Data Analytics for Inventory Management

Demand and Lead Time Estimation for Spare Parts Planning

Inauguraldissertation
zur
Erlangung des Doktorgrades
der
Wirtschafts- und Sozialwissenschaftlichen Fakultät
der
Universität zu Köln

2025

vorgelegt
von

Robin Reiners, M.Sc.

aus
Willich

Referent:	Prof. Ulrich W. Thonemann, Ph.D.
Koreferent:	Prof. Dr. Andreas Fügener
Tag der Promotion:	21.11.2025

Acknowledgements

The journey of writing this dissertation has been demanding yet immensely rewarding, made possible through the invaluable support and encouragement of others.

As such, I would like to express my sincere gratitude to my first supervisor, Prof. Ulrich W. Thonemann, whose high standards and guidance have shaped this dissertation. He provided opportunities to grow beyond academia, challenged me to think critically, and encouraged me to pursue ambitious goals, even when my own confidence faltered.

I am equally grateful to my second supervisor, Prof. Dr. Andreas Fügener, who first brought me to the department as a student assistant. While we did not work on shared topics during my dissertation, I greatly appreciate his approachability, sense of humor, and the perspective and guidance he generously offered throughout my time at the departement. I also thank Prof. Dr. Nicolas Fugger for departmenting the examination board and for his valuable feedback during my defense.

I owe special thanks to Stephanie Rauscher, whose reliability and warm efficiency have kept the department running and made everyday work far smoother for everyone.

I was fortunate to spend my time at the department surrounded by colleagues who grew into true friends. Dominik, with whom I shared the most time at the department, made every phase of this journey richer through wide-ranging conversations balancing philosophy and absurdity, “scooter gang” adventures, lively table-tennis matches, and Aperol evenings that made demanding times lighter. Sabrina provided the most valuable guidance on how to navigate a dissertation, often without speaking of research at all, and her perspective proved indispensable. Though very different in nature, Laura’s understanding and calm ensured that our differences never stood in the way, and she listened patiently

whenever broken code tested my limits. Chantale brought structure, humor, and balance to our group and became a trusted friend. Moritz and David embodied the lively chaos of my early days at the department, making even a burning SCO server an oddly enjoyable experience. Anne added grounding and clarity through long bike rides and runs. Valentin and Marco, as part of the next generation, brought energy, humor, and welcome breaks to the routine, earning their place among the wolves. Having made a lasting impression at Christiane's disputation, André has since brought perspective, humor, and many enjoyable La Fonda lunches into my life. Florian enriched every research session with enthusiasm and wit; the ISIR conference remains a lasting memory.

I am also grateful to my colleagues at the department, who made my time at the department an enjoyable experience: Saied Fahrham Nia, Henrik Franke, Nicolas Fugger, Daniel Gass, Theresa Gattermann-Itschert, Dylan Gellert, Christiane Haubitz, Cedric Lehmann, Daniel Piersig, Shuang Shi, Thomas Vogt, Monika Westphal, and Jiaqi Wang.

I owe heartfelt thanks to my friends, whose enduring presence and shared moments whether in motion or in quiet conversation brought balance and joy in this time of my life.

I owe my deepest gratitude to my family, whose constant love, patience, and confidence formed the foundation for everything I have accomplished. Their quiet support gave strength when my own wavered.

Finally, to Pia: you have seen every version of me and have stood by me through all of it. You bring lightness when I am weighed down. You challenge me when I retreat. Building a life of depth and laughter with you is the greatest privilege I know.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
CV	Cross Validation
DRF	Distributional Random Forest
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
MSE	Mean Squared Error
OLS	Ordinary Least Squares
OS	Minority Oversampling
SBA	Syntetos–Boylan Adjustment
SES	Simple Exponential Smoothing
SKU	Stock Keeping Unit
TSB	Teunter–Syntetos–Babai Adjustment
US	Minority Undersampling

Chapter 1

Introduction

1.1 Motivation

Spare parts play a crucial role in ensuring the availability of complex technical systems. Whether in manufacturing, transportation, or energy, equipment downtime caused by faulty or missing parts can lead to substantial costs and service disruptions. To prevent such outcomes, firms maintain inventories of spare parts. Spare parts are characterized by irregular demand, long and uncertain lead times, and a large assortment of parts with heterogeneous characteristics. This makes spare parts planning a particularly challenging field of inventory management (Sherbrooke 2004, Huiskonen 2001).

A central task in spare parts planning is to balance service quality and cost of working capital. Firms aim for high service levels to keep customers satisfied and avoid downtime. At the same time, inventories tie up cash, especially for spare parts that are expensive and slow-moving. Finding the right balance between service level and cost of working capital depends on information about uncertainty in demand and supply, which determines safety stocks and reorder levels.(Silver et al. 1998, Fortuin 1980).

In practice, this information is often inaccurate or incomplete. Lead times stored in enterprise resource planning (ERP) systems do not necessarily reflect actual lead times. Demand forecasts frequently rely on simple models that fail to account for the intermittent nature of spare parts demand. For many parts, data is too sparse to apply standard statistical techniques (Syntetos and Boylan 2005). As a result, firms either hold excess stock to guard against uncertainty or accept lower service levels to reduce capital lock-up Turrini and Meissner (2019).

At the same time, firms collect vast amounts of operational data in their information systems. Purchase orders, master data, and transaction histories provide a rich basis for improving forecasts and planning decisions. Modern data-driven methods, in particular machine learning, are well-suited to extract patterns from such data. These methods enable more accurate predictions, can adapt to heterogeneous part characteristics, and allow for cross-learning between parts with similar features (Makridakis et al. 2018a, 2022). By exploiting these opportunities, firms can improve their spare parts planning, reduce costs, and increase service levels.

This dissertation is motivated by the potential of data-driven methods to address fundamental challenges in spare parts planning. It investigates how modern forecasting techniques can be used to improve the accuracy of planning parameters, how uncertainty in demand can be represented more realistically, and how data scarcity can be mitigated through cross-learning.

1.2 Outline

The dissertation consists of three research papers, each addressing a specific challenge in spare parts planning. Together, they provide a coherent perspective on how data-driven methods can be applied to forecast supply and demand processes more accurately and to improve the quality of inventory decisions.

Chapter 2 examines the prediction of procurement lead times.[1] In many compa-

[1] Chapter 2 is based on the paper by Robin Reiners, Christiane Haubitz, and Ulrich Thonemann that was published in *Production and Operations Management* (Reiners et al. 2025). Dr. Christiane Haubitz was mainly responsible for collecting the data, identifying the problem, and conducting the pre-study. The remaining work, including the literature review, programming, and revisions for *Production and Operations Management*, was carried out by Robin Reiners. Professor Ulrich Thonemann contributed through discussions on the study's design and the positioning of the paper. The paper benefited from comments of two anonymous referees and the editors of Production and Operations Management as well as from feedback at the ISIR Summer School 2023 at Cardiff University, United Kingdom, on August 30, 2023.

nies, planned lead times are stored as static entries in ERP systems. These values, however, often differ substantially from actual supplier lead times. Inaccurate lead times distort inventory control parameters and lead to either excess stock or inventory shortages. This chapter develops machine learning models that combine purchase order data with item and supplier characteristics to generate more accurate lead time predictions. Using real company data, the study demonstrates that these predictions substantially outperform ERP entries and classical benchmarks, and that improved lead time information translates into significant inventory savings without compromising service quality.

Chapter 3 addresses the estimation of demand during replenishment lead times. Inventory policies require not just point forecasts but an understanding of the full distribution of demand over the lead time, especially under the intermittent demand patterns typical of spare parts. Traditional parametric assumptions, such as normal or poisson distributions, are poorly suited to such demand patterns. To overcome this limitation, the chapter proposes a non-parametric conditional density estimation framework that provides part-specific predictive distributions of lead time demand. By generating richer representations of demand uncertainty, the approach enables more robust safety stock and base stock decisions and better aligns forecasting models with the realities of spare parts demand.

Chapter 4 builds on chapter 3 by testing the robustness of cross-learning approaches in a controlled simulation environment.[2] While Chapter 3 shows that pooling information across parts can improve distributional forecasts, it remains unclear how well these methods perform when no inherent similarities exist across SKUs. Chapter 4 addresses this question by systematically varying the degree of similarity and data availability across parts. The study evaluates how the machine

[2] Chapter 4 was presented at ISIR on August 14, 2025. It is joint work with Professor Florian Sachs and Professor Ulrich Thonemann. The work was mainly carried out by Robin Reiners. Professors Florian Sachs and Ulrich Thonemann contributed through discussions on the modeling approach, the design and analysis of the simulation study, and the positioning of the paper.

learning based method behaves under varying degrees of SKU similarity and compares its performance with single-SKU methods. The results highlight both the benefits and the limits of cross-learning: substantial gains are achieved when similarities exist, but performance converges to item-level methods when they do not. This analysis provides important insights for the practical application of cross-learning, clarifying when it should be relied upon and when other methods are more favorable.

Chapter 5 concludes the dissertation by summarizing the key insights, reflecting critically on the limitations of the studies, and outlining directions for future research in data-driven spare parts planning.

1.3 Contribution

This dissertation contributes to the literature on inventory management, demand forecasting, and supply chain analytics by showing how data-driven methods solve core spare-parts planning challenges and translates inventory research into practice. The contributions are threefold.

In **Chapter 2**, we contribute by showing how operational data can be leveraged to improve supply-side information. By applying machine learning to purchase order and master data, the work develops predictive models for procurement lead times that are more accurate than static ERP entries and classical time-series benchmarks. The link between improved lead time predictions and inventory outcomes is established empirically, showing tangible benefits for firms.

In **Chapter 3**, we advance the modeling of demand uncertainty in spare parts contexts. By moving from point forecasts to full predictive distributions, the Chapter provides a framework that captures the irregular and intermittent nature of spare parts demand. This approach improves the reliability of safety stock calculations and offers a practical approach that firms can implement with the data they already collect.

In **Chapter 4**, we clarify the role of cross-learning in data-scarce settings. While prior work has suggested the promise of pooling information across parts, this dissertation systematically evaluates its robustness. It shows that cross-learning delivers substantial improvements when item similarities are informative, but that its advantages fade when similarities are absent. These findings provide both methodological and managerial guidance, specifying the boundary conditions under which cross-learning should be applied.

Taken together, the three studies demonstrate that data-driven methods can improve the quality of information used in spare parts planning, that these improvements translate into better inventory performance, and that the conditions under which different methods are effective can be explicitly understood. The dissertation thus contributes both to academic research and to practice, offering firms concrete ways to improve service levels and reduce capital lock-up by making better use of the data they already have.

Chapter 2

Lead time prediction for inventory optimization with machine learning

Modern decision-support applications build on planning parameters such as lead time, price, yield, etc., which are maintained as master data. The accuracy of master data significantly influences the viability of such applications. However, the maintenance of master data is considered a tedious and error-prone task. In this study, we explore the effectiveness of machine learning techniques to improve the accuracy of plan lead times. We apply both unsupervised and supervised learning methods for creating lead time prediction models. We test our approach using historical data of a global equipment manufacturer. In a numerical analysis the calculated plan lead times are over 30% more accurate than current plan lead times in terms of mean-squared-error (MSE). This increased accuracy of plan lead times reduces inventory investment by approximately 7%.

2.1 Introduction

Supply chain management has become an integral factor in competitive strategy to enhance organizational productivity and profitability (Li et al. 2006). Considerable research and effort have been devoted to formulating optimization methods that require master data to achieve high quality results. Maintenance of master data is a time-consuming, tedious and error prone task since the data can be volatile, ambiguous, and influenced by factors outside the company's scope (Escudero et al. 1999).

