (i.e. ETS(ANA)) is suitable (see Section 2.1).

As the second estimation method, we use feed-forward Artificial Neural Networks (ANNs) (see Section 2.2.2). We train and employ an ensemble of 50 *ANNs* with the *median* ensemble operator, as this approach is robust to the initial weights and provides reliable results (Barrow et al., 2010; Kourentzes et al., 2014). The input consists of lagged demand information (lags: 1, 2, ..., 7, 14) of each product, which is linearly scaled between 0 and 0.75, and further explanatory features (see Table 3.1). The output consists of the demand for all products.

For each forecasting method introduced in the previous section, we compute a set of common accuracy measures, including Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Relative Absolute Error (RAE) (see Section 2.3). Table 5.11 shows the average forecast accuracy over all time series by method.

Category	Method	RMSE	MAE	MAPE	SMAPE	MASE
Buns	ETS	75.15	28.50	20.39	18.91	0.84
	ANN	**44.13**	**22.00**	**16.75**	**15.59**	**0.67**
Breads	ETS	4.77	3.33	36.47	28.04	0.83
	ANN	**4.70**	**3.25**	**35.34**	**27.61**	**0.82**

Table 5.11: Forecast performance of the point predictions.

We observe that the ANN forecast outperforms the ETS forecast in all accuracy measures across both categories. The difference is relatively small for the category breads, while it is relatively large for the category buns. Though the ANN forecast is more accurate, it is not immediately clear what the impact on the overall performance in terms of profit is. Hence, we will analyze the effect of demand estimation on the average profit.

Optimization. The second step of SEO is the actual optimization of the inventory decision. There are two main modeling choices that must be made at the optimization level. First, should we consider the uncertainty associated with the forecast (deterministic vs. stochastic optimization)? Second, should we consider substitution in our decision (single product vs. multi-product)? Based on these choices, we differentiate four different optimization approaches:

- *DET*: We use deterministic optimization and do not consider substitution, i.e., the order quantity is equal to the forecast. With this approach, we can evaluate the costs of ignoring demand uncertainty *and* substitution.
- *STO*: We use stochastic optimization and do not consider substitution, i.e., we separately apply the single-product newsvendor to each product based on the individual demand distribution forecast. With this approach, we can evaluate the costs of ignoring substitution.
- *DET+SUB*: We use deterministic optimization and consider substitution, i.e., we apply the MILP introduced in Section 5.4.1 to each product category and use only the point forecasts as input. With this approach, we can evaluate the costs of ignoring demand uncertainty.
- *STO+SUB*: We use stochastic optimization and consider substitution, i.e., we apply the MILP introduced in Section 5.4.1 to each product category and use the complete demand distribution forecast as input.

We solve the optimization problem to near optimality (optimality gap $\leq 0.01\%$) with Gurobi 8.1. The average runtime for the optimization step is very short for the deterministic case with substitution (*DET+SUB*) with 0.15 seconds for one day. For the stochastic case with substitution (*STO+SUB*), the average runtime is considerably higher with 50.5 seconds for one day considering only a demand distribution including the last 100 data points.

Integrated Estimation and Optimization (IEO)

For the IEO approach, we integrate the ANN and the optimization problem. The resulting problem can be represented as

$$\max_{\mathbf{W}} \frac{1}{n} \sum_{t} \sum_{i} \left(u_i f_{i,t}(\mathbf{W}, \mathbf{x}_t) - (u_i + o_i) \left(f_{i,t}(\mathbf{W}, \mathbf{x}_t) - D^s_{i,t} \right)^+ \right), \tag{5.23}$$

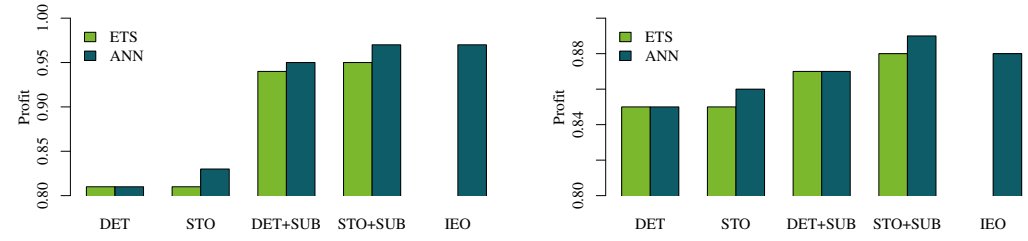
where $D^s_{i,t} = D_{i,t} + \sum_{j \neq i} \alpha_{ji} (D_{j,t} - f_{j,t}(\mathbf{W}, \mathbf{x}_t))^+$. The ith output of the ANN in period t, $f_{i,t}$ is now a prediction of the order quantity, as opposed to a prediction of demand. Put differently, we replace the loss function of the loss minimization problem by the objective function (5.23). We solve the optimization problem (5.22) with the same stochastic gradient decent algorithm as in the SEO approach. Note, that the relatively computationally expensive inventory optimization step can be omitted in the IEO approach.

5.4.2.3 Results

We apply all combinations of estimation and optimization methods as well as the *ANN IEO* approach to our dataset and present the performance in Table 5.12. Our performance measures are average profit, average product fill rate, and average category fill rate. We report the profit relative to the ex-post optimal profit (= 1.00), which is achieved by a perfect point forecast (*Perfect*) and the MILP optimization that considers substitution. Figure 5.6 illustrates the achieved profits graphically. In the following, we will disentangle the effects of considering uncertainty and substitution on the performance measure in our dataset.

	Method		Profit	Fill Rate						
	Estimation	**Optimization**		**Prod. 1**	**Prod. 2**	**Prod. 3**	**Prod. 4**	**Prod. 5**	**Prod. 6**	**Cat.**
Category Buns	Perfect	DET	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		DET+SUB	1.00	0.00	1.00	1.00	1.00	1.00	1.00	0.80
	ETS	DET	0.81	0.96	0.94	0.94	0.93	0.93	0.93	0.95
		STO	0.81	0.97	0.98	0.98	0.98	0.98	0.98	0.98
		DET+SUB	0.94	0.00	0.99	0.99	0.99	1.00	1.00	0.76
		STO+SUB	0.95	0.00	0.99	1.00	1.00	1.00	1.00	0.79
	ANN	DET	0.81	0.97	0.95	0.92	0.93	0.93	0.94	0.96
		STO	0.83	0.99	0.99	0.99	0.99	0.99	0.99	0.99
		DET+SUB	0.95	0.00	1.00	1.00	1.00	1.00	1.00	0.77
		STO+SUB	0.97	0.00	1.00	1.00	1.00	1.00	1.00	0.79
	ANN IEO		**0.97**	0.00	1.00	1.00	1.00	1.00	1.00	0.80
Category Breads	Perfect	DET	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		SET+SUB	1.00	0.98	1.00	1.00	0.02	1.00	1.00	0.93
	ETS	DET	0.85	0.93	0.90	0.93	0.93	0.91	0.90	0.93
		STO	0.85	0.95	0.95	0.97	0.95	0.96	0.96	0.96
		DET+SUB	0.87	0.97	0.96	0.96	0.00	0.98	0.96	0.86
		STO+SUB	0.88	0.97	0.97	0.98	0.00	0.99	0.98	0.88
	ANN	DET	0.85	0.92	0.86	0.95	0.93	0.91	0.90	0.92
		STO	0.86	0.95	0.93	0.98	0.95	0.97	0.96	0.97
		DET+SUB	0.87	0.97	0.94	0.97	0.00	0.98	0.96	0.85
		STO+SUB	**0.89**	0.97	0.96	0.99	0.00	0.99	0.98	0.89
	ANN IEO		0.88	0.97	0.95	0.99	0.00	0.99	0.98	0.88

Table 5.12: Average profit relative to ex-post maximum profit and resulting fill rates.



(a) Category buns. **(b)** Category breads.

Figure 5.6: Average profit relative to ex-post maximum profit.

The Effect of Demand Estimation

From the forecast accuracy measures in Table 5.11, we know that the *ANN* forecast was significantly more accurate than the *ETS* forecast in category buns. For breads, *ANN* was only slightly better. In order to evaluate the effect of estimation accuracy on the average

profit, we compare methods with the *ETS* forecast (light bars in Figure 5.6) to methods with the *ANN* forecast (dark bars in Figure 5.6). We observe that the differences in point forecast accuracy have no effect on average profits in the deterministic optimization case without substitution. The average profit can be increased by 1 percentage point in category buns if the *ANN* forecast is used instead of the *ETS* forecast and substitution is considered. This observation can be explained by the fact that the *ETS* forecast is already relatively accurate. This result is also supported by a previous study that found a similar influence of forecasting on performance in a single product newsvendor problem (Huber et al., 2019).

However, using the *ANN* as forecast method in the stochastic optimization approaches, yields better results compared to the *ETS* forecast, 2 percentage points in the category buns and 1 percentage point in the category breads. Thus, we conclude that the empirical distribution estimate of the *ANN* is more accurate, which results in higher profits.

The Effect of Uncertainty

In order to analyze the effects of uncertainty on average profits, we compare deterministic to stochastic optimization. In the case when substitution is ignored, i.e., when we compare *DET* and *STO*, the consideration of uncertainty improves the profit by 0 to 2 percentage points. The empirical distribution forecast of the *ANN* is more accurate, and therefore the value of stochastic optimization is larger if the *ANN* instead of *ETS* is employed. Similar results hold for the case when substitution is included, i.e., *DET+SUB* and *STO+SUB*.

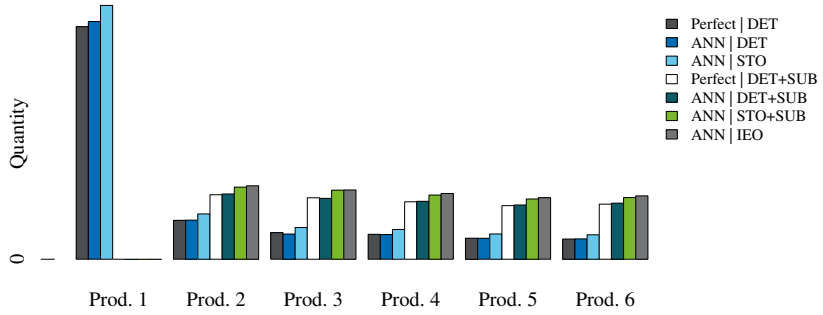
Furthermore, we study the impact of considering conditional uncertainty on overall profit in the inventory decision by looking at the results of our IEO approach. In both categories, we see no large difference in the performance of *ANN STO+SUB* and *ANN IEO*. The theoretical advantage of IEO of capturing conditional uncertainties seems to be not important in our case.

Overall, the effect of uncertainty is relatively small in our case study. This is also due to the fact that we use advanced estimation methods that do not differ widely in estimation accuracy. A previous study (Huber et al., 2019) showed that *ETS* and *ANN* outperform most forecasting approaches in a single-product newsvendor setting.

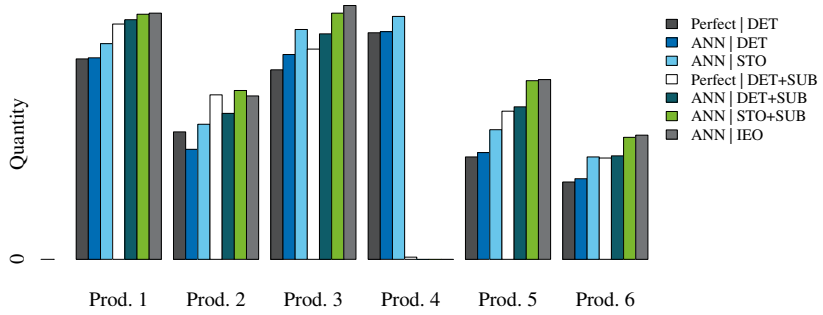
The Effect of Substitution

Next, we investigate the effect of substitution on the average profits and resulting fill rates. We can quantify the theoretical maximum profit increase of considering substitution by comparing *Perfect DET*, which is the optimal decision assuming no substitution, and *Perfect DET+SUB*, which is the optimal decision with substitution. We observe that the potential gains are high (14 percentage points) for category buns and much lower (4 percentage points) for category breads. The large gains in the category buns are mainly due to the specific characteristics of product 1. Product 1 has by far the largest share of demand within the category (64.3%), has the lowest profit margin, and the highest rates of substitution to other products. Due to these properties, it is optimal to not order product 1 at all (fill rate $= 0$) as enough customers substitute to higher margin products. Figure 5.7 illustrates these shifts in sales volumes. We see a similar but smaller effect with product 4 in category breads. It is also

the lowest margin product within the category, but it is not as dominant in terms of demand share (23.8%). The order of magnitude of these effects is similar when we compare *STO* to *STO+SUB* in both categories.



(a) Category buns.



(b) Category breads.

Figure 5.7: Average quantities of each product. The scale of the y-axis is not provided for reasons of confidentiality.

Overall, we find that accounting for substitution is the most important aspect that drives the performance of the methods under consideration. Secondly, accurate forecasting can further increase profits. Accounting for uncertainty is not as important as in the single-product newsvendor problem. Finally, we find that the estimation of conditional uncertainties as in the *ANN IEO* approach does not significantly improve the profit.

The Effect of Fill Rate Constraints

In the previous section, we observed large demand shifts from low-margin products to substitutes due to very low ordering decisions for these products that even resulted in an abandonment of these products from the assortment. We see similar effects in other applications of multi-product inventory models to real-world problems. Kök and Fisher (2007) find that "[p]roducts with low profit are dropped from the assortment, the number of facings of products with low marginal return are reduced, and the number of facings of those with higher returns are increased". However, these decisions optimize the short-term profit. Long-term effects of stock-outs (e.g. dissatisfied customers, future lost sales) are not captured in our models although they might be important for the customer's store choice (Briesch et al., 2009). Including these long-term effects in the underage costs u_i of each product i is difficult

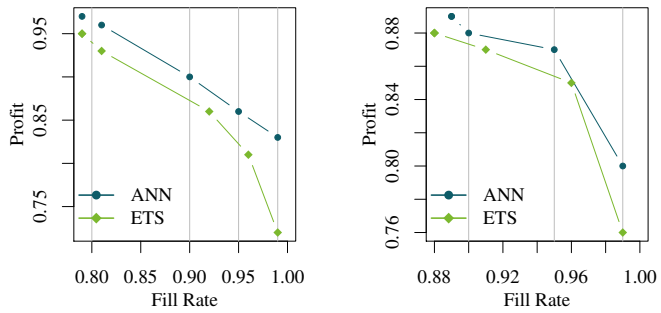
as they are hard to estimate. Therefore, we add target fill rate constraints to the MILP (5.15) to (5.21) that account for strategic service level requirements. We set a target fill rate per category as

$$\beta^{cat} \leq \frac{\frac{1}{n}\sum_t \sum_i (q_i - y_{i,t})}{\frac{1}{n}\sum_t \sum_i D_{i,t}}. \tag{5.24}$$

Additionally, we introduce target fill rate constraints per product as

$$\beta_i^{prod} \leq 1 - \frac{\frac{1}{n}\sum_t x_{i,t}}{\frac{1}{n}\sum_t D_{i,t}} \quad \forall i. \tag{5.25}$$

We set the category fill rate to values between 0.80 and 0.99 and the product fill rate to values between 0.80 and 0.95 and optimize the MILP based on the *ETS* and *ANN* distribution forecast. The resulting profits and fill rates are shown in Table 5.13.



(a) Category buns. **(b)** Category breads.

Figure 5.8: Average profit relative to ex-post maximum profit of *ETS STO+SUB* and *ANN STO+SUB* with fill rate constraints at category level.

Across all methods and categories, we can observe a trade-off between short-term profit and service level (fill rate), which is illustrated in Figure 5.8. As soon as one of the fill rate constraints is effective, the profit decreases. In category buns, a product fill rate of 0.80 already drastically reduces the profit as the substitution from product 1 to more profitable products is suppressed. In category breads, this effect is smaller as the products are more homogeneous with respect to volume and margin. Low category fill rate constraints do not harm the profit too much as they still allow for substitution.

5.5 Conclusion

We proposed a framework for how data can be leveraged in inventory problems on three different levels: demand estimation, optimization, and integrated estimation and optimization. To this end, we introduced data-driven solution approaches that go from data to decision in a single optimization problem for the single-product newsvendor problem and the multi-product newsvendor problem with substitution. We are specifically interested in the effects of newly available large datasets on the overall performance of diverse solution approaches.

	Method		Target Fill Rate		Profit	Fill Rate						
	Estimation	Optimization	per Cat.	per Prod.		Prod. 1	Prod. 2	Prod. 3	Prod. 4	Prod. 5	Prod. 6	Cat.
Category Buns	ANN	STO+SUB	-	-	0.97	0.00	1.00	1.00	1.00	1.00	1.00	0.79
			-	0.80	0.86	0.82	1.00	1.00	1.00	1.00	1.00	0.96
			-	0.90	0.84	0.92	1.00	1.00	1.00	1.00	1.00	0.98
			-	0.95	0.84	0.96	0.99	1.00	0.99	1.00	0.99	0.99
			0.80	-	0.96	0.05	1.00	1.00	1.00	1.00	1.00	0.81
			0.90	-	0.90	0.54	1.00	1.00	1.00	1.00	1.00	0.90
			0.95	-	0.86	0.80	1.00	1.00	1.00	1.00	1.00	0.95
			0.99	-	0.83	0.98	1.00	1.00	1.00	1.00	1.00	0.99
	ETS	STO+SUB	-	-	0.95	0.00	0.99	1.00	1.00	1.00	1.00	0.79
			-	0.80	0.84	0.83	0.99	0.99	0.99	0.99	0.99	0.95
			-	0.90	0.82	0.93	0.99	0.99	0.99	0.99	0.99	0.97
			-	0.95	0.80	0.96	0.99	0.99	0.99	0.99	0.99	0.98
			0.70	-	0.95	0.00	0.99	1.00	1.00	1.00	1.00	0.79
			0.80	-	0.93	0.13	1.00	1.00	1.00	1.00	1.00	0.81
			0.90	-	0.86	0.64	0.99	0.99	0.99	1.00	0.99	0.92
			0.95	-	0.81	0.87	0.99	0.99	0.99	0.99	0.99	0.96
			0.99	-	0.72	0.98	0.99	0.99	0.99	0.99	0.99	0.99
Category Breads	ANN	STO+SUB	-	-	0.89	0.97	0.96	0.99	0.00	0.99	0.98	0.89
			-	0.80	0.87	0.91	0.93	0.98	0.85	0.98	0.96	0.94
			-	0.90	0.86	0.93	0.93	0.97	0.93	0.97	0.97	0.96
			-	0.95	0.82	0.97	0.95	0.97	0.97	0.98	0.98	0.98
			0.80	-	0.89	0.97	0.96	0.99	0.00	0.99	0.98	0.89
			0.90	-	0.88	0.98	0.96	0.99	0.12	0.99	0.98	0.90
			0.95	-	0.87	0.93	0.94	0.98	0.83	0.98	0.96	0.95
			0.99	-	0.80	0.98	0.96	0.99	0.99	0.98	0.98	0.99
	ETS	STO+SUB	-	-	0.88	0.97	0.97	0.98	0.00	0.99	0.98	0.88
			-	0.80	0.86	0.90	0.95	0.96	0.84	0.98	0.97	0.94
			-	0.90	0.85	0.93	0.95	0.96	0.93	0.97	0.97	0.96
			-	0.95	0.81	0.97	0.96	0.97	0.96	0.98	0.98	0.98
			0.70	-	0.88	0.97	0.97	0.98	0.00	0.99	0.98	0.88
			0.80	-	0.88	0.97	0.97	0.98	0.00	0.99	0.98	0.88
			0.90	-	0.87	0.97	0.97	0.98	0.23	0.99	0.98	0.91
			0.95	-	0.85	0.94	0.95	0.97	0.87	0.98	0.97	0.96
			0.99	-	0.76	0.99	0.97	0.99	0.99	0.99	0.98	0.99

Table 5.13: Average profit relative to ex-post maximum profit and resulting fill rates under fill rate constraints.

For the single-product newsvendor problem, we highlight that integrated estimation and optimization in the newsvendor problem is equivalent to the Quantile Regression problem, and we introduce novel data-driven methods for the newsvendor problem based on Machine Learning and Quantile Regression. Moreover, we empirically compare the methods to well-established standard approaches on a real-world dataset. Moreover, we analyze the impact of data-driven approaches on the three levels on the overall performance.

The key result of our evaluation is that data-driven approaches outperform their model-based counterparts in most cases. In our evaluation, this finding already holds for a demand history of beyond 25 weeks (i.e. 150 data points). However, overall performance depends heavily on the demand estimation method employed. We found that poor forecasts cannot be compensated for by the choice of the subsequent optimization approach. Thus, the selection of the forecast model is the most crucial decision in the case of separated estimation and optimization.

The empirical evaluation of the Quantile Regression approaches revealed that integrating forecasting and optimization is beneficial only if enough data are available to estimate the conditional quantiles and limited to target service levels smaller than 0.8. When working with single time series, separate estimation and optimization yields superior results. This finding is in line with the empirical analysis of Ban and Rudin (2018).

More sophisticated estimation methods such as ANNs and Gradient Boosted Decision Trees require more training data in order to produce reliable results. However, these methods are also the only methods that constantly improve as more data becomes available. In our example, the demand history should contain more than six months of training data before employing Machine Learning. If a limited amount of data is available, simple methods such as the seasonal moving average can be suitable alternatives.

For the multi-product newsvendor problem with substitution, we disentangle and evaluate the effects of the estimation approach, considering uncertainty and substitution. Our key result is that the integrated approach performs competitive to state-of-the-art methods with less computational effort. Especially when the uncertainty associated with the forecast is feature-dependent its ability to estimate conditional decisions is beneficial. We also find that the most important aspect in our case is the consideration of substitution effects. This is due to the heterogeneity of the products with respect to profit margins and volumes. Considering uncertainty is less important than in the single-product newsvendor problem as the ability to substitute to another product pools some risk. Overall, combining modern ML methods and OR approaches can increase profits significantly.

The major advantage of ML methods is that they are very flexible with respect to the input and that they are naturally able to process large datasets. The ability of ML methods to leverage similarities of time series across products and stores significantly improved their performance in our case. Additionally, they do not require restrictive assumptions on the demand process. Hence, they can identify patterns that traditional time series methods cannot detect. For instance, they can model multiple seasonalities (e.g. week and year), special days

(e.g. public holidays), promotional activities and outliers (Barrow and Kourentzes, 2018). A drawback of these approaches is that they are a black box, which makes it more difficult to justify the resulting predictions. However, when improvements in forecast accuracy can be easily measured, as in the case of baked goods, the advantage of accurate predictions should outweigh the issue of interpretability.

6

Intraday Decision Support

In this chapter, we consider the last phase of the bakery supply chain, which is related to the operational decisions in the stores. Section 6.1 is based on a paper titled *"Intraday Shelf Replenishment Decision Support for perishable Goods"* by Jakob Huber and Heiner Stuckenschmidt.

6.1 Baking Plan Generation

6.1.1 Introduction

We consider the case of a bakery supply chain where the stores are daily delivered (see Section 1.2). Some products are not ready for sale when they arrive at the store and need to be processed during the day, i.e., baked and placed on the shelves. For this purpose, each store is equipped with up to three ovens. Baking goods during the day is necessary as the items have a high rate of deterioration and should be provided as fresh as possible in order to increase the customer satisfaction. Among the determination of the daily order quantity, a challenge is to provide a suitable baking plan that can be executed by the store personnel. A baking plan is a schedule that outlines when the different products have to be baked and consequently placed on the shelves (see Table 6.1). The baking plan shows which oven has to be used and which baking program has to be started. The amount of items per article that has to be baked is given in the number of baking trays. The number of items per tray is fixed for every article. The capacity of an oven corresponds to the number of baking trays that can be processed at the same time.

Time	Oven [ID]	Program [ID]	Article	Baking Trays [Qty.]
05:30	1	1	11 12	6 2
05:30	2	3	31 32 33	1 2 1
05:55	1	2	21	8
...	...	...	...	...

Table 6.1: An example of a baking plan (schedule). The plan outlines the number of baking trays that have to be placed in a specific oven and baked at given point in time. Therefore, it is necessary to start the baking program that is suitable for the items on the baking trays.

The objective of this study is to provide a solution approach for the computation of a baking plan. The expected demand should be met and the freshness of the sold items should be increased under the given constraints if the baking plan is executed accordingly by the store staff. Moreover, a further objective can be the reduction of additional operational costs. For instance, the ovens should be fully loaded in order to save energy costs and the number of starts of baking processes should be limited as this requires the staff to interrupt other tasks.

In order to be able to compute a baking schedule, we need an intraday (e.g. hourly) demand estimation that has to match the daily delivery quantity. Based on the forecasts, we compute a schedule representing the baking plan. In particular, we address the following research questions:

- Is a Machine Learning (ML) method suitable for hourly demand forecasting considering the given application scenario?
- How can the baking plan generation be formulated as a scheduling problem (Pinedo, 2016)?
- What is the effect of the forecast accuracy on the operational performance?

The remainder of this study is structured as follows: In Section 6.1.2, we outline related work concerning forecasting and in-store logistics. Our solution approach for intraday decision support for baked goods is introduced in Section 6.1.3. We present and discuss the results of the empirical evaluation in Section 6.1.4. We conclude this study with a brief summary of the most important results in Section 6.1.5.

6.1.2 Related Work

A general literature review on time series forecasting and ML is provided in Section 4.2. While we are not aware of literature that discusses intraday forecasting for perishable goods in the retail industry, there are other application areas that are concerned with intraday forecasting. A non-exhaustive list of application areas includes energy load forecasting (Hong and Fan, 2016; Marino et al., 2016), load forecasting of cooling systems (Li et al., 2015), water usage forecasting (Quevedo et al., 2014), forecasting call arrivals in call centers (Barrow and Kourentzes, 2018; Ibrahim et al., 2016), or short-term traffic forecasting (Lv et al., 2015; Ma et al., 2015). In the aforementioned studies, ML methods are frequently applied and are at least a viable contender for traditional time series models. Moreover, models based on long short-term memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997) are frequently used for intraday forecasting in recent studies (e.g. Ma et al. (2015); Ke et al. (2017b); Tian et al. (2018); Qing and Niu (2018)). Hence, we will rely on LSTM models as representatives of ML methods in our empirical evaluation in this study.