Artificial intelligence (AI) technologies have gained prominence in their application to business-related problems and have been applied to pattern recognition, inference and learning from experience (Brynjolfsson and McAfee 2017). In supply chain management, AI applications have been developed and deployed to e.g., inventory control and planning, transportation network planning, and purchasing and supply management (Min 2010).

This study is motivated by a problem faced by a global equipment manufacturer which we refer to as "the company". The company has annual sales of several billion euros and operates world-wide. To provide maintenance and repair services for specialized equipment, the service department holds spare parts in stock. While some stock-keeping-units (SKU) are produced internally, many are purchased from external suppliers with different lead times. To coordinate inventory levels across multiple levels in their supply chain, the company employs optimization methods for determining inventory control parameters, that is re-order points, base-stock levels and order quantities. The calculation of these parameters relies on plan lead times that are retrieved from their enterprise resource planning (ERP) system. Due to inaccurate plan lead times of some SKUs, many inventory control parameters are sub-optimal, resulting in lower service levels or higher inventory than optimal.

In this paper, we address plan lead times and analyze how well their accuracy can be improved by machine learning. We propose a method to derive plan lead times with machine learning and compare the performance of three different machine learning regression algorithms: linear regression, random forests, and gradient boosting. We benchmark their performance with classical statistical approaches, that is single-exponential-smoothing (SES) and the historical average, as well as the current plan lead times from the company's ERP system.

Our results show that plan lead times estimated with machine learning predict lead times more accurately than currently employed plan lead times. The best performing machine learning model improves lead time accuracy in terms of MSE by over 30% compared to current plan lead times. We also analyze the accuracy of our method for SKUs with respect to their procurement frequency and we find that our approach offers significant gains even for newly sourced SKUs. For very frequently procured SKUs, our approach still offers substantial gains in accuracy compared to classical statistical measures, but the difference is smaller than for infrequently procured SKUs.

To quantify the expected business benefits from improved plan lead times, we conducted an inventory simulation on historical data. The results from the simulation show that inventory holding can be reduced by approximately 10% while maintaining the same service level. The remainder of this paper is structured as follows. In Section 2.2, we review the relevant literature on lead time prediction. In Section 2.3, we outline the problem and identify requirements for improving plan lead times with machine learning. In Section 2.4, we present our method of both supervised and unsupervised learning. In Section 2.5, we examine the economic implications of improved plan lead times and present numerical results. In Section 2.6, we derive practical implications, outline limitations and conclude.

2.2 Literature review

Lead times are an important parameter in inventory optimization, extensively discussed in literature (Muthuraman et al. 2015, Silver et al. 2016, Wang 2012). However, the literature has paid little attention to analyzing lead time accuracy and predicting lead times, especially in the context of inventory planning and optimization.

In practice, plan lead times are often derived from supply chain contracts, employee experience, or computed from historical data (Grout and Christy 1993, Lawrenson 1986, Urban 2009, Lingitz et al. 2018). In some papers, lead times are predicted but not for stock replenishment. Berlec et al. (2008) examine how to accurately predict product delivery times, a critical factor in effective contract negotiations. Their study focuses on identifying commitment-worthy lead times for customers, ensuring that contracts reflect achievable delivery schedules. Similarly, Duffie et al. (2017) and Yang and Geunes (2007) explore aspects of customer lead time within contractual contexts, they do not specifically address its prediction for inventory replenishment.

Some recent studies have made promising strides in the field of lead time prediction through analytics. While these explorations are noteworthy, they present opportunities for further refinement and validation within the peer-reviewed scholarly community. Banerjee et al. (2015) combine matrix gamma distribution and step-wise linear regression to predict lead times. Liu et al. (2018) and de Oliveira et al. (2021) employ regression models and compare various machine learning techniques, demonstrating high accuracy in predicting supplier lead times. However, these studies do not address the unique challenges in inventory management posed by spare parts, which are seldom re-ordered, leading to scarcity in historical data at the SKU level. These studies also neither address implementation of the approaches nor the financial implications of implementing such predictive models.

To address these gaps, our research leverages machine learning for lead time prediction. Insights from studies in related domains, such as transportation (Hofleitner et al. 2012, Choi et al. 2016, Barbour et al. 2018) and manufacturing (Lingitz et al. 2018, Burggräf et al. 2020, Bender and Ovtcharova 2021), indicate the effectiveness of ensemble decision tree models. Specifically, the studies by Oeztürk et al. (2006), Lingitz et al. (2018), Gyulai et al. (2018) and Bender and Ovtcharova (2021) have consistently demonstrated the superior performance of ensemble decision tree models such as random forest and gradient boosting in predicting various lead times.

While the existing literature offers insights into prediction methodologies, several open questions remain regarding how these predictions can be leveraged to derive accurate plan lead times for inventory control. Existing research focuses on predicting lead times at the individual purchase order level, whereas inventory control requires a lead time parameter to estimate the lead time demand distribution to compute base stock levels, safety stock or reorder points. This gap between order-level lead time predictions and the need for suitable parameters for inventory management is addressed in this paper.

Moreover, while the literature addresses how lead time variability impacts base stock levels and inventory investments, it is typically assumed that the lead time distribution is already known or can be estimated from historical data. This assumption is particularly problematic for spare parts, where data is often sparse, making it difficult to estimate moments from historical observations.

Building on existing research in lead time prediction, we extend the literature by examining how varying SKU order frequencies affect bias in predictive models. The inherent bias introduced by SKU order frequency discrepancies presents a key challenge in procurement operations. Frequently ordered SKUs generate extensive historical data, allowing for higher prediction accuracy in machine learning models. Conversely, infrequently ordered SKUs have sparse historical data, leading to greater uncertainty in lead time predictions. This imbalance reflects the operational nature of procurement systems, where demand varies widely among SKUs. We address this challenge through a comprehensive approach combining feature engineering techniques and systematic evaluation of balancing methods for regression tasks.

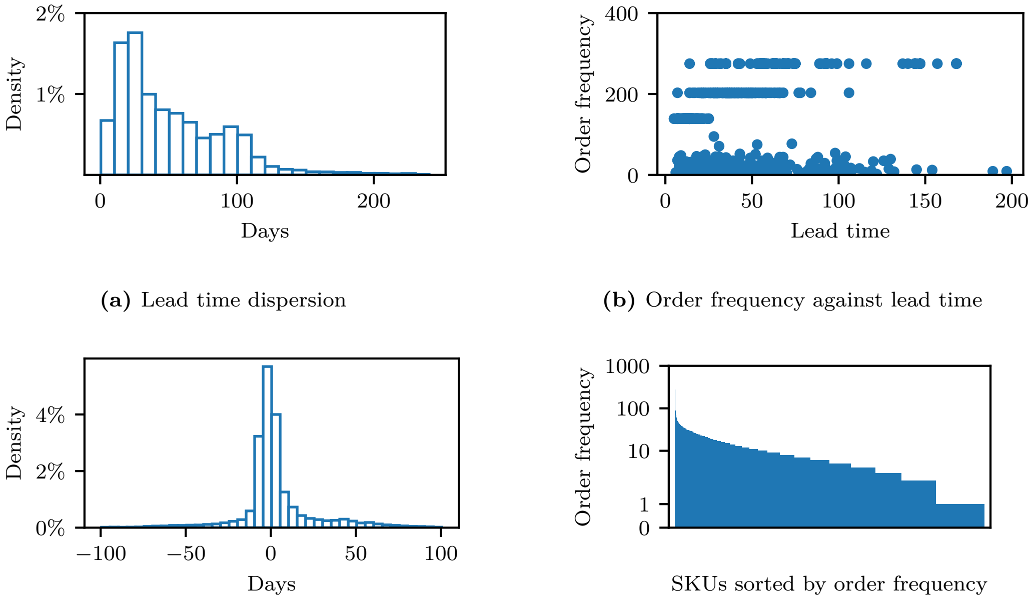
Building on these findings, our analysis focuses on three machine learning models: linear regression, random forest, and gradient boosting (Weisberg 2005, Breiman 2001, Friedman 2002). By applying these models to predict supplier lead times accurately, we aim to develop a framework that not only enhances inventory planning and optimization but also addresses the unique challenges posed by spare parts, thus contributing significantly to the existing body of knowledge in the field.

2.3 Problem description

To offer repair and maintenance services to its customers, the company operates a warehouse network for its spare parts. To efficiently manage inventories, the service division of the company regularly optimizes its inventory control parameters (such as re-order points and base-stock levels) under service-level constraints. Due to inaccurate lead time parameters, inventory levels have been sub-optimal.

When analyzing the historical lead times, we observe a high dispersion. Figure 2.1 provides some information on lead times at the company. This data and other data of the company had to be sanitized by the company's request, but the general insights and relative performance data that we report are not affected by the sanitation. Figure 2.1(a) shows the lead time distribution of all SKUs. We

Figure 2.1: Plan and actual lead time characteristics of all purchase orders



(a) Lead time dispersion

(b) Order frequency against lead time

(c) Plan lead time accuracy

(d) SKU order frequency

can see that the distribution is highly skewed with a long right tail. Figure 2.1(b) plots order frequencies against procurement lead times. While procurements with longer lead times generally belong to SKUs with lower order frequencies, no strong relationship exists between these variables. This is evidenced by a single SKU ordered approximately 280 times spanning lead times from 20 to 170 days. Figure 2.1(c) shows the accuracy of current plan lead times. We can see that the deviations from the planned lead times occur with similar magnitude in both directions. Figure 2.1(d) shows the order frequency of all SKUs. We find a highly skewed number of purchase orders per SKU. While a few SKUs are ordered at a high frequency, the majority of SKUs are ordered infrequently.

Plan lead times are maintained as master data. These plan lead times are static and are rarely updated. The vendor management and the inventory management teams manage and maintain the plan lead times on the basis of the contractually agreed lead times. For most SKUs, the plan lead time is agreed upon with the individual supplier. For the remaining SKUs, the plan lead times are generated by calculating the historical average lead times of the SKU. Considering how infrequently most SKUs are procured, estimates based solely on historical lead time data (interpolation) are limited by the available observations. Together with experts from the company, we discussed which data from their ERP might contain causal information about lead times. After this qualitative approach, we queried operational attributes from the ERP system. We worked with two different data sources: SKU master data and transactional purchase order data.

2.4 Methodology for accurate plan lead time prediction

Our methodology is structured into three fundamental stages. The first stage involves a rigorous data preparation process. We briefly touch our cleaning and merging process of the raw data sets, ensuring the integrity of the data. Subsequently, we conduct an insightful initial analysis to identify features with potential significance for our predictive models. The second stage encompasses feature engineering. This step is dedicated to the creation and transformation of features to improve the performance of the applied machine learning techniques. In the final stage we build upon the prepared data set and apply machine learning algorithms to refine our predictive models. We define robust evaluation techniques, serving as benchmarks for comparing and selecting the most effective models. Through regularization and hyper-parameter optimization, we tailor the models to predict plan lead times accurately.

2.4.1 Data preparation and exploratory analysis

We work with two different data sources: SKU master data and transactional purchase order data. SKU master data contain data such as the supplier from which the SKU is sourced, the country it is sourced from, and the current plan lead time. Transactional purchase data contain information on purchased SKUs over four years (2018 - 2022), including the request and delivery date and the plan lead time at the time when the SKU was ordered. Therefore, this data set holds information on how master data, such as plan lead time, has changed over time.