Hierarchical forecasting can be applied to connect the daily level and the hourly level. While most studies are concerned with leveraging the organizational structure, Athanasopoulos et al. (2017) and Kourentzes et al. (2017) discuss the challenges related to temporal hierarchies. They emphasize that the signal-to-noise ratio can be strengthened and that the effect of

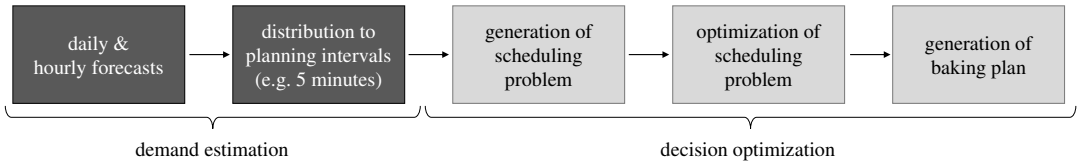


Figure 6.1: Overview on the different phases of our approach.

outliers at lower levels can be reduced. In particular, temporal aggregation allows to reduce intermittency in the time series data (Nikolopoulos et al., 2011; Petropoulos et al., 2016).

A review on inventory systems for deteriorating goods is provided by Bakker et al. (2012). Hübner and Kuhn (2012) elaborate on in-store logistics planning in the context of shelf space management. van Donselaar et al. (2006) emphasize the importance of developing automated ordering system for perishable goods as their characteristics differ from other product categories. Recent inventory models acknowledge the costs for in-store logistics (van Zelst et al., 2009; Curşeu et al., 2009; van Donselaar et al., 2010; Taube and Minner, 2018; Mou et al., 2018). Hofer et al. (2016) and Teller et al. (2018) report that low on-shelf availability is caused by poor forecasting, inefficient backroom operations, and replenishment policies. Reiner et al. (2013) and Teller et al. (2018) also claim that measures taken at store level are highly effective and have an immediate impact. However, specific decision support systems for in-store operations are not proposed. With this study, we want to address this gap in the context of baked goods and extend the literature in various ways:

- We propose an intraday decision support system for baked goods.
- We perform hourly forecasting in the retail domain.
- We conduct an empirical evaluation of our solution approach.
- We evaluate the influence of the prediction model on the operational performance.

6.1.3 Methodology

We present a solution approach for intraday baking that consists of two distinct phases: forecasting and scheduling (see Figure 6.1). The forecasting phase (see Section 6.1.3.1) is concerned with providing intraday demand estimations that serve as input for the subsequent scheduling phase (see Section 6.1.3.2). The initial forecasts will be transformed to jobs associated with deadlines and costs for earliness and tardiness. The jobs are assigned to machines (i.e. ovens) according to certain requirements. The resulting schedule represents the baking plan that can be executed in the stores.

6.1.3.1 Forecasting

An essential input for the baking plan generation is the intraday demand estimation. The considered application scenario requires to completely bake the daily delivered order quantity

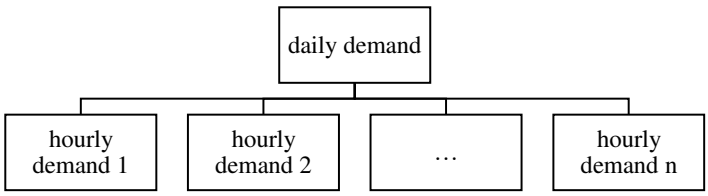


Figure 6.2: The temporal hierarchy for intraday baking. The daily order quantity has to match the sum of the demand forecasts at the hourly level.

on the same day. The goods are ordered on the previous day, which means that we also need to compute the daily demand of each product for the next day. As the sum of the hourly predictions has to match the forecast at the daily level, we exploit the temporal hierarchy (see Figure 6.2).

In order to connect both temporal levels, we consider a bottom-up approach and a top-down approach. The bottom-up approach requires the direct computation of the demand forecasts at the hourly level. This approach has the advantage that the hourly predictions are directly given and the daily demand can also be easily obtained. However, the data on the hourly level is rather noisy and some products are not sold or demanded every hour. Consequently, the accuracy of the resulting predictions at the daily level can be negatively affected. Another option is to compute the daily demand directly, which has subsequently to be distributed to the different hours. The top-down distribution is closely connected to the actual operational process. However, in order to obtain the hourly forecasts, we additionally need to forecasts an intraday demand profile. Intraday profiles reflect the percentage of the daily demand that is sold in each hour and can be used to distribute the daily quantity top-down. The advantage of this approach is that the data on the daily level is less noisy than data on the hourly level. Moreover, the intraday demand profiles are more robust to changes in the demand level. Hence, we evaluate forecasts for three different target levels: (1) daily demand, (2) hourly intraday profiles, and (3) hourly demand (see Figure 6.3). However, for productive usage of the system, we need either daily forecasts and hourly profile forecasts, or just hourly forecasts. The only requirement is that the sum of the hourly forecasts matches the daily order quantity.

We employ a long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997), which is a recurrent neural network that processes the input features in sequential order by applying the same network to each step in a sequence. The input features of the LSTM are lagged time series observations (lags: 14, 7, 6, …, 1) which cover the complete last week and the observation on the same weekday two weeks ago as the demand for baked goods is subject to a strong weekly seasonality. For daily forecasts, the network only forecasts the demand for the next day. At hourly level (demand & profile), the network predicts all hourly values at once, i.e., the output dimension reflects the maximum number of opening hours. Consequently, the lagged observations are also included for each hour. Additionally, we enhance each step of the sequence with explanatory feature data (see Table 3.1). The feature

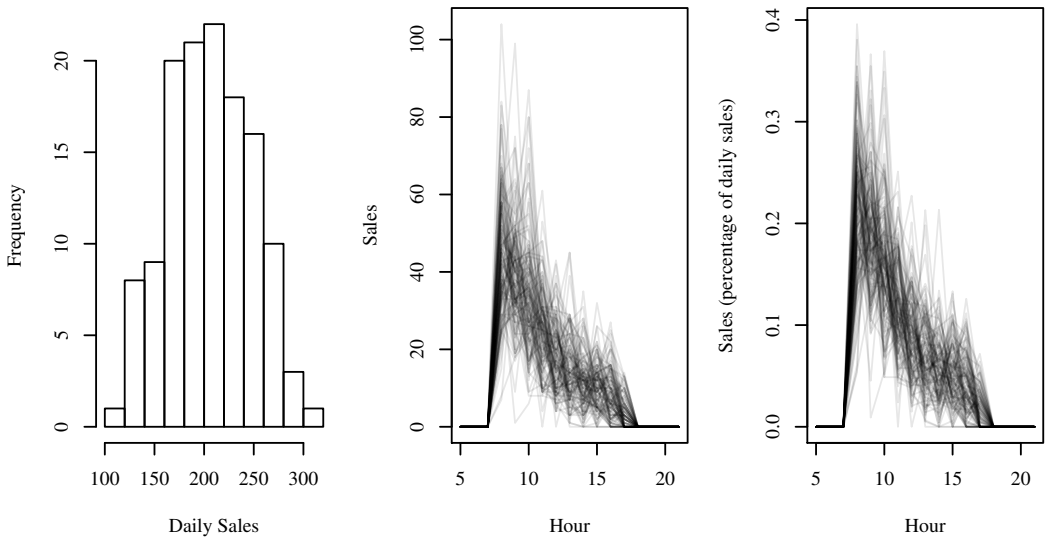


Figure 6.3: The charts depict the demand distribution of a single product in a store on Sunday at different levels: daily sales distribution (left), hourly sales (center), and intraday profiles (right).

data is not only derived from the enterprise resource planning system of the company but also consists of external data like calendric information and weather. Moreover, some feature data are not specific per time series like the characterization of the location. Such feature data is useful as we train global models that are able to forecast any time series of the dataset. Hence, the model is able to learn patterns across products and stores. An additional advantage of this approach is that we have more training data and need to train less models.

Even though the planning granularity of the baking plan is lower than one hour, we only compute hourly forecasts as the data is already quite noisy at the hourly level. Moreover, we only have access to aggregated hourly point-of-sales data. In order to obtain a demand estimation for shorter intervals, we linearly distribute the hourly forecasts.

6.1.3.2 Scheduling

The output of the forecasting phase is the expected demand at the hourly level. In the next phase, those forecasts are used to compute the actual recommendation for action with respect to intraday baking. The store personnel need to know when the different items have to be baked and consequently placed on the shelves. Hence, our goal is to provide a baking plan, i.e., a schedule that supports the decisions. A baking plan considers all relevant articles and needs to be provided per store and day. We formulate a mixed-integer program for this problem that is closely connected to the actual process in the store. The main task for the personnel is to place the baking trays into the ovens. Hence, we model this problem such that placing a baking tray into an oven is a job that needs to be scheduled.

The jobs (j) can be derived from the forecasts by distributing the expected demand to baking trays. Only items from one product can be placed on a baking tray. The number of items that can be placed on a baking tray is fixed per product. Additionally, each product is

assigned to a baking program that specifies the duration of a job (r_j). Products belonging to the same program can be processed in parallel and build job families (i). The deadline d_j of a job is the time at which the first item on a baking tray is expected to be sold minus the duration of the baking program. It is desired to bake the products as close to the deadline as possible because late jobs cause stock-outs while the freshness of the goods is reduced if they are baked earlier than necessary. Thus, we introduce costs for earliness e_j and tardiness t_j for each a job. The costs w_j^e and w_j^t reflect the expected average revenue per time instant of the job. We set symmetric penalties but our approach is also applicable if this is not the case. In summary, the jobs are associated with the following information: `job id`, `article id`, `quantity`, `family id`, `deadline`, `duration`, `earliness costs`, `tardiness costs`.

The jobs need to be assigned to the ovens, which are the machines of the scheduling problem. The ovens are only characterized by their capacity, which is the number of baking trays that can be processed at the same time: `machine id`, `capacity`.

The stores typically operate more than one oven, which means that we have a parallel machine environment. The ovens should be loaded to their full capacity. A job has to be processed by exactly one machine, and a started baking process cannot be interrupted. After completion of a baking process, the items on the baking trays are placed on the shelves and can be purchased by the customers.
We formulate the following integer linear program to solve the scheduling problem:

Sets, Indices, Parameters:

- time $t \in [1,\ T]$; T also reflects the planning horizon
- machine $k \in [1,\ K]$
  - b_k: capacity of machine k
- job $j \in [1,\ J]$
  - d_j: deadline of job j
  - r_j: duration of job j
  - f_j: job family of job j
  - w_j^e, w_j^t: penalties for earliness, tardiness of job j
- job family $i \in [1,\ I]$
  - r_i: duration of jobs in job family i

Variables:

- s_j: start time, e_j: earliness, t_j: tardiness of job j
- x_{jt}: job j is started at time t
- x_{jkt}: job j is processed by machine k at time t
- y_{jk}: job j is processed by machine k
- m_{ikt}: program i is started on machine k at time t
- m_{kt}: machine k is active at time t

The variables s_j, e_j, and t_j are integer variables in the range from 1 to T for s_j, and 0 to $T-1$ for e_j and t_j. The other variables (x_{jt}, x_{jkt}, y_{jk}, m_{ikt}, m_{kt}) are all binary ($[0,\ 1]$) and only one if the respective event is true. The variables x_{jt} and m_{ikt} are only defined from 1 to T_j^{max} (T_i^{max}). T_j^{max} is the latest starting time of a job ($T_j^{max} = T - r_j + 1$) as a started job has to be finished during the planning horizon. The objective is to minimize the earliness-tardiness costs of the jobs:

$$\text{minimize} \sum_{j=1}^{J} (w_j^e e_j + w_j^t t_j)$$

The objective function is subject to a set of constraints (C1-C22). The first group of constraints comprises cardinality and count constraints:

$$\begin{array}{lllll}
\text{(C1)} & \sum_{t=1}^{T_j^{max}} x_{jt} & = 1 & \forall j \\
\text{(C2)} & \sum_{k=1}^{K} y_{jk} & \leq 1 & \forall j \\
\text{(C3)} & \sum_{k=1}^{K} x_{jkt} & \leq 1 & \forall j, t \\
\text{(C4)} & \sum_{k=1}^{K} \sum_{t=1}^{T} x_{jkt} & = r_j & \forall j, k \\
\text{(C5)} & \sum_{j=1}^{J} x_{jkt} & \leq b_k & \forall k, t
\end{array}$$

Constraint C1 ensures that each job is exactly started once. A job is processed by at most one machine in general (C2) and also per time instant (C3). The sum of the activities of a job has to match its processing duration (C4) and the capacity of a machine cannot be exceeded (C5). We define the following constraints in order to obtain the starting time s_j of a scheduled job (C7) and to determine if the job is early or late (C6):

$$\begin{array}{lllll}
\text{(C6)} & s_j - t_j + e_j & = d_j & \forall j \\
\text{(C7)} & \sum_{t=1}^{T_j^{max}} (t\, x_{jt}) - s_j & = 0 & \forall j, k, t
\end{array}$$

The constraint C7 connects the binary starting variables of the jobs x_{jt} to a numeric value s_j that can be used to calculate the earliness e_j and tardiness t_j (C6) which are used in the objective function of the linear program. The remaining constraints are all given as logical clauses for better readability but can be transformed to linear constraints. The next group of constraints models the machine activity:

$$\begin{array}{lll}
\text{(C8)} & m_{kt} \Rightarrow \neg(\bigvee_{i=1}^{I} m_{ikt}) & \forall k, t \\
\text{(C9)} & \bigvee_{i=1}^{I} m_{ikt} \Rightarrow \neg m_{kt} & \forall k, t \\
\text{(C10)} & m_{ikt} \Rightarrow \bigwedge_{s=1}^{r_i - 1} m_{k(t+s)} & \forall i, k, t \in \{1 \leq t \leq T_j^{max}\} \\
\text{(C11)} & m_{kt} \Rightarrow \bigvee_{i=1}^{I} \bigvee_{t'=t-r_i+1}^{t-1} m_{ikt'} & \forall k, t \\
\text{(C12)} & m_{ikt} \Rightarrow \neg \bigvee_{i'=1 | i \neq i'}^{I} m_{i'kt} & \forall i, k, t \\
\text{(C13)} & m_{ikt} \Rightarrow \neg m_{k(t+r_i)}) & \forall i, k, t \in \{1 \leq t \leq T_i^{max}\}
\end{array}$$