Most pre-processing of our data consisted of storing information as correct data types and identifying SKUs with missing entries. Since we could not fix missing values, these SKUs representing less then 2% of the total data set were dropped.

An analysis of categorical variables reveals that a large share of purchase orders is sourced from the country in which the global warehouse is located (Figure 2.11a). We also observe countries of origin with only few purchase orders and therefore we clustered countries from the same continent with less than 1% of the order volume. The remaining categorical variables seem plausible and do not require further processing.

While assessing the numerical features of our data sets, we observe some extreme outliers for the variable's valuation price and purchase order quantity. We decide to winsorize them based on Tukey's rule of thumb (Tukey 1992). This method is widely used in statistical analysis (Carling 2000, Wiley and Wiley 2019) and applied in machine learning to ensure models account for all data points while mitigating the impact of extremes, where models benefit from capturing the full range of data without being skewed by extremes. It identifies outlier values based on the inter-quartile range (IQR). Corresponding outliers are truncated and set to a constant value equal to the IQR times a factor, which, following Tukey's rule,

Figure 2.2: Lead time volatility over time

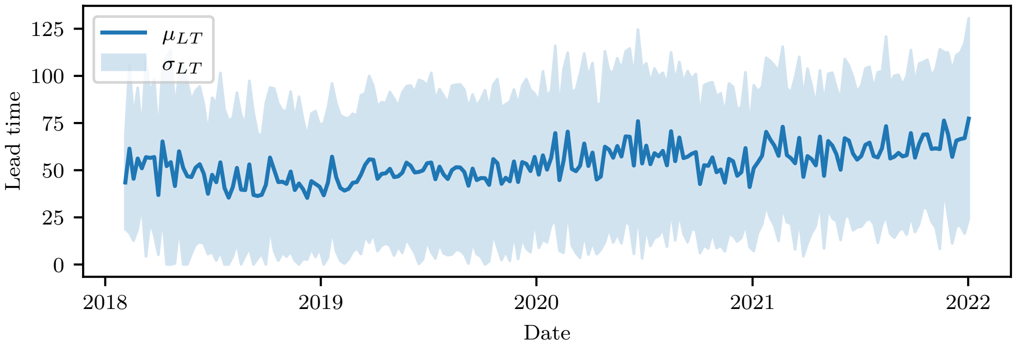


Table 2.1: Pre-processed features

Column	Data type	Training set statistics		Test set statistics	
Purchase order	Category	Unique: 144,059		Unique: 15,989	
SKU description	String	Unique: 16,090		Unique: 7,667	
SKU ID	Category	Unique: 16,090		Unique: 7,667	
Supplier ID	Category	Unique: 331		Unique: 246	
Business unit	Category	Unique: 12		Unique: 12	
Business line	Category	Unique: 13		Unique: 13	
Business type	Category	Unique: 2		Unique: 2	
Business area	Category	Unique: 5		Unique: 5	
Criticality	Category	Unique: 3		Unique: 3	
ABC / XYZ	Category	Unique: 9		Unique: 9	
Country code	Category	Unique: 11		Unique: 11	
Requested in time	Boolean	Y (27 %)	N (73 %)	Y (36 %)	N (66 %)
Repairable part	Boolean	Y (4 %)	N (96 %)	Y (4 %)	N (96 %)
Manual correction	Boolean	Y (2 %)	N (98 %)	Y (1 %)	N (99 %)
Long tail lead time	Boolean	Y (6 %)	N (94 %)	Y (8 %)	N (92 %)
Valuation price	Float	μ: 331.17	σ: 571.21	μ: 335.41	σ: 565.82
Order quantity	Integer	μ: 9.31	σ: 13.73	μ: 11.38	σ: 15.35
Current plan lead time	Integer	μ: 48.32	σ: 39.16	μ: 54.14	σ: 41.21
End of support (days)	Integer	μ: 3,527.19	σ: 571.21	μ: 3,546.50	σ: 535.16
Lead time	Integer	μ: 49.37	σ: 39.16	μ: 56.04	σ: 44.60

we set to 1.5. For linear regression, we normalize numerical data by re-scaling features to the range of zero to one. We do not normalize the numerical data for random forest and gradient boosting, as tree based models do not require feature scaling (Brownlee 2020).

Despite initial concerns about the potential impact of COVID-19 on lead-time volatility, our analysis did not uncover any significant fluctuations or patterns in the data, as illustrated in Figure 2.2. The lead times remain relatively stable over the observed period, with no extreme variances that would necessitate adjustments to the data set.

Our final data set includes 160,048 purchase orders across 17,728 SKUs, which are used for model training and testing (cf. Table 2.1). For performance evaluation, we implement a fixed origin evaluation procedure, splitting the data into a training set and a test set (Tashman 2000). The first 42 months of data comprise the training set, while the final six months, selected to coincide with the company's semi-annual review period, form the test set.

To validate that the data of the training set do not deviate heavily from the test set, we compare both sets in Table 2.1. Key statistics of all features for each set indicate that both share similar characteristics.

Although the overall lead time volatility remains relatively consistent over time, we observe an increase of 6.67 days in average lead times compared to the training set. This shift does not indicate a broader pattern but rather reflects normal operational variations. We explored various splitting points as suggested by Hyndman and Athanasopoulos (2018), but ultimately, we chose the above mentioned split to maintain sufficient observations. We exercise the train-test split at this point to ensure that no information from the test set is used for feature engineering.

2.4.2 Feature engineering

We next present feature engineering techniques to extract meaningful information from the existing data set and to transform this information so that machine learning models can process it.

In our data set, we have short descriptions of the SKUs. We anticipate that our models can learn from other SKUs with similar lead times. For example, electronic parts with a similar description of "8 GB DDR4 RAM" and "6 GB

DDR4 Memory" can have similar lead times.

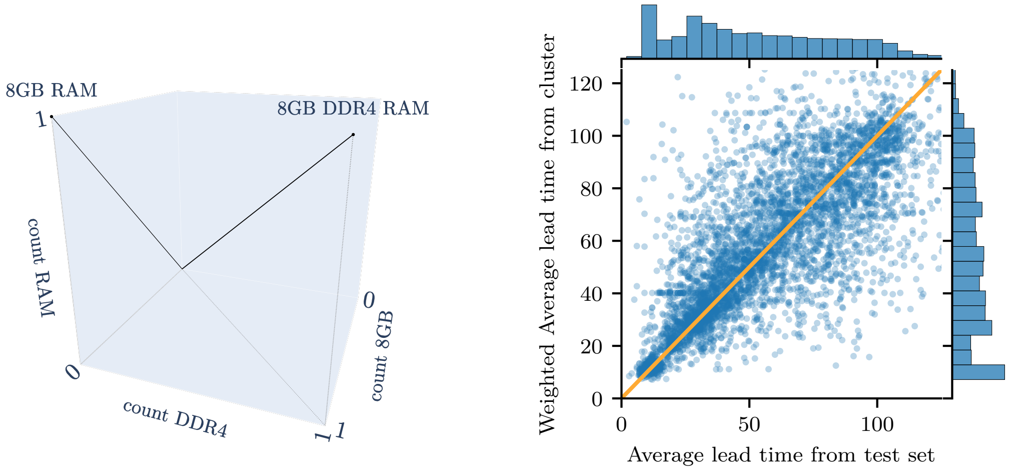
Strings cannot be interpreted by machine learning algorithms and we must convert them into ratio scale numeric representations that machine learning algorithms can process. We use KNN-regression to identify similar SKUs from the same supplier based on their description (unsupervised learning). For each SKU, we calculate a measure of lead time central tendency from its similar SKUs and use it as a feature for the machine learning algorithm.

Before starting with feature engineering, we have to pre-process some data. Because the SKU descriptions are not very extensive, most data cleaning, like removing stop words or expanding contractions, is redundant (Webster and Kit 1992). We merely remove extra white spaces, lowercase all characters, and split the descriptions into single words (tokenization).

Because the SKU descriptions are concise, we use the bag of words count vectorizer, a vector space representation model for unstructured text (Zhang et al. 2010). From a total of N SKU descriptions, each SKU description n can be represented as a set of unique words (tokens) S_n. The corpus is the union from these sets $\cup_{n=1}^{N} S_n$. It includes every unique word from each SKU description and a single part description can be represented as a $(1 \times \cup_{n=1}^{N} S_n)$ vector, where each column corresponds to a specific word from the corpus. The value of each column is the frequency of how often that word appears in the description. Table 2.2 illustrates the concept.

We cluster SKUs which are sourced from the same supplier based on their cosine similarity. To evaluate if these clusters are meaningful to predict lead times, we analyze if SKUs from the same group have similar lead times. The part descriptions of the SKUs shown in Table 2.2, for example, have a cosine similarity of 0.83. For each SKU, we calculate its average lead time and compare it to the average of all historical lead times from the SKUs in the same cluster (cf. Figure

Figure 2.3: Creation of similarity feature



(a) 3D vector space cosine similarity

(b) Average lead time correlation between SKU and cluster

Table 2.2: Example count vectorization

		Word counts					
SKU	Description	8	16	GB	DDR4	RAM	Memory
00001	8 GB DDR4 RAM	1	0	1	1	1	0
00002	16 GB DDR4 Memory	0	1	1	1	0	1

2.3(b)). We see that the average lead time of an SKU correlates with the average lead time of the SKUs from the same group. Therefore, we add the average lead time from the clustered SKUs as an additional feature. For SKUs which are not in any cluster, we take their average historical lead time. For those SKUs that have not been procured before we take the average lead times from the corresponding supplier.

As shown in Figure 2.1(a), the distribution of the target variable (lead time) is skewed, indicating an imbalance with a long tail extending towards longer lead times. To address this, we explored various balancing techniques such as random over- and under-sampling, as well as synthetic minority oversampling for regression, which have been proposed for handling imbalanced regression tasks (Branco et al. 2017, Torgo et al. 2013). In our analysis, these sampling methods did not improve results, probably because such techniques are more suited for contexts where the under-represented numeric value interval is most critical, and a wrongful prediction can be costly, often at the expense of overall performance (Torgo et al. 2013).

We conducted a point-biserial correlation analysis, finding a weak but significant negative correlation $r = -0.109, p \leq 0.001$ between lead times and punctuality (defined as a deviation of ±3 days from the contractually agreed lead time). To improve the machine learning algorithm's ability to detect this correlation, we include a Boolean indicator for long lead times in the training set. We define long lead times as those in the 95th percentile, corresponding to a value of 119 days. This threshold represents the minority class and aligns with the typical range for rare events reported in the literature (Torgo et al. 2013).

As an additional feature, we compute a measure of supplier performance according to the business rules applied at the company. At the company, the supplier performance is measured as the share of orders delivered in time compared to all orders placed at the supplier. We first compute this measure for each supplier in the training set. Afterwards, we append this measure to the corresponding supplier in the test set. Some suppliers included in the test set do not appear in the training set. For those, we take the average supplier performance across all suppliers in the training set to complete the records.

As in most machine learning applications, the target variable is influenced by categorical (nominal scale) features. One-hot encoding, the most widely used coding scheme (Rodríguez et al. 2018), but in our case leads to undesirable sparsity in the data, especially with numerous categories such as the supplier ID (Gupta and Asha 2020, Prokhorenkova et al. 2017). To address this, we implement multiple Bayesian encoding techniques: target encoder, polynomial encoder, helmert encoder, james-stein encoder, m-estimate encoder, weight of evidence encoder and catboost encoder – for a full reference, please refer to Pedregosa et al. (2011) and Prokhorenkova et al. (2017). We choose the one which performs best in cross-validation, an approach that we cover in the next Section.

2.4.3 Model optimization and evaluation

Informed by our literature review, we train three established machine learning algorithms to predict lead times: linear regression, random forest, and gradient boosting. Linear regression serves as our baseline, leveraging its capacity for modeling relationships between scalar responses and explanatory variables through a least squares approach (Weisberg 2005). The random forest algorithm, aggregates predictions from a multitude of decision trees trained in parallel on random data subsets to enhance predictive accuracy and mitigate over-fitting (Breiman 2001). Gradient boosting constructs an additive model in a forward stage-wise fashion, and is particularly adept at addressing the residuals of preceding trees, sharpening accuracy on more complex or noisy data sets (Friedman 2002).

Our primary performance metric is the MSE as the performance of an inventory control system heavily depends on achieving a low MSE in lead time demand forecasts (Syntetos et al. 2009). The MSE penalizes large deviations from observations disproportionately compared to minor deviations.

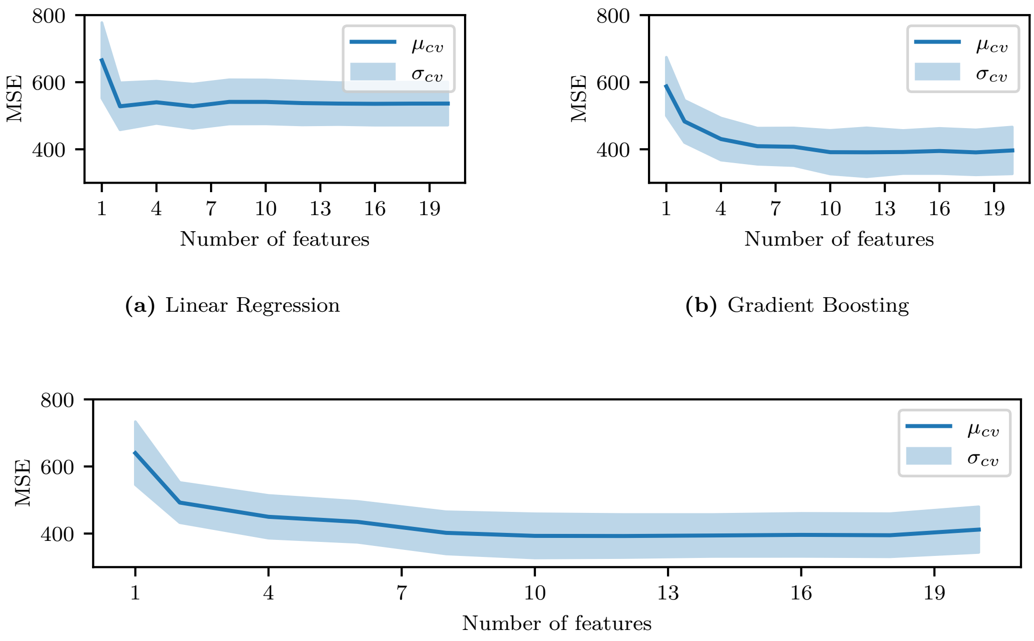
However, we recognize that the MSE can be challenging to interpret due to its dependence on the scale of lead time, especially when assessments are conducted across various SKUs. To address this, we also report the mean-absolute-percentage-error (MAPE), a scale-free metric, which prevents high or low performance in terms of MSE from being disproportionately influenced by a few SKUs. In addition to MSE and MAPE, we also evaluate the bias (mean forecast error). Evaluating bias is crucial because even with high accuracy, a model that consistently over- or underestimates can lead to poor inventory performance.

As discussed in Section 2.3, the procurement frequency of SKUs in our dataset is highly skewed, potentially leading to bias for infrequently ordered items. To address this, we explored balancing techniques such as minority over-sampling (OS) and majority under-sampling (US), as well as sample weighting based on the inverse of order frequency. These methods aimed to ensure a more balanced representation of SKUs and improve the model's ability to generalize across both frequently and infrequently ordered items. However, our results indicated that these techniques did not consistently improve model performance or significantly reduce bias. Therefore, we do not use these methods in the main analysis. Detailed results from the experiments are provided in Table EC.2 of the e-companion for reference and transparency.

To determine the effectiveness of these techniques, as well as to optimize hyper-parameters and perform feature elimination, we calibrate our models using a cross-validation procedure based on a rolling forecasting origin, applied to the training data (Hyndman 2014). This technique takes into account the temporal nature of our data. The process involves dividing the training data set into several subsets, or 'folds,' considering the sequential order of the data. During each iteration, one fold is reserved for validation, while the remaining folds are used for training the model. This sequential approach ensures that the model is trained on past data and tested on future data. For our analysis, we implement the commonly used $k = 5$ folds.

In the initial configuration, we evaluate a base machine learning model, utilizing all available features and default hyper-parameters. This configuration serves as a benchmark for assessing model performance. To improve the model, we then perform hyper-parameter tuning through a grid search in a cross-validation framework, optimizing the model's performance with the full feature set (Kuhn

Figure 2.4: Cross-validation results of reverse-feature elimination with hyper-parameter tuning process



(a) Linear Regression

(b) Gradient Boosting

(c) Random Forest

and Johnson 2019).

For regularization, we use the recursive feature elimination method (Athey and Imbens 2019). In an iterative process, the models are trained on the full feature set. In each iteration, the one feature with the lowest importance score is eliminated. This step is repeated until all but one features have been removed. In order to find appropriate hyper-parameter values we use a grid search approach after every feature elimination round. The grid design includes 100 parameter combinations (Santner et al. 2003). For our final model we choose the feature sub-set and hyper-parameter configuration that minimizes the MSE averaged over the five folds (cf. Figure 2.4). A summary for each algorithm of the included features, as well as hyper-parameters and their optimal values can be found in Table 2.3 of the appendix.

To estimate plan lead times, we need to derive a representative purchase order from historical purchase orders for each SKU. We pass the representative purchase order's feature set into our trained machine learning model and use the response value as new plan lead times. The approach is rather trivial: To calculate a representative purchase based on information from the train-set, we take averages for the numerical values. For categorical values we take the most frequent value.

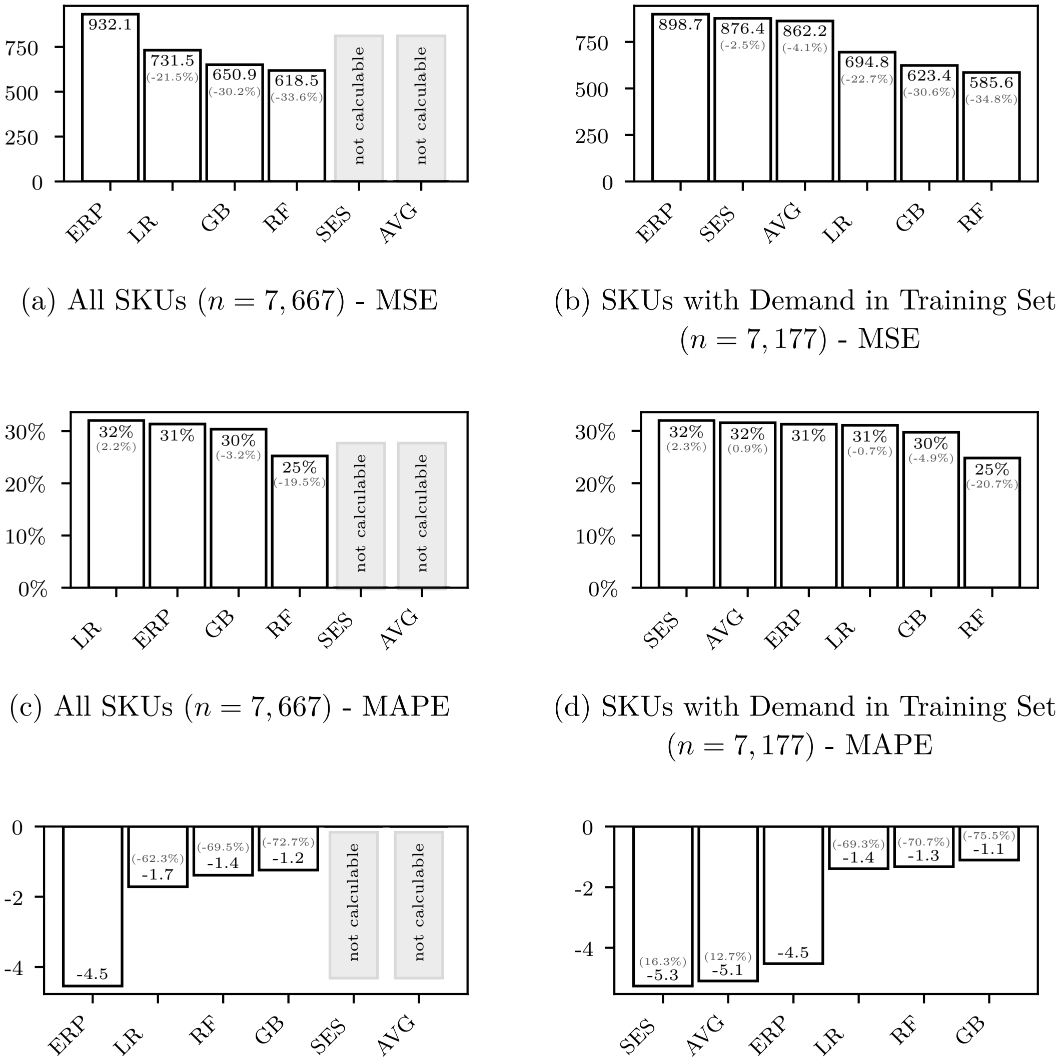
2.5 Numerical results

2.5.1 Prediction accuracy

We analyze the accuracy of our models in predicting purchase order lead times for our test set. 490 SKUs were newly introduced, for which traditional forecasting techniques like SES can not be applied due to the absence of historical data. Figure 2.5 illustrates the prediction errors for different lead time forecasting methods. While the top graphs show the MSE, the graphs in the middle show the MAPE, and finally the bottom graphs show the bias. The metrics for the three graphs on the left were calculated for all 7,667 SKUs and the metrics for the graphs on the right were calculated for the subset of 7,177 SKUs for which historical data exists.

To analyze our numerical results, we first focus on MSE for SKUs with demand in the training set, as shown in Figure 2.5(b). The data indicates that algorithms, in general, provide more accurate lead time predictions than those from the ERP system. In bench-marking classical methods against regression methods, the latter demonstrate improved MSE outcomes compared to methods such as SES and historical averages. SES and historical averages perform similarly. The inclusion of additional features in regression models appears to contribute to their enhanced predictive performance. Within the category of regression models, it is the machine learning models, gradient boosting and random forest, that further

Figure 2.5: Accuracy of different potential plan lead times



(a) All SKUs ($n = 7,667$) - MSE

(b) SKUs with Demand in Training Set ($n = 7,177$) - MSE

(c) All SKUs ($n = 7,667$) - MAPE

(d) SKUs with Demand in Training Set ($n = 7,177$) - MAPE

(e) All SKUs ($n = 7,667$) - Bias

(f) SKUs with Demand in Training Set ($n = 7,177$) - Bias

enhance the predictive accuracy. Within this subset, random forest stands out, improving MSE by 34.84% over current ERP plan lead times, marking a significant improvement in forecast accuracy.