If a machine is not idle at a given point in time, it can either be active or a program is started (C8, C9). The machine has to be active during the duration of a program, i.e., a program

cannot be interrupted (C10), and a machine can only be active if a program has been started before (C11). Only one program can be started at a time on each machine (C12) and a machine cannot be active directly after a program ends (C13). The remaining constraints ensure that the jobs are assigned to machines and processed accordingly:

$$
\begin{array}{lll}
\text{(C14)} & x_{jkt} \Rightarrow y_{jk} & \forall j,k,t \\
\text{(C15)} & x_{jkt} \wedge m_{ikt} \Rightarrow x_{jt} & \forall j,k,t \\
\text{(C16)} & x_{jt} \Rightarrow \bigvee_{k=1}^{K} m_{ikt} & \forall j,t \\
\text{(C17)} & x_{jt} \Rightarrow \bigwedge_{t'=t}^{t+r_i-1} \bigvee_{k=1}^{K} x_{jkt'} & \forall j,t \\
\text{(C18)} & y_{jk} \Rightarrow \bigvee_{t=1}^{T_j^{max}} m_{ikt} & \forall j,k \\
\text{(C19)} & x_{jkt} \Rightarrow m_{ikt} \vee m_{kt} & \forall j,k,t \\
\text{(C20)} & x_{jkt} \wedge m_{ikt} \Rightarrow \bigwedge_{t'=t+1}^{t+r_j-1} m_{ikt'} & \forall j,k,t \\
\text{(C21)} & m_{kt} \Rightarrow \bigvee_{j=1}^{J} x_{jkt} & \forall k,t \\
\text{(C22)} & m_{ikt} \Rightarrow \bigvee_{j=1}^{J} x_{jkt} & \forall i,k,t
\end{array}
$$

The constraint (C14) ensures that each job is assigned to the machine it is processed on. If a job is active and associated with a machine that starts a program at the same time, the job has to be started at this time instant (C15). Constraint C16 states that a machine has to be started when a job is started. If a job is started, it has to be active during its duration, i.e., a running job cannot be interrupted (C17). A suitable program has to be started on a machine if a job is assigned to it (C18) and a machine has to be either started or active if a job is active (C19). A machine must be active during the subsequent steps after the start of a program (C20). Finally, a job has to be active if a machine is active (C21) or started (C22).

A linear program solver can solve this scheduling problem. There are different possibilities to obtain the baking plan (see Table 6.1) from the solved linear program as several variables provide information about the assignment of the jobs to the machines and the starting times of the jobs. For instance, x_{jt} or s_j in combination with y_{jk} can be used for this purpose. The resulting schedule is provided to the personnel in the stores who are responsible for filling the ovens and the shelves.

Reduction of the Problem Size

The problem size of the aforementioned scheduling program can be reduced in order to make solving it more feasible. It is a requirement to fully load the ovens as the store staff should not unnecessarily interrupt their other tasks, which are mostly related to serving the customers, and energy costs for running the ovens are also a factor. Jobs belonging to the same family can be processed concurrently by an oven. Hence, jobs of a job family can be grouped by the size of the smallest oven after they are ordered by their deadlines. As an oven is able to process at least four baking trays at a time and the capacity of the larger ovens are multiples of the smallest oven, the problem size with respect to the jobs that need to be scheduled can be reduced by roughly 75%. The deadline of a derived job is the earliest deadline, and the penalties for earliness and tardiness are the average penalties of the grouped

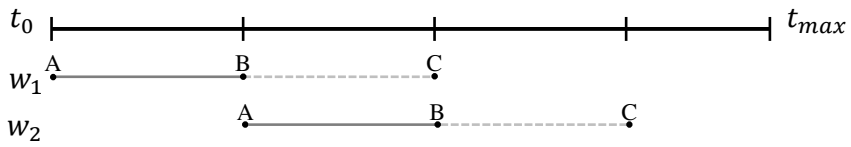


Figure 6.4: The figure illustrates the rolling scheduling approach. For each planning window, we consider jobs having deadlines between A (or earlier) and B. The selected jobs are then scheduled within the time window between A and C. After schedule optimization, jobs having starting times between A and B will be fixed. Scheduled jobs that are planned to start after point B will be postponed to the next window.

jobs. Additionally, the capacity of the ovens is also reduced accordingly, i.e., the capacity of the smallest oven is one, which also reduces the complexity of the problem.

Moreover, planning the full day at once is not necessary as decisions in the morning hardly influence decisions in the afternoon. This allows us to employ a rolling scheduling approach (Sridharan et al., 1987) to solve the scheduling problem as illustrated in Figure 6.4. We divide a day into windows that are optimized in sequence. As the windows overlap, it is possible that jobs that are scheduled in a previous window are still in process in the subsequent windows. In order to block the machines in the subsequent planning steps, we manually set the values of m_{ikt} and m_{kt} to zero. The support of rolling scheduling makes the problem formulation also directly applicable to an application scenario where intraday updates are possible (e.g. bake-off in supermarkets). At any point during the day, it is not only possible to update the schedule for the remaining part of the day but the jobs, which need to be scheduled, can also be changed.

6.1.4 Empirical Evaluation

The empirical evaluation aims to assess the performance of the introduced approach and falls into two parts: First, we focus on the forecast accuracy (see Section 6.1.4.2.1) of the intraday and daily forecasts. Consequently, we evaluate the impact of the forecast performance on the inventory or rather the operational performance in Section 6.1.4.2.2.

6.1.4.1 Experimental Design

We use dataset v3b for the evaluation of our solution approach for intraday baking (see Section 3.2). The dataset comprises hourly sales data of 14 different baked goods from 9 stores over a period of 987 days (i.e. 141 weeks or 2.7 years). As the assortment varies among the stores, we only need to consider 121 time series. Moreover, information about the available ovens in the different stores as well as the baking program assignment of the products is available. The first 110 weeks ($\approx 78\%$) of the data serve as training data while only the latter 22% (31 weeks) are used for the evaluation. We compute hourly forecasts and baking plans in order to demonstrate the viability of our approach. The computed baking plans consider all articles and empirical constraints with respect to the available ovens and the duration of the

baking processes. Hence, we meet all requirements of a productive application. We compute a baking plan for each store and date when the store is not closed, i.e., 1820 baking plans per forecasting method.

The method to optimize the scheduling problem introduced in Section 6.1.3.2 is identical for all forecasting methods and yields the optimal baking plans given hourly forecasts. However, we consider different options to compute the hourly forecasts that have to match the daily order quantity. We can either compute the hourly forecasts directly or distribute the daily forecasts to the hourly level (see Figure 6.1). Thus, we consider the following targets and different modes:

- **target: daily**
  - **direct:** We compute the daily demand directly, i.e., using daily data.
  - **bottom-up:** We obtain the daily forecast by accumulating the hourly forecasts.
- **target: hourly profile**
  - **direct:** We compute the hourly profiles directly, i.e., based on past hourly profiles.
  - **one:** We scale the sum of the hourly profile forecasts obtained from mode "direct" to one in order to be able to distribute the daily quantity.
- **target: hourly**
  - **direct:** We compute the hourly demand directly, i.e., using hourly data.
  - **top-down:** We distribute the daily forecasts using hourly profile predictions (mode: one).

We evaluate all possible approaches and only use the best approach for the computation of the baking plans. We suspect that a top-down approach is an alternative as the hourly data is quite noisy.

In order to evaluate the forecasts, we compute the mean absolute scaled error (MASE), the symmetric mean absolute percentage error (SMAPE), the root mean square error (RMSE), and the mean absolute error (MAE). Moreover, we compute the fill rate (FR), the loss rate (OR), and the service level (SL) in order to assess the supply chain performance (see Section 2.3). As either the fill rate or the overage rate can be manipulated, we sum the deviation from the optimum of both key figures and refer to it as total loss ($\mathrm{Loss} = (1 - \mathrm{FR}) + \mathrm{OR}$).

In addition to the ML method (LSTM) described in Section 6.1.3.1, we also consider baseline forecasting approaches and the time series model ETS(ANA) (Hyndman et al., 2008) (see Section 2.1). All baseline methods consider the weekly seasonality and simulate the typical decision process of humans that are likely to take the previous weeks into account if they have access to this information. We cannot consider the decision quality based on judgmental forecasts because this information is not available. However, it is unreasonable to assume that untrained store clerks, some of whom are part-time employees, are able to outperform statistical methods. Moreover, the store personnel are also responsible for other

tasks, and they cannot dedicate time to the optimization of baking plans. The forecast of S-Naïve is the last observation from the same part of the season (e.g. weekday) and S-Mean (S-Median) computes the mean (median) of the last four observations. For the intraday forecasts, each hour is represented by a separate time series, i.e., we only have to deal with one primary seasonality. For the ETS model, we employ a rolling origin evaluation, which means that the model is re-fitted every day. The hyper-parameters of the LSTM model are determined by a random search in combination with cross-validation on the training set (Bergstra and Bengio, 2012). We train an ensemble of 50 LSTM models for each target (i.e. daily, hourly, hourly profile) and employ the median ensemble operator to obtain the final prediction (Barrow et al., 2010; Kourentzes et al., 2014).

We do not consider other prediction methods as our goal is to investigate the influence of the forecasting phase on the operational performance. The selected methods offer different characteristics and are sufficient to answer the research questions.

6.1.4.2 Results

We subsequently present the results of the two main parts of the proposed approach.

6.1.4.2.1 Forecasting

We compute and evaluate forecasts for three different targets: daily demand (see Table 6.2), hourly profile (see Table 6.3), and hourly demand (see Table 6.4). The goal is to provide hourly forecasts that are consolidated with the daily demand and serve as input for the baking plan generation.

The main observation is that the ranking of the methods is mostly independent from the forecast accuracy measure and forecast target. The ML method LSTM constantly outperforms all other approaches, but there is also a significant gap between ETS and the baseline methods, which supports the plausibility of the results. With respect to the baselines, we can report that it is not sufficient to only consider a single value as S-Naïve provides worse results than any other evaluated method. S-Median is more accurate than S-Mean as it is more robust with respect to outliers.

For the daily forecasts (see Table 6.2), the bottom-up approach to calculate the forecasts is either comparable with direct forecasts or slightly worse. However, the exceptions are S-Median and LSTM which systematically underestimate the demand at the daily level even though the hourly forecasts are reasonably accurate. A reason for this is that the data on the hourly level is quite noisy. For example, some products are not sold or there is no demand for goods in every hour causing frequent zeros in the time series. Consequently, the service level for LSTM (bottom-up) is only 19.4% which is far below the expected service level of unbiased forecasts (50%). While LSTM (bottom-up) has the lowest RMSE, it also has the lowest overage rate and lowest fill rate. Despite the huge bias, it is still the second best method with respect to the total loss but it is significantly outperformed by the LSTM model

Method	Mode	MASE	SMAPE	RMSE	MAE	SL	FR	OR	Loss
S-Naïve	direct	0.980	29.47	66.27	21.39	52.3	88.2	21.5	33.3
	bottom-up	0.980	29.47	66.27	21.39	52.3	88.2	21.5	33.3
S-Mean	direct	0.814	23.74	47.94	17.83	55.7	92.1	20.3	28.2
	bottom-up	0.818	24.05	48.41	17.86	54.3	91.5	19.5	28.0
S-Median	direct	0.773	22.77	43.96	16.30	53.8	91.4	17.4	26.1
	bottom-up	0.852	27.96	44.95	16.84	37.0	84.0	10.1	26.1
ETS	direct	0.726	21.72	42.42	15.62	55.7	92.5	17.5	24.9
	bottom-up	0.735	22.06	42.15	15.77	58.4	**93.0**	18.5	25.5
LSTM	direct	**0.674**	**20.06**	34.38	**14.67**	**50.4**	92.0	14.9	**23.0**
	bottom-up	0.881	27.56	**33.08**	16.65	0.194	80.0	**4.9**	24.9

Table 6.2: Daily forecast performance based on $24,194$ observations per method. The bottom-up predictions are obtained from the direct hourly forecasts (see Table 6.4). The best method in each column is statistically different from the other evaluated methods.

Method	Mode	MASE	MAE	RMSE
S-Naïve	direct	1.108	0.063	0.104
	one	1.108	0.063	0.104
S-Mean	direct	0.935	0.053	0.083
	one	0.939	0.053	0.083
S-Median	direct	**0.881**	0.051	0.085
	one	0.925	0.054	0.092
ETS	direct	**0.873**	0.049	0.076
	one	0.880	0.049	0.078
LSTM	direct	0.875	**0.048**	**0.075**
	one	0.876	0.048	0.075

Table 6.3: Hourly profile forecast performance based on $324,948$ observations per method. As the sum of the direct prediction is not necessarily equal to one per day, we also scale them to one in order to be able to distribute the full daily demand. The best method is shown in boldface and underlined while methods that are not significant different at 0.05 significance level are only print in bold.

that computes daily forecasts directly. Based on the conducted experiments, we conclude that the daily demand should be directly forecast.

The relative advantage of LSTM diminishes for the profile forecasts (see Table 6.3) which means that the intraday demand profile is quite robust over time. As the profile forecasts are intended to be used for a top-down distribution of the daily quantity, we also scale the sum per day to one. The post-processing step only negatively affects S-Median whose sum was frequently smaller than one.

The forecast accuracy for hourly sales is presented in Table 6.4. We compute the hourly demand top-down by relying on the direct daily forecasts and the hourly profile forecasts. The top-down approach leads to comparable results for most approaches. Only S-Median and LSTM, which underestimate the daily demand, perform noticeably worse. The results show that it is important to measure the performance indicators at different levels of aggregation. Nevertheless, LSTM (top-down) is still more accurate at the hourly level than any other forecasting method except LSTM (direct). In order to make the intraday forecasts more

Method	Mode	MASE	MAE	RMSE
S-Naïve	direct	1.053	4.586	12.639
	top-down	1.053	4.586	12.639
S-Mean	direct	0.898	3.741	9.607
	top-down	0.902	3.712	9.405
S-Median	direct	0.835	3.585	9.287
	top-down	0.876	3.674	9.292
ETS	direct	0.835	3.375	8.586
	top-down	0.835	3.345	8.575
LSTM	direct	**0.778**	**3.099**	**7.456**
	top-down	0.822	3.181	7.490

Table 6.4: Hourly forecast performance based on $324,948$ observations per method. The top-down predictions are obtained from the daily forecasts (mode: direct) (see Table 6.2) and the predicted day profiles (mode: one) (see Table 6.3). The best method in each column is statistically different from the other evaluated methods.

reliable, it may be helpful to consider the data at a lower granularity (e.g. morning, midday, afternoon) for products that are usually not sold every hour.

In summary, we rely on the hourly forecasts that are obtained from the top-down distribution of the daily forecast using the intraday profile forecasts for the generation of the schedules. For the present use case, it is a requirement that the daily order quantity has to be processed during the same day which makes it reasonable to put more emphasis on the forecast accuracy at the daily level.

6.1.4.2.2 Intraday Baking

While the demand forecasts are a necessary input for the scheduling problem, they do not represent the decisions. We want to investigate the effect of the forecast accuracy on the operational performance. Thus, we compute the fill rate and overage rate at the end of the day. Moreover, we determine the average age of goods at the selling time if the computed schedule is executed accordingly. In order to obtain the key figures, we iteratively compute the shelf load. The shelves are empty at the beginning of the day and directly filled after a job ends, i.e., starting time of the job plus its duration. If the demand can be fulfilled, the shelf load is reduced accordingly. The shelf load cannot be below zero and items are only removed if they are sold, i.e., they are not removed during the day by the store personnel. However, items that are not sold by the end of the day have to be discarded.

We use the hourly forecasts (mode: top-down) presented in the previous section in order to create the instances of the scheduling problem as described in Section 6.1.3.2. Additionally, we also consider the perfect forecast (i.e. sales) to validate our results. A planning step of the schedule comprises 5 minutes. Hence, we linearly distribute the hourly data to the planning steps. As the scheduling problem is based on demand given in integers, we add fractions to the earlier time step and reduce the succeeding steps accordingly. Moreover, baking should end a couple of hours before the store closes which makes it necessary to prepone the quantities that are required to fulfill the expected demand of the last opening hours.

Intraday Baking	Method	Age	Service Level	Fill Rate	Overage Rate	Loss
no	Perfect	55.31	100.0	100.0	0.0	0.0
	S-Naïve	**49.41**	88.0	89.2	20.0	30.9
	S-Mean	50.93	91.0	92.0	20.1	28.1
	S-Median	50.50	90.3	91.3	17.2	25.9
	ETS	51.20	**91.6**	**92.5**	17.4	24.9
	LSTM	50.67	90.9	92.0	**14.8**	**22.9**
yes	Perfect	25.43	98.4	98.4	1.6	3.3
	S-Naïve	**28.90**	84.9	84.9	24.3	39.4
	S-Mean	30.25	89.3	89.3	22.8	33.5
	S-Median	29.60	88.4	88.4	20.1	31.7
	ETS	30.50	**90.1**	**90.0**	19.9	29.8
	LSTM	30.05	89.5	89.5	**17.3**	**27.7**

Table 6.5: Scheduling: Operational key figures without and with intraday baking. The best method in each column is statistically different from the other evaluated methods. The key figures are given for *intraday baking* and alternatively for *no intraday baking*, i.e., all items are baked before the store opens. The variant *no intraday baking* is not desirable as the goods deteriorate too quickly but it illustrates the performance loss due to intraday decisions. The average age of goods at selling time is given in planning steps, i.e., one planning step comprises five minutes.

We will investigate the impact of the prediction model on the operational performance (1), the general benefits gained due to intraday baking (2), and also discuss the characteristics of the scheduling tasks as well as the optimization of the scheduling problem (3).

Effect of the Forecasting Model

An important observation is that the operational performance is directly linked to the accuracy of the provided forecasts (see Table 6.5 (intraday baking: yes)). This is very apparent as perfect forecasts (i.e. sales) substantially outperform the actual forecasts with respect to all key figures. A comparison of the forecasting methods reveals that not only the ranking based on the forecast accuracy (see Section 6.1.4.2.1) is preserved but also that the relative difference with respect to the total loss is fairly comparable. Hence, we can report that the choice of the prediction model is the most crucial decision, while the scheduling phase has a significantly smaller impact. Moreover, we notice that the scheduling phase has only a minor impact on the operational performance as perfect forecasts (i.e. sales) lead to an almost perfect performance.

The baseline S-Naïve does more often underestimate the demand, which leads to a lower age of goods compared to the more advanced prediction models. On the contrary, it is also true that overestimating the demand at the beginning of the day makes customers buy older items later due to the "first in - first out" assumption. However, the absolute average age difference is under 10 minutes for all evaluated forecasting methods and has no practical relevance. Moreover, we want to point out that the reported service level is measured based on the observations on each planning step (i.e. 5 minutes) and thus rather comparable to the fill rate. Even without dedicated safety stocks, it is possible to serve around 90% of the customers. A comparison between LSTM and ETS reveals that the fill rate of ETS is only 0.5 percentage points higher but the overage rate of LSTM is 13.1% lower while the average age

of goods is similar. Hence, we can conclude that the LSTM is the best approach among the evaluated methods, which is also reflected in the lowest total loss. It is important to limit the overage of goods as a higher fill rate or service level can be achieved by adding safety stock. For instance, the fill rate of LSTM can be increased while still maintaining a lower loss rate than ETS.

Effect of intraday Baking

We want to measure the effect of the scheduling part of our solution approach by comparing the operational key figures with a *no intraday baking* approach (see Table 6.5). *No intraday baking* means that the full predicted daily demand is placed on the shelves when the store opens. In practice, this might not be feasible as the available shelf space is limited, which makes filling up shelves still a requirement.

A major reason to bake the goods during the day is to provide them as fresh as possible. Hence, we compare the average age of sold goods in order to measure this effect (see Table 6.5). When comparing the average age of goods between *intraday baking* and *no intraday baking*, it has to be considered that the difference is underestimated. First, the baking process of all provided items cannot exactly end when the stores open. The goods would need to be baked in a separate facility, which would also add additional delivery time. Second, at the beginning of the day, the difference between both approaches is negligible while larger differences are expected in the later parts of the day. We note that the average age of sold goods can be reduced by roughly 54%, i.e., from 04:37h to 02:07h, for perfect forecasts (i.e. sales) and 41%, i.e., from 04:13h to 02:29h, for the schedules based on forecasts (see Table 6.5). Hence, intraday baking allows to significantly reduce the age of sold goods. It has to be noted that the age is correlated with the fill rate, e.g., by only serving customers in the mornings, a very low average age can be measured. Thus, the age has to be concurrently viewed with other key figures like the fill rate. For instance, for S-Naïve the average age of sold goods is comparable low, but this is also true for the fill rate as the predictions are not well aligned with the demand which also leads to high overages.

With respect to the other key figures, we notice that the scheduling part has a negative impact. The reason for this is that some items are not baked in time. Hence, a part of the demand cannot be fulfilled, which negatively influences the fill rate, overage rate, and total loss. For the perfect forecast (i.e. sales), the total loss only increases by 3.3 percentage points as 1.6% of the demand cannot be fulfilled. For the actual forecasting methods (e.g. LSTM), the increase of total loss is on average 5.9 percentage points (22%). The relative increase of the overage rate is most noticeable and can be decreased by increasing the age of goods, e.g., by baking some goods earlier.One way to achieve this is to set asymmetric penalties for earliness and tardiness.

Schedule Optimization

The jobs of the scheduling problems are based on the forecasts. For every day and store, we create a schedule considering all articles that are relevant for intraday baking. On every day,

	Jobs		Jobs (grouped)	
Method	**N**	**avg. penalty**	**N**	**avg. penalty**
Perfect	120.8	1.294	32.0	1.366
S-Naïve	122.5	1.308	32.5	1.370
S-Mean	124.1	1.225	32.9	1.281
S-Median	121.0	1.195	32.1	1.249
ETS	123.0	1.206	32.7	1.267
LSTM	122.9	1.228	32.6	1.275

Table 6.6: Scheduling: Number of jobs. The average planning horizon comprises 143.3 steps ($\approx$ 12 hours). Due to the aggregation, the number of jobs that need to be scheduled can be reduced by 73.5%.

		P1	P2	P3	P4	P5
jobs	**N**	59.2	22.9	24.5	6.4	9.6
	pct.	47.7%	18.4%	20.6%	5.1%	8.2%
jobs (grouped)	**N**	15.2	6.1	6.5	2.0	2.8
	pct.	46.0%	18.5%	20.5%	6.0%	9.0%
duration		4	5	3	4	11

Table 6.7: Scheduling: Average number of jobs (N) per program (P1 - P5) before and after grouping.

we have to schedule on average more than 120 jobs per store, i.e., baking trays that have to be put in an oven (see Table 6.6). The jobs can be grouped by their program assignment, which reduces the problem size by 73.5% and ensures a high utilization of the ovens. The average penalty for earliness or tardiness is around 1.30 after grouping. The majority of the jobs belong to program P1 (46%), P2 and P3 each cover around 20% while P4 and P5 account together for only 15% (see Table 6.7). Hence, most jobs have a short duration of less than 5 planning steps (25 minutes). Longer baking durations are not usual for goods that are baked in the stores. The average planning horizon depends on the opening hours of the stores and comprises 143.3 steps ($\approx$ 12 hours). We notice that the average number of jobs is fairly comparable among the forecasting methods while the operational performance is still significantly different (see Table 6.5). Hence, jobs derived from more accurate prediction models are better aligned with the actual demand, which translates to a better performance.

In order to solve the scheduling problems, we employ the rolling approach outlined in Section 6.1.3.2 as this significantly reduces the runtime. We compared both approaches for

	Runtime [min]		Objective		Objective per job	
Method	**mean**	**median**	**mean**	**median**	**mean**	**median**
Perfect	5.224	1.132	17.313	5.352	0.458	0.203
S-Naïve	9.728	1.553	15.961	5.213	0.426	0.194
S-Mean	7.928	1.603	13.400	5.313	0.367	0.197
S-Median	8.148	1.558	12.932	4.804	0.358	0.179
ETS	6.790	1.559	12.668	5.228	0.355	0.198
LSTM	7.045	1.588	12.785	4.935	0.355	0.192

Table 6.8: Scheduling: The results concerning the optimization of the linear programs.

solving the schedules obtained from perfect forecasts (i.e. sales) and can report that roughly 30% of the scheduling problems could not be solved within 2 hours. For the successfully terminated schedules, we can report that the objective of the rolling approach was less than 10% higher but the runtime could be reduced by 98% compared to directly optimizing the schedule for the whole day. Those results make it reasonable to rely on the rolling approach to solve the schedules as it is much faster and makes the application of our solution approach more feasible. However, the presented results indicate that a slightly increased objective of the optimization problem has no noticeable effect on the operational performance, i.e., a globally optimal schedule is not required.

In general, the objective of the optimized schedules cannot be compared among different methods as each scheduling instance depends on the initial forecasts which determine the number of jobs as well as their deadlines. Hence, the objectives are not directly comparable if the forecasts differ. Nevertheless, we present the results concerning the optimization of the linear programs as they are an additional indicator for the suitability of our solution approach. It takes on average roughly 8 minutes to compute a schedule, but a majority can be computed in less than 2 minutes. The given runtimes are only estimations because we conducted the experiments on a virtual machine and run several processes in parallel. With respect to the objective, we measure that the average penalty per job is 0.387 after the optimization. We link this result to the average earliness and tardiness penalty per job which is 1.30 (see Table 6.6) and infer that the average deviation of a job from the deadline is only 0.30 planning steps, i.e., less than 90 seconds. Hence, the rolling approach to optimize the scheduling problem is reasonably accurate. Moreover, the equipment of the stores with respect to the available ovens is sufficient to fulfill the demand.

6.1.5 Conclusion

We introduced a solution approach for intraday shelf replenishment of perishable goods. Its purpose is to assist the store personnel by baking goods during the day. Therefore, we compute hourly demand forecasts that are used to optimize a schedule that reflects a baking plan. The baking plan can either be provided as part of an interactive mobile application or be print on paper depending on the requirements and preferences of the bakery. Based on our empirical evaluation, we conclude that our solution approach serves its purpose. If the resulting schedules are executed as suggested, most customers can be served with freshly baked goods. The average age of goods is significantly lower due to intraday baking in comparison with baking before the store opens.

With respect to the forecasting phase of our approach, we can report that the ML model outperformed the reference methods for all evaluated levels. A general observation for perishable goods is that the demand at the hourly level is quite noisy and hard to predict. As the intraday demand profiles are more stable compared to the actual demand, it is advisable to follow a top-down forecasting approach. We also measured that the operational perfor-

mance mostly depends on the accuracy of the demand estimation, which means that it is most beneficial to develop an accurate prediction model.