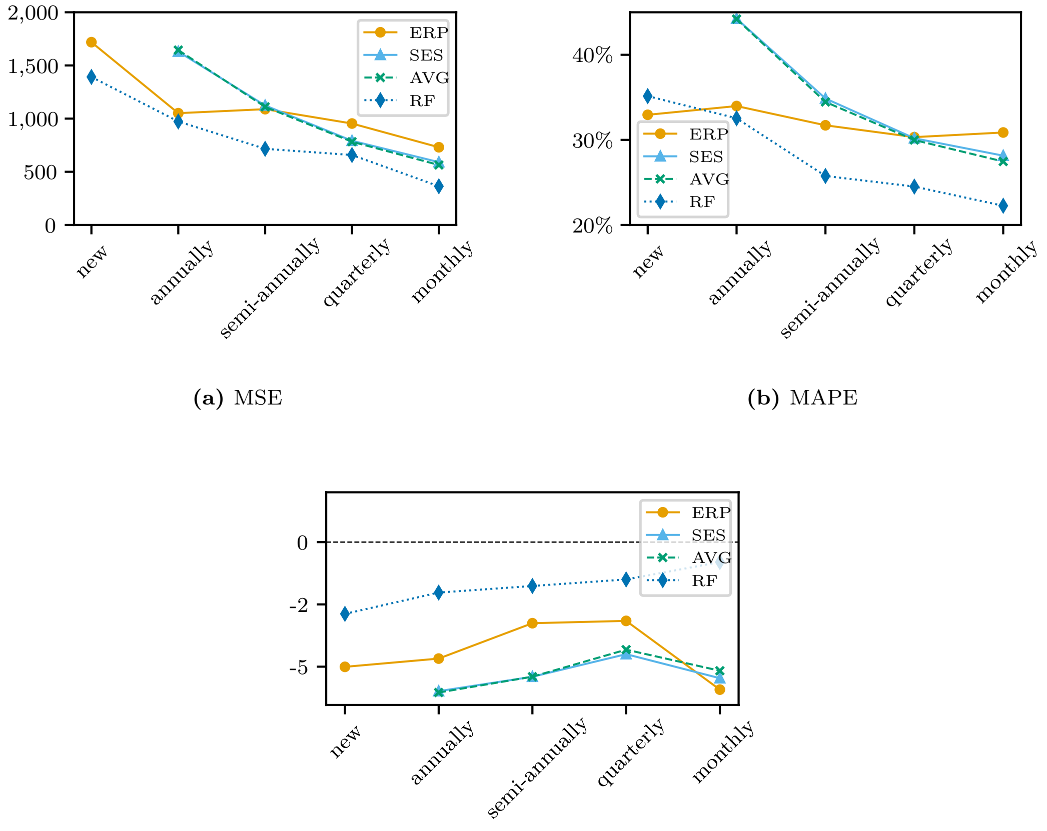
After confirming the effectiveness of the random forest model in terms of MSE for SKUs with demand in the training set, we examine the other graphs for consistent patterns. Upon extending the analysis to all SKUs as shown in Figure 2.5(a), consistency in algorithmic performance is observed even when evaluating SKUs without demand in the training set, with the random forest maintaining its lead in predictive accuracy. A similar improvement is observed in Figures 2.5(c) and 2.5(d). The random forest model not only lowers MSE but also reduces MAPE by 2.62 and 3.11 percentage points, respectively, which corresponds to improvements of 8.36% and 9.94%. These reductions underscore the model's increased predictive accuracy, confirming that the improvements are not disproportionately affected by outliers.

Continuing the analysis, we observe that both historical averages and SES exhibit a notable negative bias, as shown in Figure 2.5(f), aligning with expectations given the approximate 6-day shift in average lead times between the training and test sets (cf. Section 2.4.1). These methods inherently lack the capability to adapt to future developments. Consequently, their performance during inference is suboptimal. In contrast, current plan lead times (ERP) incorporate some level of foresight by reflecting temporal adjustments made by planners or adjustments derived from contractual obligations. This foresight allows for the anticipation of potential shifts. However, these plan lead times systematically underestimate actual lead times, as shown in Figures 2.5(e) and 2.5(f), indicating a persistent negative bias.

Regression models, particularly machine learning approaches like gradient boosting and random forest, are designed to identify and adjust for underlying data patterns. This capability enables them to effectively "debias" their predictions, as evidenced by the minimal bias observed even amidst slight shifts between training and test data. The high accuracy and low bias exhibited by these models demonstrate their robustness and their ability to generalize effectively to new data.

By evaluating model performance in relation to the order frequency in the training set, we gain insights into the reliability of our predictive methods for SKUs with varying levels of historical data. Figure 2.6 presents the performance of

Figure 2.6: Accuracy of different potential plan lead times relative to order frequency



(a) MSE

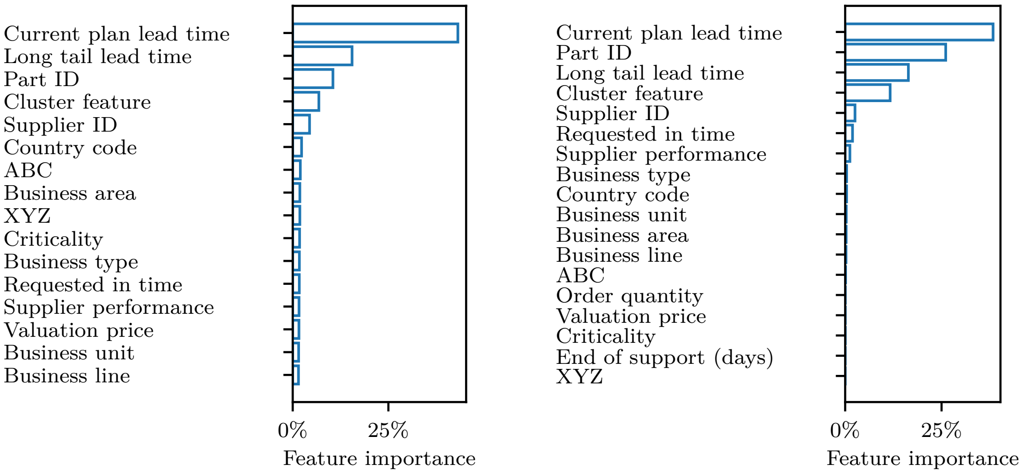
(b) MAPE

(c) Bias

different forecasting methods against the procurement frequency of SKUs from the training set. The frequency categories range from " new ", indicating no historical procurement data in the train-set, to "monthly", indicating frequent procurement. The graph plots the performance of our best performing machine learning model (random forest), current plan lead times (ERP), as well as conventional statistical methods (SES and historical averages). It illustrates a trend towards improved performance as the procurement frequency increases, signifying better accuracy with more data.

Notably, the random forest model consistently outperforms both the current plan lead times and classical statistical measures across all performance measures and almost all frequency categories. For instance, for new SKUs our method delivers an

Figure 2.7: Feature importance



(a) Random forest

(b) Gradient boosting

MSE roughly 25% lower than that of the current plan lead times. This significant reduction in error is indicative of the robustness of our method, particularly for SKUs without prior order history.

To generate some insights into how the features contribute to the prediction, we show the feature importance for random forest and gradient boosting in Figure 2.7. It becomes evident that the current plan lead time is a dominant feature, suggesting that the models heavily utilize this information to refine their predictions.

However, the significance of the “Long tail lead time”, “Cluster feature” and the advanced encoding of “Part ID” and “Supplier ID” emphasize the critical role of feature engineering in enhancing model performance. It indicates that the models are not solely reliant on current lead time parameters and historical lead times but also on the nuanced interplay of SKU characteristics. These features capture underlying patterns and similarities across SKUs that are otherwise not directly observable, thus providing powerful predictors that complement and enhance the historical data. Their substantial role in both models underscores the effectiveness of our feature engineering process, greatly enhancing the model’s ability to forecast lead times accurately.

In conclusion, the findings suggest that machine learning regression models, particularly the random forest, offer substantial improvements over traditional statistical methods and current ERP planning estimates in predicting lead times.

2.5.2 Inventory performance

To quantify the expected business benefits of increased plan lead time accuracy we conduct a simulation study. We evaluate how the more accurate plan lead times also improve the company's inventory performance. In our simulation, we compare how inventory levels evolve, using (1) the current plan lead times, and (2) the improved plan lead times. We use actual historical lead times for our benchmark. Our purchase orders hold information on when replenishment orders were submitted and when they arrived. We can simulate a realistic inventory development on an SKU level by modelling a periodic review base-stock policy with base-stock level S (Silver et al. 2016). This inventory control policy is applied in many real-world spare parts inventory systems (Cavalieri et al. 2008, Boylan and Syntetos 2010, Syntetos et al. 2012, Wang 2012) and also used by the company.

The intraperiod timing assumptions for the base-stock policy are as follows: Each SKU is reviewed daily, and if the inventory position is below the base-stock level S, a corresponding replenishment order is submitted to the supplier. After submitting the replenishment order, previous replenishment orders may arrive with lead times according to historical data. Subsequently, potential back-orders are fulfilled. Then demands d_t arrive and are either fulfilled or back-ordered. We assume that lead times and demands are independent from each other.

To determine the appropriate base-stock level S that meets the target β service level, the model must account for uncertainties in both demand and lead time.

Traditionally, this uncertainty is addressed by modeling the shortfall, which includes variability in order arrival sequences. In the presence of stochastic lead times, order crossover can occur, where orders are received in a different sequence than they were placed. This may result in higher inventory levels when earlier orders arrive. As reported by Robinson et al. (2001) neglecting order crossover under order-up-to inventory policies can lead to significantly higher inventory costs. However, we do not consider the effect of order crossover due to its complexity, its limited relevance to our primary objectives, and its low occurrence rate in our data set (0.18% of all purchase orders). Most SKUs are sourced from a single supplier with long intervals between orders relative to the variability in lead time.

Therefore, we use the lead time demand distribution to calculate the base-stock level S^*. Lead time demand describes the cumulative demand in the risk period with the duration of the lead time plus the review interval $LTD = \sum_{t=1}^{L+1} d_t$. Employing a base-stock policy, our objective is to ensure that a certain percentage of lead time demand is satisfied from stock (β service level). To determine the appropriate base-stock level S a probability function of lead time demand exceeding S is required. xx For fast-moving items, the assumption of normally distributed lead time demand is typically reasonable. However, our SKUs often exhibit intermittent demand patterns that the normal distribution may not represent. This is almost invariably the case for spare parts (Turrini and Meissner 2019). For slow-moving items the poisson distribution typically offers a good fit for the arrival of spare part demand (Silver et al. 1998). For demands that are not unit-sized but of constant size the resulting distribution of demand per period can be modelled by a "package poisson" distribution (Ritchie and Kingsman 1985). If demand size is not constant, it is reasonable to assume that demand arrivals are poisson distributed, and the order size follows a logarithmic distribution. In such cases, total demand is negative binomial distributed over lead time (Quenouille 1949). For our simulation, we use negative binomial distributions to model our spare part demand, noting that the poisson distribution is a special case of this distribution.

To compute the moments of the distributions, it is necessary to forecast both lead time and demand. We utilize the predicted lead times to estimate the average lead time, denoted as μ_L. To quantify lead time uncertainty, we employ the root-mean-squared-error (RMSE) as an estimator for the standard deviation of lead time, denoted as σ_L, following the methodology suggested by Barrow (2016). Expected demand, μ_D, and demand variance, σ_D^2, are derived from the purchase order data within the training dataset.