In terms of the applicability of our approach in a large-scale application scenario that requires offering more than 100 baking plans per day, the time to solve a single schedule, which was on average above 5 minutes (see Table 6.8), could be an obstacle. However, we need to point out that the impact of the scheduling phase is overall negligible, which means that an exact solution might not be required. Hence, the results based on the proposed problem formulation as an integer linear program can serve as a reference for heuristics that can be developed. Our proposed problem formulation and the considered evaluation criteria are closely linked to the process in the stores and do not depend on the actual solution approach. Hence, the evaluation criteria can be used to compare different solution approaches for the scheduling problem.

While we focus on the case of a typical bakery that is daily delivered with goods that need to be baked during the same day, we want to highlight that our approach is also applicable for scenarios that enable real-time intraday adjustments. For instance, the bake-off sections in supermarkets also rely on intraday baking, but the unprocessed goods can be in storage at the store for several days. Hence, the daily quantity has not to be fixed before the store opens. Consequently, the forecasts as well as the resulting schedules can be updated during the day. For instance, it is possible to set a higher service level for the first half of the day in order to serve the full customer demand. At midday, the current shelf load can be aligned with the expected demand for the remaining opening hours. It is also possible to update the demand estimate for the remaining hours during which the store is open. For instance, the hourly demand profiles can be used to interpolate sales in order to obtain an updated estimation of the demand for the second half of the day (Lau and Lau, 1996). This should improve the operational performance as the absolute uncertainty associated with the second half of the day is less than for the whole day.

Part IV

Wrap-up

7 Conclusions

At the beginning of the thesis, we listed central research questions that are the subject of the main parts of the thesis (see Section 1.3). In Section 7.1, we provide a summary of our research and offer answers to the research questions. Subsequently, we briefly discuss practical and managerial implications of the deployment of a decision support system (DSS) in the considered application domain in Section 7.2. Finally, we outline further research opportunities that extend or enhance our work in Section 7.3.

7.1 Summary

Our research is motivated by the requirements of retailers offering perishable goods or more precisely baked goods like buns and breads. Frequent ordering decisions are required due to high rates of deterioration of perishable goods. Store managers are typically responsible for making such decisions that are critical for the performance of the company. This approach is not optimal as store managers rely predominantly on their experience and have limited information about the sales history. This approach is quite common in the fresh food sector (van Donselaar et al., 2006) even though it is very inefficient, i.e., it is time consuming and not reliable as the skills of the store managers differ across the stores.

However, companies in the retail industry operate numerous stores and gather a growing sales history over the years. Additionally, the sales history can be enhanced with external information like calendric events or information about the local environment of each store. Hence, a large data pool is available for exploitation in order to build a DSS that assists or even automates ordering decisions and shelf replenishment decisions.

From existing literature, it is unclear how such a data pool can be exploited and what methods are suitable to optimize decisions based on data. For instance, the literature on retail forecasting is mostly concerned with weekly data and statistical time series methods or simple linear regression models. More general studies on business forecasting often emphasize that pure Machine Learning (ML) methods perform rather poorly and are not suitable for time series forecasting. Operations research literature is mostly concerned with the theoretical properties of inventory models and does typically rely on specific assumptions on the demand distribution. Hence, the literature streams on demand forecasting and inventory management rarely overlap. Our goal is to connect these literature streams by answering our central research question:

RQ1 How can available data be leveraged in order to support and optimize operational decisions in the present application scenario?

In order to answer this question, we analyze the whole process from data to decisions and propose data-driven solution approaches that are able to leverage large datasets. For this purpose, we study inventory management models for perishable goods which optimize order quantities that maximize the profit. The considered inventory management models, i.e., newsvendor models, require information about the demand distributions and costs. To this end, we investigate different ways to leverage data in order to contribute to the following research questions:

RQ2 Are data-driven methods for inventory management of perishable goods a viable alternative to model-based approaches?

RQ3 Can the typically separated phases of estimation and optimization (SEO) be in integrated in a single optimization problem (IEO)?

Newsvendor models are usually solved in two phases: First, the demand distribution is estimated. Second, the inventory decision is optimized. Hence, we rely on data-driven methods, i.e., ML methods, to estimate the demand distributions and demand uncertainty. The key result of our evaluation is that data-driven approaches outperform their model-based counterparts in most cases if enough training data is available (i.e. at least 150 data points). Moreover, it is better to rely on empirical forecast errors rather than specific error distributions in order to model uncertainty. Hence, data-driven decision support without specific assumptions about the demand distribution, which is unknown in real-world settings, is actually feasible and leads to an improved operational performance. We also propose solution approaches that integrate the estimation step and the optimization step in a single optimization problem. Those approaches based on artificial neural networks (ANNs) are very competitive which is in particular beneficial for the multi-product newsvendor problem where the optimization phase is computationally demanding. However, IEO performs noticeably worse compared to SEO for target service levels above 80% in the single product newsvendor problem which implies that there are opportunities for improvement.

RQ4 Has the forecast accuracy of a prediction model a noticeable influence on the operational performance?

The initial forecasts do not reflect the optimal decisions as safety stock has to be added. Moreover, the estimated quantities are only input of a scheduling problem that needs to be solved in order to obtain a baking plan that reflects intraday decisions. However, despite the fact that the optimized quantity can significantly differ from the initial point forecast, the performance is highly correlated with the forecast accuracy. This can be explained with the fact that less uncertainty is involved if the prediction is comparatively precise. In the context of intraday decision support, accurate predictions cause a better alignment of the baking

process with the actual demand. Hence, a larger share of the demand can be fulfilled with even fresher goods. However, the results of the evaluation of the multi-product newsvendor model with substitution revealed that it is even more important to correctly model the specifics of the use case. For instance, the decisions are not optimal if substitution can be assumed but it is not considered by the inventory model. Similarly, if the costs of demand underestimation and demand overestimation are not symmetric, it is important take demand uncertainty into account in order to determine optimal order quantities. However, if the correct assumptions are made, the model with the highest forecast accuracy also yields higher profits. Hence, it is reasonable to study which forecasting models are suitable and what influences their performance:

RQ5 Are Machine Learning methods suitable for retail demand forecasting? What factors affect the performance of Machine Learning methods?

We conducted comprehensive experiments in order to demonstrate the suitability of ML methods in the present application scenario. The most important result is that ML methods are indeed a viable alternative to established approaches for large-scale retail forecasting. ML methods require more training data in order to produce reliable results but are also the only methods that continuously improve as more data becomes available. It is in particular beneficial to pool data across stores and articles. If training data is pooled, the learning algorithm can also extract more value from additional explanatory feature data and overfitting is less likely. Models that are trained based on a pooled dataset are quite robust and do not require frequent re-training which makes the application of such sophisticated models in productive settings feasible. We can also report that ANNs outperform gradient boosted decisions trees. Moreover, a transformation of the regression problem to a classification problem is beneficial and has also the advantage that the predictions can be interpreted as density forecasts. However, if only a limited amount of data is available, it is reasonable to rely on a statistical time series model.

7.2 Practical & Managerial Implications

Different parts of the presented solution approaches are implemented at bakery chains (Huber et al., 2017). The roll-out of the DSS had various effects on the companies. In the past, the ordering decisions were mostly based on the judgment of store managers or regional managers that had no systematic access to information of past sales. Due to the use of a DSS, store managers are no longer required to estimate future demand and to make inventory decisions. Hence, the required skill set of the store managers changes and the time saved can be used for other tasks. Hence, the DSS provides already a benefit even if it would only match the store managers' performance with respect to the decision quality. Actually, the employed solution approaches led to an increased product availability but more noticeably to a reduction of discarded goods because past service levels were already quite high. Moreover, the ownership of the forecasts is no longer with the store managers. Instead, dedicated positions at

the headquarters were created that are in charge of monitoring the recommendations provided by the DSS. Thereby, they rely in particular on the aggregated forecasts at different levels of the hierarchy. In this regard, the aggregated forecasts at higher levels of the hierarchy are valuable as they are based on less noisy data and, thus, are more accurate than forecasts at the store-article level which helps to identify erroneous decision recommendations.

7.3 Future Work

We conclude this thesis with an outline of future work. This includes extensions as well as further improvements of the proposed approaches. Moreover, future work can validate our answers to the research questions and address limitations of the presented research. In the following, we dedicate a section to forecasting and another section to decision support.

A general limitation of our study is that we are only concerned with a single application scenario and do only use proprietary datasets. While this is rather common in certain research communities, where also real-world data is rarely used, it still hinders to some degree the reproducibility of research. Thus, it would be interesting to repeat the analysis on other datasets, including other products and application scenarios (e.g. e-commerce). For instance, the newsvendor models are applicable to perishable products with repetitive sales (e.g. bread, fresh produce, newsprint). Moreover, in some situations little or no historical data may be available (e.g. fashion, electronics, or sport events). In that case, forecasting requires other leading indicators than historical sales. It will be interesting to investigate the performance of alternative approaches to derive decisions from data under those circumstances.

Forecasting

We noticed that the initial forecast accuracy has a significant impact on the operational performance. Hence, it is reasonable to study other approaches to further improve the predictions.

Methods. We only evaluated standard versions of state-of-the-art ML methods. However, recently architectures based on ANNs have been proposed that lead to promising results. For instance, ES-RNN by Slawek Smyl, who combines exponential smoothing and recurrent neural networks, won the M4 Forecasting Competition in 2018 (Makridakis et al., 2018b). Oreshkin et al. (2019) propose a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers which also performs reasonably well on the dataset of the M4 competition.

Combination of methods. We also noticed that various model types provide different forecasts. Hence, an easy way to boost the performance is to build ensembles of different types of models (e.g. neural network, decision trees) or different learning tasks (e.g. regression, classification). Another possibly is to build stacked models that are more computationally demanding and usually require more training data but often improve results.

Leveraging the hierarchies. We only relied on the hierarchies to illustrate the competitiveness of ML methods at different levels and to show possibilities to reduce the computational costs. However, methods that compute an optimal hierarchical reconciliation of predictions at different levels have been proposed (Hyndman et al., 2016; Pennings and van Dalen, 2017; Wickramasuriya et al., 2019) and also successfully applied in other settings (Kourentzes and Athanasopoulos, 2019). Hence, leveraging the organizational hierarchy (e.g. stores vs. regions), the product hierarchy (e.g. article vs. category), and the temporal hierarchy (e.g. day vs. hour) can be an opportunity to improve the predictions. Moreover, retailers are usually interested in reconciled predictions with respect to quantities as well as revenue.

Automation. The setup of a ML model is a tedious task without strict guidelines and often more of an art than a science. However, the results of our evaluation indicate that a more elaborate model building and selection process is feasible in a productive setting as the evaluated models are competitive over a long application phase and the ranking is also stable for different forecasting horizons. In order to standardize the process, it would be necessary to employ algorithms for model selection (e.g. neural architecture search), hyper-parameter optimization, and also feature selection as well as feature engineering. With respect to the evaluated ML methods, it would be interesting to investigate if an automatized model building process leads to better predictions. In productive settings, it would also be useful to integrate a feedback loop in order to update and to change the models if the performance decreases.

Transformation to a classification problem. In our study, the approaches based on a transformation of the regression problem to a classification problem have the highest accuracy even though we applied a simple approach to create the target classes. Hence, other approaches for binning the target values should be explored and evaluated. Moreover, it can be discussed if the classification approach is a suitable method for density forecasts on large datasets as our preliminary experiments indicate.

Feature importance. Adding explanatory feature data to the training data of a ML model significantly improved the results. However, we did not conduct an in-depth analysis of the importance of the used features. The features are selected based on domain knowledge but it is unclear how they influence the performance. We suspect that a feature importance analysis of the trained models leads to domain-specific insights and a better understanding of the advantages of ML.

Cold start problem. Retailers frequently change the assortment, offer new products, or even open new stores. In such cases, a demand history is not available which hinders the usage of time series models. However, we offered empirical evidence that data pooling improves the results for existing time series. Hence, it would be interesting to analyze how well the trained models perform on completely unseen data, e.g., from a new store.

Decision Support

The interface between Machine Learning and Operations Research (e.g. inventory management) is an active field of research that provides a variety of opportunities for future research.

We introduced solution approaches for newsvendor problems that integrate the estimation phase and the optimization phase by leveraging large datasets in order to directly compute a decision from data. However, we still need to prepare the data, i.e., decensor and deflate the historical sales, and estimate substitution rates in order to apply our approach. Hence, another step towards integrated data-driven inventory management would be to include these steps in the main optimization problem.

Moreover, an extension of the analysis could include multi-period considerations, where products can be reordered during the selling season. For instance, the intraday decisions are essentially a multi-period case, but even certain bakery products can be sold over multiple days. Thus, expanding the model to a multi-period inventory model is reasonable. It would widen the application of the model to many other grocery products that can be reordered during the selling season. There are several papers that deal with the multi-period problem with an unknown demand distribution (e.g. Godfrey and Powell (2001), Levi et al. (2007)). Given the inherent similarity between reorder point calculations and newsvendor trade-offs, one may expect ML approaches to also be beneficial in that context. In our application scenario, there is no lead time. However, in other problem settings, lead time plays an important role. Prak et al. (2017) show that using one-period-ahead forecast errors to optimize inventories leads to insufficient safety stock levels in case of a positive lead time. ML methods are able to estimate prediction intervals for multi-step forecasts (Makridakis et al., 2018b).

With respect to the generation of the intraday baking plans, it would be reasonable to develop faster heuristics to solve the scheduling problem. A solution approach based on reinforcement learning is an interesting direction to extend the presented approach and would also be a contribution to scheduling literature. If the scheduling problem can be solved with reinforcement learning, it should also be possible to integrate the demand estimation step in the optimization problem.

Bibliography

J. Aastrup and H. Kotzab. Forty years of out-of-stock research–and shelves are still empty. *The International Review of Retail, Distribution and Consumer Research*, 20(1):147–164, 2010.

L. Aburto and R. Weber. Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1):136 – 144, 2007.

D. Adebanjo. Understanding demand management challenges in intermediary food trading: a case study. *Supply Chain Management: An International Journal*, 14(3):224–233, 2009.

D. Adebanjo and R. Mann. Identifying problems in forecasting consumer demand in the fast moving consumer goods sector. *Benchmarking: An International Journal*, 7(3):223–230, 2000.

M. Adya and F. Collopy. How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*, 17(5-6):481–495, 1998.

N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. *Econometric Reviews*, 29(5-6): 594–621, 2010.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

I. Alon, M. Qi, and R. J. Sadowski. Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, 8(3):147–156, 2001.

R. Anupindi, M. Dada, and S. Gupta. Estimation of consumer demand with stock-out based substitution: An application to vending machine products. *Marketing Science*, 17(4):406–423, 1998.

N. S. Arunraj and D. Ahrens. A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, 170:321–335, 2015.

G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, and F. Petropoulos. Forecasting with temporal hierarchies. *European Journal of Operational Research*, 262(1):60–74, 2017.

M. Bakker, J. Riezebos, and R. H. Teunter. Review of inventory systems with deterioration since 2001. *European Journal of Operational Research*, 221(2):275–284, 2012.

G.-Y. Ban and C. Rudin. The Big Data Newsvendor: Practical Insights from Machine Learning. *Operations Research*, 67(1):90–108, 2018.

D. Barrow and N. Kourentzes. The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *European Journal of Operational Research*, 264(3):967–977, 2018.

D. Barrow, S. Crone, and N. Kourentzes. An evaluation of neural network ensembles and model selection for time series prediction. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2010.

D. K. Barrow and S. F. Crone. Cross-validation aggregation for combining autoregressive neural network forecasts. *International Journal of Forecasting*, 32(4):1120–1137, 2016.

S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067–7083, 2012.

A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, New Jersey, 2009.

C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, 120:70 – 83, 2018.

J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2546–2554. Curran Associates Inc., 2011.

J. Bergstra, D. Yamins, and D. D. Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning - Volume 28*, pages 115–123. JMLR.org, 2013.

D. Bertsimas and N. Kallus. From predictive to prescriptive analytics. *Management Science*, forthcomin, 2018.

D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

A. L. Beutel and S. Minner. Safety stock planning under causal demand forecasting. *International Journal of Production Economics*, 140(2):637–645, 2012.

B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchette, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, and J. Vanschoren. Aslib: A benchmark library for algorithm selection. *Artificial Intelligence*, 237:41 – 58, 2016.

G. Bontempi, S. B. Taieb, and Y.-A. L. Borgne. Machine Learning Strategies for Time Series Forecasting. In *Business Intelligence*, Lecture Notes in Business Information Processing, pages 62–77. Springer, Berlin, Heidelberg, 2012.

G. E. Box and G. M. Jenkins. *Time series analysis: forecasting and control, revised ed.* Holden-Day, 1976.

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

R. A. Briesch, P. K. Chintagunta, and E. J. Fox. How Does Assortment Affect Grocery Store Choice? *Journal of Marketing Research*, 46(2):176–189, 2009.

E. Brynjolfsson and K. McElheran. The Rapid Adoption of Data-Driven Decision-Making. *American Economic Review*, 106(5):133–139, 2016.

K. Campo, E. Gijsbrechts, and P. Nisol. Towards understanding consumer response to stock-outs. *Journal of Retailing*, 76(2):219–242, 2000.

J. R. Cancelo, A. Espasa, and R. Grafe. Forecasting the electricity load from one day to one week ahead for the Spanish system operator. *International Journal of Forecasting*, 24(4): 588–602, 2008.

A. J. Cannon. Quantile regression neural networks: Implementation in R and application to precipitation downscaling. *Computers and Geosciences*, 37(9):1277–1284, 2011.

R. Carbonneau, K. Laframboise, and R. Vahidov. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, 184(3): 1140–1154, 2008.

C. Chatfield. Prediction intervals for time-series forecasting. In *Principles of Forecasting: A Handbook for Researchers and Practitioners*, pages 475–494. Springer US, Boston, MA, 2001.

S. Chaudhuri, U. Dayal, and V. Narasayya. An overview of business intelligence technology. *Communications of the ACM*, 54(8):88–98, 2011.

T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, 2016.

J. Cheng, Z. Wang, and G. Pollastri. A neural network approach to ordinal regression. In *IEEE International Joint Conference on Neural Networks 2008*, pages 1279–1284, 2008.

C.-W. Chu and G. P. Zhang. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, 86(3):217–231, 2003.

S. A. Conrad. Sales data and the estimation of demand. *Operational Research Quarterly*, 27 (1):123–127, 1976.

L. G. Cooper, P. Baron, W. Levy, M. Swisher, and P. Gogos. PromoCast^{TM}: A New Forecasting Method for Promotion Planning. *Marketing Science*, 18(3):301–316, 1999.

D. Corsten and T. Gruen. Desperately seeking shelf availability: an examination of the extent, the causes, and the efforts to address retail out-of-stocks. *International Journal of Retail & Distribution Management*, 31(12):605–617, 2003.

S. F. Crone and N. Kourentzes. Forecasting seasonal time series with multilayer perceptrons-an empirical evaluation of input vector specifications for deterministic seasonality. In *Proceedings of The 2009 International Conference on Data Mining*, pages 232–238, 2009.

S. F. Crone, M. Hibon, and K. Nikolopoulos. Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, 27(3):635–660, 2011.

A. Curşeu, T. van Woensel, J. Fransoo, K. van Donselaar, and R. Broekmeulen. Modelling handling operations in grocery retail stores: an empirical analysis. *Journal of the Operational Research Society*, 60(2):200–214, 2009.

J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.

G. Di Pillo, V. Latorre, S. Lucidi, and E. Procacci. An application of support vector machines to sales forecasting under promotions. *4OR*, 14(3):309–325, 2016.

S. Divakar, B. T. Ratchford, and V. Shankar. CHAN4cast: A Multichannel, Multiregion Sales Forecasting Model and Decision Support System for Consumer Packaged Goods. *Marketing Science*, 24(3):334–350, 2005.

P. Doganis, A. Alexandridis, P. Patrinos, and H. Sarimveis. Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196–204, 2006.

J. C. Ehrenthal and W. Stölzle. An examination of the causes for retail stockouts. *International Journal of Physical Distribution & Logistics Management*, 43(1):54–69, 2013.

L. W. Emmelhainz, M. A. Emmelhainz, and J. R. Stock. Logistics Implications of Retail Stockouts. *Journal of Business Logistics*, 12(2):129–142, 1991.

A. Farahat and J. Lee. The Multiproduct Newsvendor Problem with Customer Choice. *Operations Research*, 66(1):123–136, 2018.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28*, pages 2962–2970. Curran Associates, Inc., 2015.

R. Fildes, K. Nikolopoulos, S. F. Crone, and A. Syntetos. Forecasting and operational research: a review. *Journal of the Operational Research Society*, 59(9):1150–1172, 2008.

R. Fildes, P. Goodwin, M. Lawrence, and K. Nikolopoulos. Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. *International Journal of Forecasting*, 25(1):3–23, 2009.

R. A. Fildes, S. Ma, and S. Kolassa. Retail forecasting: research and practice, 2018. URL `http://eprints.lancs.ac.uk/128587/`.

M. Fisher and R. Vaidyanathan. A Demand Estimation Procedure for Retail Assortment Optimization with Results from Implementations. *Management Science*, 60(10):2401–2415, 2014.

G. Gallego and I. Moon. The distribution free newsboy problem: Review and extensions. *The Journal of the Operational Research Society*, 44(8):825–834, 1993.

E. S. Gardner. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1): 1–28, 1985.

E. S. Gardner. Exponential smoothing: The state of the art — Part II. *International Journal of Forecasting*, 22(4):637–666, 2006.

T. Gneiting. Making and Evaluating Point Forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011.

T. Gneiting and M. Katzfuss. Probabilistic Forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

G. A. Godfrey and W. B. Powell. An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Science*, 47(8):1101–1112, 2001.

C. W. Gross and J. E. Sohl. Disaggregation methods to expedite product line forecasting. *Journal of Forecasting*, 9(3):233–254, 1990.

T. W. Gruen and D. S. Corsten. *A comprehensive guide to retail out-of-stock reduction in the fast-moving consumer goods industry*. Grocery Manufacturers of America, 2007.

T. W. Gruen, D. S. Corsten, and S. Bharadwaj. *Retail out-of-stocks: A worldwide examination of extent, causes and consumer responses*. Grocery Manufacturers of America, 2002.

Ö. Gür Ali, S. Sayın, T. van Woensel, and J. Fransoo. SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348, 2009.

P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez. Ordinal Regression Methods: Survey and Experimental Study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, Jan. 2016.

R. Hackathorn. The BI watch real-time to real-value. *DM Review*, 14:24–29, 2004.

G. Hahn and J. Packowski. A perspective on applications of in-memory analytics in supply chain management. *Decision Support Systems*, 76:45 – 52, 2015.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, NY, USA, 2009.

A. H. Hübner and H. Kuhn. Retail category management: State-of-the-art review of quantitative research and software applications in assortment and shelf space management. *Omega*, 40(2):199–209, 2012.

R. Helm, T. Hegenbart, and H. Endres. Explaining costumer reactions to real stockouts. *Review of Managerial Science*, 7(3):223–246, 2013.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

C. Hofer, M. A. Waller, I. Moussaoui, B. D. Williams, and J. A. Aloysius. Drivers of retail on-shelf availability: Systematic review, critical assessment, and reflections on the road ahead. *International Journal of Physical Distribution & Logistics Management*, 46(5): 516–535, 2016.

E. Hofmann and E. Rutschmann. Big data analytics and demand forecasting in supply chains: a conceptual analysis. *The International Journal of Logistics Management*, 29(2):739–766, 2018.

C. W. Holsapple and M. P. Sena. ERP plans and decision-support benefits. *Decision Support Systems*, 38(4):575–590, 2005.

C. W. Holsapple and A. B. Whinston. *Decision support systems: A knowledge-based approach*. West Publishing Company, 1996.

M. Holweg, S. Disney, J. Holmström, and J. Småros. Supply chain collaboration:: Making sense of the strategy continuum. *European Management Journal*, 23(2):170–181, 2005.

T. Hong and S. Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

T. Huang, R. Fildes, and D. Soopramanien. The value of competitive information in forecasting FMCG retail product sales and the variable selection problem. *European Journal of Operational Research*, 237(2):738–748, 2014.

J. Huber, A. Gossmann, and H. Stuckenschmidt. Cluster-based hierarchical demand forecasting for perishable goods. *Expert Systems with Applications*, 76:140–151, 2017.

J. Huber, S. Müller, M. Fleischmann, and H. Stuckenschmidt. A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278(3):904–915, 2019.

A. Hübner, H. Kuhn, and S. Kühn. An efficient algorithm for capacitated assortment planning with stochastic demand and substitution. *European Journal of Operational Research*, 250 (2):505–520, 2016.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439 – 454, 2002.

R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

R. J. Hyndman, A. J. Lee, and E. Wang. Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis*, 97:16–32, 2016.

R. Ibrahim, H. Ye, P. L'Ecuyer, and H. Shen. Modeling and forecasting call center arrivals: A literature survey and a case study. *International Journal of Forecasting*, 32(3):865–874, 2016.

K. B. Kahn. Revisiting top-down versus bottom-up forecasting. *The Journal of Business Forecasting*, 17(2):14, 1998.

W. Kaiser. Fast moving consumer goods. In *Qualitative Marktforschung in Theorie und Praxis*, pages 605–615. Springer, 2011.

M. Kalchschmidt, R. Verganti, and G. Zotteri. Forecasting demand from heterogeneous customers. *International Journal of Operations & Production Management*, 26(6):619–638, 2006.

Y. Kang, R. J. Hyndman, and K. Smith-Miles. Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, 33(2):345–358, 2017.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017a.

J. Ke, H. Zheng, H. Yang, and X. M. Chen. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85:591–608, 2017b.

M. S. Kim. Modeling special-day effects for forecasting intraday electricity demand. *European Journal of Operational Research*, 230(1):170–180, 2013.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

A. G. Kök and M. L. Fisher. Demand estimation and assortment optimization under substitution : Methodology and application. *Operations Research*, 55(6):1001–1021, 2007.

A. G. Kök, M. L. Fisher, and R. Vaidyanathan. Assortment Planning: Review of Literature and Industry Practice. In *Retail Supply Chain Management*, volume 122, pages 99–153. Springer, 2015.

S. Kolassa. Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, 32(3):788–803, 2016.

N. Kourentzes and G. Athanasopoulos. Cross-temporal coherent forecasts for Australian tourism. *Annals of Tourism Research*, 75:393–409, 2019.

N. Kourentzes and F. Petropoulos. Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*, 181:145–153, 2016.

N. Kourentzes, D. K. Barrow, and S. F. Crone. Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, 41(9):4235 – 4244, 2014.