The moments of the lead time demand distribution are calculated using established formulas for the mean and variance of the sum of a random number of random variables $\mu_{LTD} = \mu_D \cdot \mu_{LT}$ and $\sigma_{LTD}^2 = \mu_{LT}\sigma_D^2 + \mu_D^2\sigma_{LT}^2$ (Zipkin 2000, p. 285).

To determine the appropriate base-stock level, we aim to ensure that the expected back-orders during a replenishment cycle do not exceed a specified fraction of demand within the same cycle. This objective is formalized through the following optimization problem (Sieke et al. 2012):

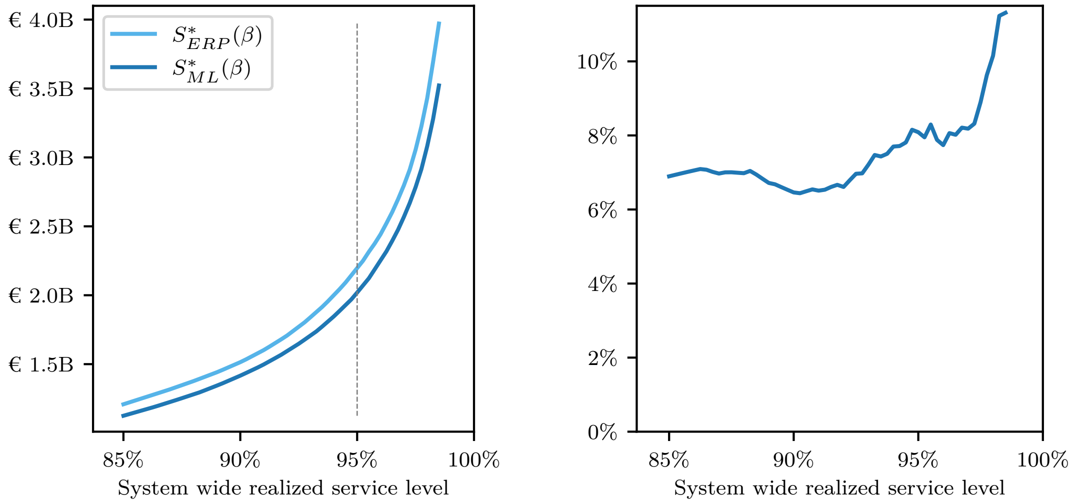
$$\begin{aligned} &\min S \\ \text{s.t.} \quad &\left[\sum_{y=S+1}^{\infty} (y-S) \cdot f_{LT+R}(y) - \sum_{y=S+1}^{\infty} (y-S) \cdot f_{LT}(y)\right] \leq (1-\beta)\mu_R \end{aligned} \tag{2.1}$$

Here, f_{LT+R} and f_{LT} are the demand distributions over lead time plus review period, and lead time, respectively.

We use S_{ML}^* to refer to the base-stock level based on the plan lead time derived from our proposed machine learning method, and S_{ERP}^* to refer to the base-stock level based on the current plan lead time.

For our simulation, the initial inventory position starts at the respective base-stock level S_{ML}^* or S_{ERP}^*. Assessing inventory performance right away might be

Figure 2.8: Simulation results



(a) Capital lockup for a given realized service level

(b) Capital savings for a given realized service level

unrepresentative, as the system has not had time to settle into its steady-state. Therefore, we do not measure inventory performance for the duration of the training set but only evaluate on the test set.

The amount of capital lockup has been scaled linearly in order to not reveal information about the company. Figure 2.8 summarizes our simulation results. The trade-off curve shows the capital lockup for a given realized service level. The grey curve shows inventory performance based on base-stock levels S^*_{ERP} (β), while the black curve describes inventory performance based on base-stock levels S^*_{ML} (β).

The black curve is constantly below the grey curve, implying that compared to the current plan lean times, our approach leads to stock levels that achieve higher service levels while having the same capital lockup. To derive some financial implications of our proposed method, we investigate inventory performance for an achieved 95% service level. We find that to attain the targeted service level, base-stock levels S^*_{ML} require roughly 7% less capital than base-stock levels S^*_{ERP}.

2.6 Discussion

2.6.1 Implications

The implications that we derive from our results provide recommendations for implementing lead time prediction approaches with machine learning and how they can be used in practice to exploit potential benefits.

Effective feature engineering can provide significant benefits. In our spare parts setting, data on SKU level is often limited. We successfully develop a text classification algorithm to incorporate information from similar SKUs based on their descriptions. Additionally, we enrich our data set with general information on supplier performance. State-of-the-art algorithms allow us to train our models on sparse categorical features. While conducting our study, we have found that there is a relationship between the extent of information extraction from the available data and the accuracy of our models. The feature importance analysis in Section 2.4.2 has clearly indicated that the additional features derived from feature engineering significantly contribute to accuracy improvements for all machine learning models we investigate.

Machine learning enables a more accurate prediction of lead time solely based on observed purchase orders. There are very few studies attempting to predict lead times solely based on the limited information available at a company. Companies often depend on (1) lead times specified in supply contracts, (2) employees' experience, or (3) average values based on historical data. We have seen that (1) lead times specified in supply contracts can significantly deviate from actual lead times. (2) Relying on expert knowledge can result in predictions being biased by stock-out aversion. Additionally, by relying on human experience, the company is exposed to the risk of unique human knowledge leaving the company in employee churn. (3) Average values based on historical data have been shown to be limited by the frequency of observations.

Our results align well with current evidence from an emerging stream of literature addressing the viability of machine learning applications for lead time prediction. We contribute to this literature stream by confirming that machine learning applications can reliably predict lead times with high accuracy. This is an opportunity for many companies to predict lead times operationally and take short-term measures. Indications as to whether an order arrives earlier or later than expected can be used to manage critical inventories tightly.

Plan lead times derived from machine learning are particularly valuable for procurement structures with large number of different SKUs with only a few purchase orders each. We have shown that compared to other alternatives, our method performs best for SKUs that are more frequently procured than once every half a year. This demonstrates a major advantage of our machine learning approach to determine plan lead times compared to methods that rely solely on past data: the data for available SKUs can be exploited to predict lead times for SKUs with an insufficiently large set of historical data. With increasing procurement frequency the accuracy of other methods converge, due to many observations being available from which estimates can be derived.

Plan lead time accuracy improves inventory performance. When lead time is used in the planning process to set inventory control parameters, increased lead time accuracy is expected to improve business outcomes. The business benefits of increased plan lead time accuracy are challenging to quantify since they are based on adjusted processes to reflect the improved accuracy. Therefore, we simulated inventory developments based on current and improved plan lead times. We estimate that for achieving a targeted service level of 95%, our approach is expected to reduce capital lockup by 7%.

While the findings in our study are promising, it is important to discuss the generalizability of these results. Our primary goal is not to identify a universally superior model but to demonstrate the benefits of applying machine learning techniques to lead time prediction in contexts characterized by sparse data and lead time uncertainty, such as spare parts management. A distinct and important contribution of our paper, compared to the existing literature, is our focus on predicting the lead time distribution from the supplier's perspective. Suppliers, who order from various manufacturers, often have limited information about the production state of products and manage a wide variety of product types. Our methodology addresses this complexity by using machine learning to effectively handle diverse and incomplete data, enabling more accurate predictions of lead times.

The higher accuracy of the random forest model in our study can be attributed to its robustness against skewed feature distributions and its ability to handle complex, non-linear relationships. However, we recognize that these findings may not directly apply to all contexts. Supply chain environments vary significantly, with differences in data availability, supplier behavior, and SKU characteristics. Consequently, no single model is likely to consistently outperform others across all scenarios. Therefore, while random forest demonstrates high effectiveness in our setting, we do not claim it will always be the best-performing model.

Our findings emphasize the critical importance of robust feature engineering and the incorporation of domain-specific knowledge. These elements are likely to be more decisive in the success of any machine learning model for lead time prediction than the specific choice of algorithm. The general methodology we use is likely to yield good results in similar contexts where suppliers face uncertainty due to diverse products and limited visibility into production processes. This adaptability underscores the broader applicability of our approach beyond our specific study.

Our study offers a scalable and solid methodology that can be adapted by researchers and practitioners alike to address the challenge of lead time prediction in various contexts, particularly those involving spare parts with limited historical data. As more companies develop in-house data science capabilities, implementing these advanced techniques becomes increasingly feasible and less resource-intensive. Although implementing data collection processes and operationalizing machine learning pipelines may initially seem complex and costly compared to simpler methods like SES, the significant accuracy improvements demonstrated in this study make a strong case for the investment.

We have successfully implemented this machine learning tool within a company, demonstrating its practical applicability and tangible benefits. The long-term cost savings from more accurate lead time predictions, as evidenced by our simulation, are likely to outweigh the initial implementation costs. The strategic advantage gained through improved inventory management and procurement processes justifies the complexity of the machine learning methods used. This approach provides a robust framework for companies to enhance operational efficiency and make informed decisions, thereby transforming how lead time predictions are handled in practice.

2.6.2 Limitations

In this study, we derived accurate plan lead times for SKUs. These plan lead times are point estimates of lead time which are used by optimization tools for inventory planning to set inventory control parameters. While many firms use inventory management software ignoring lead time uncertainty (Dolgui et al. 2013), lead time fluctuations strongly degrade the tools' performance and cause high inventory costs. Chopra et al. (2004), Song et al. (2010), Babai et al. (2022), as well as others, have demonstrated the economic implications of lead time uncertainty.

They have shown the importance of accounting for its effects regardless of the procedures used to compensate for demand uncertainty.

In addition to our proposed machine learning approach, it's important to acknowledge that alternative or complementary strategies, such as continuous improvement methodologies, could be employed to improve plan lead time accuracy. Continuous improvement could involve identifying and focusing on the most critical or problematic parts and suppliers to either better predict lead times or create robust processes that can respond to the uncertainty. By integrating such continuous improvement efforts with machine learning models, companies can potentially enhance their overall planning accuracy and operational resilience.

In this study, we also illustrate the impact that ill-estimated lead times can have on inventory performance. Yet, we only cover the forecasting of point estimates of lead time. Modern optimization tools account for lead time uncertainty and depend on estimates on the lead time uncertainty, i.e., the variance of the response value (Cachon and Terwiesch 2008, Song et al. 2010, Silver et al. 2016).

In the future it would be great if the methods we describe in our paper, were used for non-parametric conditional density estimation (Bertsimas and Kallus 2020, Dalmasso et al. 2020, Pospisil and Lee 2018). The conditional density estimation could be used to estimate lead time demand directly fitting into recent research of Boylan and Babai (2022) and Babai et al. (2022). Their research is dedicated to estimating lead time demand distributions directly, albeit considering only univariate lead time samples.

2.6.3 Conclusion

In this study, we addressed the challenge of maintaining inaccurate master data on which many inventory optimization models rely. We investigated with special regard to plan lead times how well the maintenance of this planning parameter can be automated by means of machine learning.