N. Kourentzes, B. Rostami-Tabar, and D. K. Barrow. Demand forecasting by temporal aggregation: Using optimal or multiple aggregation levels? *Journal of Business Research*, 78: 1–9, 2017.

J. P. Lam and M. R. Veall. Bootstrap prediction intervals for single period regression forecasts. *International Journal of Forecasting*, 18(1):125–130, 2002.

H.-S. Lau and A. H. L. Lau. Estimating the demand distributions of single-period items having frequent stockouts. *European Journal of Operational Research*, 92(2):254–265, 1996.

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. In *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, pages 9–48. Springer, Berlin, Heidelberg, 2012.

H. L. Lee, V. Padmanabhan, and S. Whang. The bullwhip effect in supply chains. *MIT Sloan Management Review*, 38(3):93, 1997.

R. Levi, R. O. Roundy, and D. B. Shmoys. Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32(4):821–839, 2007.

R. Levi, G. Perakis, and J. Uichanco. The data-driven newsvendor problem: New bounds and insights. *Operations Research*, 63(6):1294–1306, 2015.

N. Li, K. Wang, and J. Cheng. A research on a following day load simulation method based on weather forecast parameters. *Energy Conversion and Management*, 103:691–704, 2015.

Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 16: 865–873, 2015.

S. Ma and R. Fildes. A retail store SKU promotions optimization model for category multi-period profit maximization. *European Journal of Operational Research*, 260(2):680–692, 2017.

S. Ma, R. Fildes, and T. Huang. Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research*, 249(1):245–257, 2016.

X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):1–26, 2018a.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802 – 808, 2018b.

D. L. Marino, K. Amarasinghe, and M. Manic. Building energy load forecasting using Deep Neural Networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046–7051, 2016.

S. Mou, D. J. Robb, and N. DeHoratius. Retail store operations: Literature review and research directions. *European Journal of Operational Research*, 265(2):399–422, 2018.

A. Musalem, M. Olivares, E. T. Bradlow, C. Terwiesch, and D. Corsten. Structural estimation of the effect of out-of-stocks. *Management Science*, 56(7):1180–1197, 2010.

S. Netessine and N. Rudi. Centralized and Competitive Inventory Models with Demand Substitution. *Operations Research*, 51(2):329–335, 2003.

K. Nikolopoulos, A. A. Syntetos, J. E. Boylan, F. Petropoulos, and V. Assimakopoulos. An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society*, 62(3):544–554, 2011.

B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, 2019. URL `http://arxiv.org/abs/1905.10437`.

A. Oroojlooyjadid, L. V. Snyder, and M. Takác. Applying deep learning to the newsvendor problem. *CoRR*, abs/1607.02177, 2016. URL `http://arxiv.org/abs/1607.02177`.

I. P. Panapakidis. Application of hybrid computational intelligence models in short-term bus load forecasting. *Expert Systems with Applications*, 54:105–120, 2016.

M. Parlar and S. K. Goyal. Optimal Ordering Decisions for two Substitutable Products with Stochastic Demand. *OPSEARCH*, 21(1):1–15, 1984.

C. L. P. Pennings and J. van Dalen. Integrated hierarchical forecasting. *European Journal of Operational Research*, 263(2):412–418, 2017.

G. Perakis and G. Roels. Regret in the newsvendor model with partial information. *Operations Research*, 56(1):188–203, 2008.

F. Petropoulos, S. Makridakis, V. Assimakopoulos, and K. Nikolopoulos. 'Horses for Courses' in demand forecasting. *European Journal of Operational Research*, 237(1):152–163, 2014.

F. Petropoulos, N. Kourentzes, and K. Nikolopoulos. Another look at estimators for intermittent demand. *International Journal of Production Economics*, 181:154–161, 2016.

M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer International Publishing, 5 edition, 2016.

H. Plattner. A common database approach for oltp and olap using an in-memory column database. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, pages 1–2, 2009.

D. J. Power. Decision support systems: a historical overview. In *Handbook on Decision Support Systems 1*, pages 121–140. Springer, 2008.

D. Prak and R. Teunter. A general method for addressing forecasting uncertainty in inventory models. *International Journal of Forecasting*, 35(1):224 – 238, 2019. Special Section: Supply Chain Forecasting.

D. Prak, R. Teunter, and A. Syntetos. On the calculation of safety stocks when demand is forecasted. *European Journal of Operational Research*, 256(2):454–461, 2017.

Y. Qin, R. Wang, A. J. Vakharia, Y. Chen, and M. M. H. Seref. The newsvendor problem: Review and directions for future research. *European Journal of Operational Research*, 213 (2):361–374, 2011.

X. Qing and Y. Niu. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy*, 148:461–468, 2018.

J. Quevedo, J. Saludes, V. Puig, and J. Blanch. Short-term demand forecasting for real-time operational control of the Barcelona water transport network. In *22nd Mediterranean Conference on Control and Automation*, pages 990–995, 2014.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

U. Ramanathan and L. Muyldermans. Identifying demand factors for promotional planning and forecasting: A case of a soft drink company in the UK. *International Journal of Production Economics*, 128(2):538–545, 2010.

U. Ramanathan and L. Muyldermans. Identifying the underlying structure of demand during promotions: A structural equation modelling approach. *Expert Systems with Applications*, 38(5):5544–5552, 2011.

G. Reiner, C. Teller, and H. Kotzab. Analyzing the Efficient Execution of In-Store Logistics Processes in Grocery Retailing –The Case of Dairy Products. *Production and Operations Management*, 22(4):924–939, 2013.

I. Rojas, O. Valenzuela, F. Rojas, A. Guillén, L. J. Herrera, H. Pomares, L. Marquez, and M. Pasadas. Soft-computing techniques and arma model for time series prediction. *Neurocomputing*, 71(4):519–537, 2008.

A.-L. Sachs. Data-driven order policies with censored demand and substitution in retailing. In *Retail Analytics*, volume 680 of *Lecture Notes in Economics and Mathematical Systems*, pages 57–78. Springer International Publishing, 2015.

A. L. Sachs and S. Minner. The data-driven newsvendor with censored demand observations. *International Journal of Production Economics*, 149:28–36, 2014.

B. Sahay and J. Ranjan. Real time business intelligence in supply chain analytics. *Information Management & Computer Security*, 16(1):28–48, 2008.

H. E. Scarf. A min-max solution of an inventory problem. In *Studies in the Mathematical Theory of Inventory and Production*, pages 201–209. Stanford University Press, Stanford, 1958.

J. Schlapp and M. Fleischmann. Technical Note — Multiproduct Inventory Management Under Customer Substitution and Capacity Restrictions. *Operations Research*, 66(3):740–747, 2018.

A. Shapiro. Monte carlo sampling methods. In *Handbooks in Operations Research and Management Science*, volume 10, pages 353–425. Elsevier Science B.V., Boston, USA, 2003.

V. Sikka, F. Färber, W. Lehner, S. K. Cha, T. Peh, and C. Bornhövd. Efficient transaction processing in sap hana database: The end of a column store myth. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 731–742, 2012.

E. A. Silver, D. F. Pyke, and D. J. Thomas. *Inventory and production management in supply chains*. Taylor and Francis, New York, 2017.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

L. J. Soares and M. C. Medeiros. Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data. *International Journal of Forecasting*, 24(4):630–644, 2008.

V. Sridharan, W. L. Berry, and V. Udayabhanu. Freezing the Master Production Schedule under Rolling Planning Horizons. *Management Science*, 33(9):1137–1149, 1987.

D. Srinivasan, C. S. Chang, and A. C. Liew. Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting. *IEEE Transactions on Power Systems*, 10(4):1897–1903, 1995.

A. A. Syntetos, K. Nikolopoulos, and J. E. Boylan. Judging the judges through accuracy-implication metrics: The case of inventory forecasting. *International Journal of Forecasting*, 26(1):134–143, 2010.

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

I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

K. Talluri and G. van Ryzin. Revenue Management Under a General Discrete Choice Model of Consumer Behavior. *Management Science*, 50(1):15–33, 2004.

F. Taube and S. Minner. Data-driven assignment of delivery patterns with handling effort considerations in retail. *Computers & Operations Research*, 100:379–393, 2018.

A. S. Tay and K. F. Wallis. Density forecasting: a survey. *Journal of Forecasting*, 19(4): 235–254, 2000.

J. W. Taylor. A quantile regression approach to estimating the distribution of multiperiod returns. *The Journal of Forecasting*, 19:299–311, 2000.

J. W. Taylor. A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center. *Management Science*, 54(2):253–265, 2007a.

J. W. Taylor. Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational Research*, 178(1):154–167, 2007b.

C. Teller, C. Holweg, G. Reiner, and H. Kotzab. Retail store operations and food waste. *Journal of Cleaner Production*, 185:981–997, 2018.

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

Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang. LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

H. Topaloglu. Joint stocking and product offer decisions under the multinomial logit model. *Production and Operations Management*, 22(5):1182–1199, 2013.

J. R. Trapero, D. J. Pedregal, R. Fildes, and N. Kourentzes. Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, 29(2):234–243, 2013.

J. R. Trapero, N. Kourentzes, and R. Fildes. On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, 66(2):299–307, 2015.

J. R. Trapero, M. Cardós, and N. Kourentzes. Empirical safety stock estimation based on kernel and GARCH models. *Omega*, 84:199 – 211, 2019.

K. van Donselaar, T. van Woensel, R. Broekmeulen, and J. Fransoo. Inventory control of perishables in supermarkets. *International Journal of Production Economics*, 104(2):462–472, 2006.

K. H. van Donselaar, V. Gaur, T. van Woensel, R. A. Broekmeulen, and J. C. Fransoo. Ordering behavior in retail stores and implications for automated replenishment. *Management Science*, 56(5):766–784, 2010.

K. H. van Donselaar, J. Peters, A. de Jong, and R. A. C. M. Broekmeulen. Analysis and forecasting of demand during promotions for perishable items. *International Journal of Production Economics*, 172:65–75, 2016.

H. J. van Heerde, P. S. Leeflang, and D. R. Wittink. The Estimation of Pre- and Postpromotion Dips with Store-Level Scanner Data. *Journal of Marketing Research*, 37(3):383–395, 2000.

H. J. van Heerde, P. S. H. Leeflang, and D. R. Wittink. How Promotions Work: SCAN*PRO-Based Evolutionary Model Building. *Schmalenbach Business Review*, 54(3):198–220, 2002.

G. J. van Ryzin and S. Mahajan. On the Relationship Between Inventory Costs and Variety Benefits in Retail Assortments. *Management*, 45(11):1496–1509, 1999.

T. van Woensel, K. van Donselaar, R. Broekmeulen, and J. Fransoo. Consumer responses to shelf out-of-stocks of perishable products. *International Journal of Physical Distribution & Logistics Management*, 37(9):704–718, 2007.

S. van Zelst, K. van Donselaar, T. van Woensel, R. Broekmeulen, and J. Fransoo. Logistics drivers for shelf stacking in grocery retail stores: Potential for efficiency improvement. *International Journal of Production Economics*, 121(2):620–632, 2009.

S. Viswanathan, H. Widiarta, and R. Piplani. Evaluation of hierarchical forecasting for substitutable products. *International Journal of Services and Operations Management*, 4(3): 277–295, 2008.

G. Vulcano, G. van Ryzin, and W. Chaar. Choice-Based Revenue Management: An Empirical Study of Estimation and Optimization. *Manufacturing & Service Operations Management*, 12(3):371–392, 2010.

G. Vulcano, G. van Ryzin, and R. Ratliff. Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, 60(2):313–334, 2012.

M. Wan, Y. Huang, L. Zhao, T. Deng, and J. C. Fransoo. Demand estimation under multi-store multi-product substitution in high density traditional retail. *European Journal of Operational Research*, 266(1):99–111, 2018.

A. J. Wang and B. Ramsay. A neural network based estimator for electricity spot-pricing with particular reference to weekend and public holidays. *Neurocomputing*, 23(1), 1998.

J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

H. J. Watson. Tutorial: business intelligence-past, present, and future. *Communications of the Association for Information Systems*, 25(1):39, 2009.

S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman. Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

H. Widiarta, S. Viswanathan, and R. Piplani. Forecasting aggregate demand: An analytical evaluation of top-down versus bottom-up forecasting in a production planning framework. *International Journal of Production Economics*, 118(1):87 – 94, 2009.

F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

B. D. Williams and M. A. Waller. Top-down versus bottom-up demand forecasts: The value of shared point-of-sale data in the retail supply chain. *Journal of Business Logistics*, 32(1): 17–26, 2011.

G. Zhang. Neural networks for time-series forecasting. In *Handbook of Natural Computing*, pages 461–477. Springer Berlin Heidelberg, 2012.

G. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35 – 62, 1998.

J. Zhang, W. Xie, and S. C. Sarin. Multi-Product Newsvendor Problem with Customer-driven Demand Substitution : A Stochastic Integer Program Perspective, 2018. URL `https://ssrn.com/abstract=3188361`.

Y. Zhang and J. Gao. Assessing the performance of deep learning algorithms for newsvendor problem. In *Neural Information Processing*, pages 912–921, Cham, 2017. Springer International Publishing.

W. Zinn and P. C. Liu. Consumer response to retail stockouts. *Journal of Business Logistics*, 22(1):49–71, 2001.

W. Zinn and P. C. Liu. A comparison of actual and intended consumer behavior in response to retail stockouts. *Journal of Business Logistics*, 29(2):141–159, 2008.

G. Zotteri, M. Kalchschmidt, and F. Caniato. The impact of aggregation level on forecasting performance. *International Journal of Production Economics*, 93-94:479 – 491, 2005. Proceedings of the Twelfth International Symposium on Inventories.