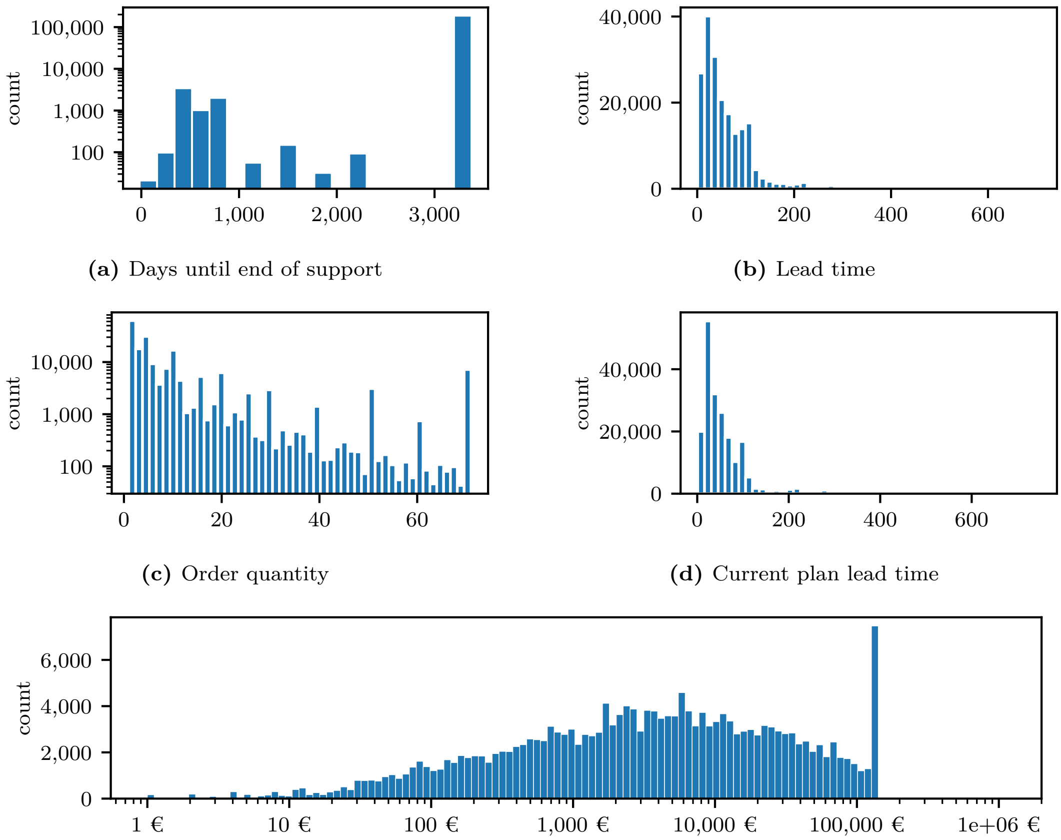
We applied unsupervised and supervised machine learning in our lead time prediction approach. Based on natural language processing techniques, we identified similar SKUs and used this information to generate features for our model. To derive plan lead times from our machine learning models, we used the features of representative purchase orders to generate a response value from our model. This value can be used as plan lead time.

We tested our approach on historical data of a global equipment manufacturer. We bench-marked the results against current plan lead times from the company as well as simple statistical measures of central tendency. We found that our approach significantly increases the accuracy of plan lead times by over 30% compared to current plan lead times. Simulations have shown that the increased accuracy of plan lead times reduces capital lockup by approximately 10% while increasing the likelihood of achieving the targeted service level for a given capital lockup. This translates into significant cost reduction for inventory-keeping as well as into reputational benefits due to fewer back-orders. Moreover, we developed a method to predict lead times for new SKUs, which can be used as decision support for contract negotiations with the suppliers. With specific regard to plan lead times, machine learning has proven to be a viable technique for improving the quality of master data. Thus our method creates tangible and intangible added value by automating a notoriously tedious and error-prone task.

Appendix of Chapter 2

2.A Univariate analysis of available features

Figure 2.9: Numerical data



(a) Days until end of support

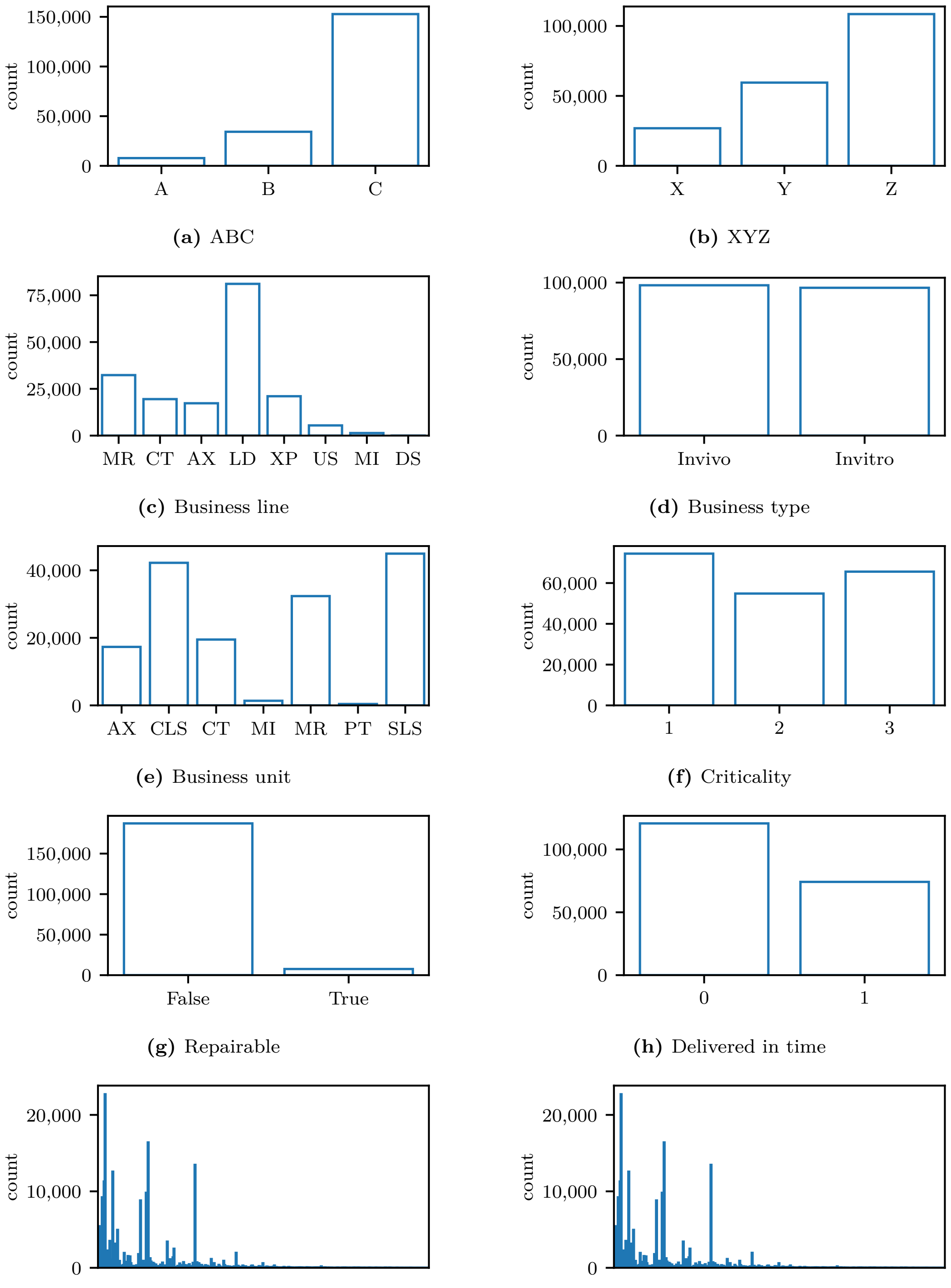
(b) Lead time

(c) Order quantity

(d) Current plan lead time

(e) Valuation price

Figure 2.10: Categorical data



(a) ABC

(b) XYZ

(c) Business line

(d) Business type

(e) Business unit

(f) Criticality

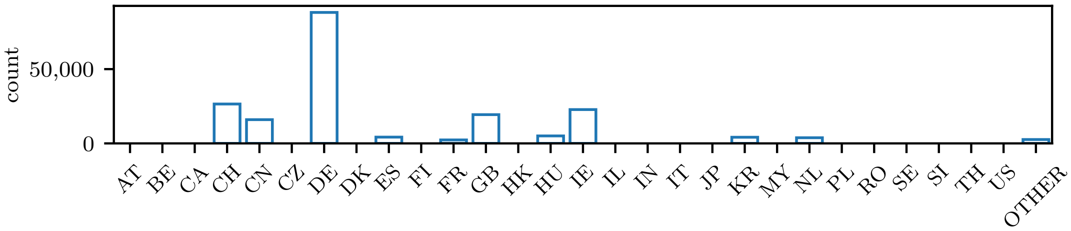
(g) Repairable

(h) Delivered in time

(i) Supplier

(j) SKU

Figure 2.11: Categorical data (Part 2)



(a) Country code

2.B Summary of tuned models

Table 2.3: Model hyper-parameters and selected features

Model	Hyper-parameters			Selected features
Gradient boosting	n_estimators: 100 max_depth: 3 max_features: None	max_leaf_nodes: None min_samples_leaf: 1 min_samples_split: 2	sub_sample: 1 learning_rate: 0.1	18
Random forest	n_estimators: 100 max_depth: None max_features: 1	max_leaf_nodes: None min_samples_leaf: 1 min_samples_split: 2	bootstrap: True cpp_alpha: 0 max_samples: None	16
Linear regression	fit_intercept: True	positive: False		2

2.C Summary of model performance

Table 2.4: Model performance overview

		train			test					
					All SKUs			Only SKUs in train set		
					(n = 7, 667)			(n = 7, 177)		
	n	MSE	MAPE	Bias	MSE	MAPE	Bias	MSE	MAPE	Bias
ERP	1	830.6	39.35%	-3.15	932.1	31.34%	-4.54	898.7	31.28%	-4.51
Averages	1	387.2	28.91%	0.00	-	-	-	862.2	31.57%	-5.09
SES	1	403.3	29.87%	-0.15	-	-	-	876.4	31.99%	-5.26
LR*	2	484.9	34.00%	1.41	731.5	32.02%	-1.71	694.8	31.07%	-1.39
Weighted LR	4	492.2	35.62%	0.77	724.0	33.02%	-1.98	691.5	32.13%	-1.76
LR (US)	4	627.9	37.95%	0.64	720.7	32.67%	-2.23	691.0	31.85%	-2.09
LR (OS)	16	479.0	34.42%	1.99	788.7	32.89%	-1.56	753.7	31.72%	-1.24
Gradient boosting*	18	413.6	33.29%	1.35	650.9	30.36%	-1.24	623.4	29.73%	-1.11
Weighted gradient boosting	10	407.1	32.63%	1.07	646.4	29.75%	-1.55	620.2	29.13%	-1.42
Gradient boosting (US)	10	517.3	35.61%	0.68	646.8	30.23%	-1.57	618.8	29.63%	-1.47
Gradient boosting (OS)	14	431.1	33.37%	1.80	705.6	30.92%	-1.75	677.3	30.07%	-1.56
Random forest*	12	473.5	33.29%	0.39	618.5	25.24%	-1.39	585.6	24.81%	-1.32
Weighted random forest	12	504.2	34.52%	0.72	632.6	26.97%	-0.77	599.9	26.51%	-0.77
Random forest (OS)	18	557.6	36.82%	0.63	663.5	28.71%	-0.84	631.0	27.87%	-0.87
Random forest (US)	10	573.4	33.06%	-0.81	631.5	25.94%	-1.43	602.9	25.64%	-1.37

* Selected in cross-validation

Table 2.5: Simulation results for all methods

Service level	Model	Capital lockup	Capital savings
85%	ERP	1,218,123,965.88 €	—
	SES	1,153,739,599.75 €	5.286%
	LR	1,143,935,411.35 €	6.090%
	GB	1,141,816,798.08 €	6.264%
	RF	1,135,381,412.14 €	6.793%
90%	ERP	1,518,342,779.62 €	—
	SES	1,456,810,798.41 €	4.053%
	LR	1,443,757,794.03 €	4.912%
	GB	1,442,369,887.88 €	5.004%
	RF	1,424,368,535.77 €	6.189%
95%	ERP	2,200,420,450.99 €	—
	SES	2,115,939,863.53 €	3.839%
	LR	2,087,833,984.71 €	5.117%
	GB	2,070,314,210.50 €	5.913%
	RF	2,026,603,605.10 €	7.899%
98%	ERP	3,426,263,745.37 €	—
	SES	3,331,896,626.09 €	2.754%
	LR	3,309,870,953.32 €	3.397%
	GB	3,254,059,394.89 €	5.026%
	RF	3,126,957,703.56 €	8.736%

Chapter 3

Learning lead time demand distributions from sparse data: A conditional density estimation approach

The cumulative distribution function of lead time demand is central to inventory control applications (Babai et al. 2022). Historically, the inventory control literature has focused on theoretical lead time demand distributions. Some research has used non-parametric methods such as bootstrapping, particularly for items with non-standard demand patterns (Saldanha et al. 2023). However, non-parametric methods are difficult to apply when demand history is sparse, demand patterns are lumpy, or lead times are long (Zhou and Viswanathan 2011).

We consider such a setting in the context of spare parts inventory management. We develop a new non-parametric conditional density estimation approach based on machine learning that overcomes limitations of existing non-parametric techniques. To evaluate its performance, we use a periodic review cost model. We benchmark our method against established parametric and non-parametric alternatives using historical data for 6,695 SKUs from a global equipment manufacturer. Our results indicate that the proposed approach can lead to substential cost savings compared to the alternative approaches.

3.1 Introduction

Effective inventory management is critical to achieving high part availability at low inventory costs. Inventory management of spare parts is particularly challenging and faces unique challenges: demand is typically intermittent, lead times are long and volatile, and historical data is often sparse. Accurate estimation of lead time demand distributions allows for well-balanced inventory and service reliability.

Estimating the cumulative distribution function of lead time demand is challenging. Traditional inventory models often assumes specific probability distributions, such as the normal or gamma distributions for faster-moving items, and poisson, compound poisson, negative binomial or stuttering poisson distributions for low or intermittent demand patterns (Axsäter 2015). Empirical studies, however, have demonstrated that such parametric distributions frequently fail to accurately represent real-world lead time demand, particularly for spare parts characterized with intermittent and volatile demand (Syntetos et al. 2012, Turrini and Meissner 2019). This can lead to solutions resulting in excess inventory and capital lock-up or in back-orders and thus penalty costs.

The global equipment manufacturer ("company") we worked with faced this challenge. The company is a large global manufacturer of highly specialized industrial equipment, with multi-billion annual revenues and operations across multiple continents. To support its maintenance and repair services, the company maintains an extensive inventory of spare parts, comprising both internally produced components and externally sourced SKUs from a diverse global supplier network.

The company employs state-of-the-art optimization models that require lead time and demand distributions to determine reorder points, base-stock levels, and order quantities. The company found that the underlying assumptions about lead time demand distributions (e.g. poisson, negative binomial or normal) do not appropriately capture their setting. Demand patterns are highly intermittent and sporadic, and lead times are long. Because of production delays, shipping disruptions, and supply chain disruptions lead times are quite volatile. Lead time demands frequently exhibit skewness, multimodality, and other characteristics not captured by standard demand distributions, leading to suboptimal inventory decisions.

Given these limitations, non-parametric methods, notably bootstrapping techniques involving resampling historical demand data, may be suitable alternative approaches (Willemain et al. 2004, Syntetos et al. 2015, Hasni et al. 2019). Bootstrapping avoids rigid parametric assumptions and directly utilizes historical observations, making it potentially effective in capturing complex lead time demand patterns. However, despite their practical applications and adoption in commercial software (Willemain and Smart 2001), bootstrapping methods have important limitations. They often suffer from high variance and biased tail estimates when the sample size is small, which is common for spare parts with intermittent demand (see, e.g., Efron 1992, Hall 1992).

Recent developments in machine learning offer promising avenues for demand forecasting, extending beyond simple point predictions to distribution estimations. Machine learning methods have shown to capture intricate data patterns, by leveraging cross-learning across similar SKUs (Semenoglou et al. 2021, Makridakis et al. 2018a). Such cross-learning effects can significantly improve estimation accuracy. Despite these potential advantages, the application of machine learning for estimating lead time demand distributions in a multi-period setting, particularly within the context of intermittent demand typical of spare parts inventories, has not been addressed in the literature.

Our research addresses this gap by proposing a novel approach combining the strengths of bootstrapping methods with the capabilities of machine learning-based conditional density estimation. Specifically, we use distributional random forests integrated with bootstrapping to create a robust approach for estimating lead time demand distributions. The distributional random forest approach leverages cross-learning among SKUs, enhancing tail estimation accuracy when few historical data are available.

We also contribute to the literature by empirically validating our approach against classical parametric and non-parametric distribution-fitting approaches. Our results indicate substantial improvements offered by our approach, achieving reductions in inventory costs relative to traditional parametric methods, and performance superior to traditional bootstrapping techniques, in particular when back-order penalty costs are high.

The remainder of this paper is structured as follows: Section 3.2 reviews the literature on lead time demand estimation. Section 3.3 describes the problem setting and the data employed in our empirical analysis. Section 3.4 details our approach, its implementation. Section 3.5 presents the data used in our empirical analysis, describes the benchmark methods, and details the simulation-based evaluation framework. Section 3.6 presents and analyzes the empirical results. Finally, Section 3.7 discusses theoretical and practical implications, concludes our findings, and identifies future research opportunities.

3.2 Literature review

Base-stock is used to buffer against the upper tail of the lead time demand distribution function; mis-estimation of the distribution therefore results in high inventory levels or high back-order levels (Axsäter 2015). Estimating this distribution can be challenging as demand and lead time records are typically limited, irregular, noisy, and often statistically not well-behaved. Saldanha et al. (2023) categorized research for the estimation of lead time demand into two streams of literature – parametric and non-parametric approaches. In the following review, we provide an overview of the existing literature before synthesizing the motivation for our hybrid machine learning and bootstrapping approach.

3.2.1 Parametric approaches

Parametric approaches assume a specific lead time demand distribution and estimate the parameters of that distribution from historical data. The parametric stream originates in Fetter and Dalleck (1961) and Hadley and Whitin (1963), who express lead time demand as the random sum of order sizes that arrive during a stochastic lead time.

The main attraction of the approach is analytical convenience: solutions can be directly derived from the distributions. For fast moving items the normality assumption is often reasonable. For slow-moving or intermittently demanded items this assumption is generally inappropriate (Syntetos et al. 2011), leading to systematic inventory errors when normality is imposed (Turrini and Meissner 2019, Tyworth and O'neill 1997, Vernimmen et al. 2008). Therefore, several alternatives to the normal distribution have been proposed for modeling lead time demand, including the gamma, erlang, weibull, lognormal, and negative binomial distributions (Keaton 1995, Turrini and Meissner 2019, Tyworth and O'neill 1997, Vernimmen et al. 2008, Tadikamalla 1984, Levén and Segerstedt 2004, Shore 1986). For slower-moving demand, the negative binomial (poisson–logarithmic) and stuttering poisson (poisson–geometric) are often recommended (Axsäter 2015). However, these approaches share a common limitation: they perform poorly when the true demand distribution differs significantly from the assumed form, for instance, when empirical lead time demand shows multi-modality. Large-scale goodness-of-fit studies suggest that no single distribution consistently fits lead time demand across SKUs (Turrini and Meissner 2019, Syntetos et al. 2012). These studies indicate that standard distributions fail to provide an adequate fit for a substantial share of demand series, especially when assessed with goodness-of-fit tests sensitive to inventory-relevant tail behavior.

Parametric models offer analytical simplicity and work well when their assumptions hold, yet they struggle with multimodality and sparse data. These weaknesses have motivated distribution-free alternatives.

3.2.2 Non-parametric approaches

To address the limitations of rigid distributional assumptions, a growing body of research explores non-parametric methods. These approaches do not impose a specific functional form on lead time demand and are well-suited to spare parts with erratic or intermittent demand patterns (Saldanha et al. 2023).

A straightforward non-parametric approach is to use the empirical distribution derived directly from past lead time and demand observations (e.g. Fetter and Dalleck 1961). While simple, the empirical estimator is sensitive to sampling noise, particularly when data is limited.

Within the sample average approximation framework, Levi et al. (2007) show that the empirical solution performs well, if the sample size is large and the target service level is high. Levi et al. (2015) further show that the required sample size depends on demand volatility.

To improve robustness in data-scarce environments, resampling techniques such as bootstrapping have been proposed (Efron and Tibshirani 1986). Bootstrapping generates pseudo-samples by resampling with replacement from historical data. This enables estimation of the sampling distribution of key statistics, such as the critical fractile, and allows for bias correction and confidence interval construction

(Boylan and Babai 2022).

Several bootstrap variants have been developed for inventory problems. Bookbinder and Lordahl (1989) apply the bootstrap to lead time demand data, arguing that it handles skewed distributions effectively. However, this requires a sufficient number of lead time demand observations, which are often unavailable for spare parts. Moreover, the classical bootstrap appraoch assumes independent and identically distributed observations (iid). Wang and Rao (1992) address this by fitting an autoregressive model to daily demand and bootstrapping the residuals. While this accounts for autocorrelation, it introduces new model assumptions that may not hold under sparse data.

A two-step bootstrap method has also been proposed for cases where demand and lead time data are recorded separately: first sampling a lead time, then resampling the corresponding number of demands (Fricker and Goodhart 2000). Alternative approaches separately bootstrap inter-demand intervals and demand sizes (Viswanathan and Zhou 2008, Zhou and Viswanathan 2011) . While this method works well on simulated data, Zhou and Viswanathan (2011) show that it underperforms on real-world datasets with limited history, where parametric methods prove more robust. Huh et al. (2011) propose using the Kaplan–Meier estimator for censored sales data. This approach tends to struggle under long lead times and sparse demand.

More recently, a multivariate bootstrap framework has been introduced to accommodate stochastic lead times and irregular demand patterns (Saldanha et al. 2023). However, their experiments show it still requires a fair number of replenishment cycles to be reliable. As reviewed by Hasni et al. (2019), bootstrapping methods offer flexibility but often face difficulties when data is sparse or highly volatile – conditions common in spare parts environments.

A more recent stream of research integrates machine learning with inventory models. Ban and Rudin (2019) embed a regularized quantile regression into the newsvendor cost function and establish finite-sample bounds. Huber et al. (2019) compare tree ensembles, neural networks, and classical quantile regression, showing that most of the benefit comes from improved demand prediction. Cao and Shen (2019) develop a neural network for non-stationary demand quantiles and demonstrate improved performance in rolling inventory decisions. However, these methods remain largely confined to the single-period newsvendor setting. They do not directly address continuous-review inventory models with stochastic lead times.

Parametric methods are analytically convenient but often misrepresent the irregular and multimodal nature of lead time demand for spare parts. Non-parametric approaches, such as bootstrapping, avoid distributional assumptions but struggle with high variance and unreliable tail estimates when data is sparse. Machine learning methods offer promising improvements by leveraging patterns across SKUs, but existing applications focus on point forecasts or single-period quantiles.

A consistent challenge across all approaches is data sparsity at the individual SKU level. No existing method adequately addresses multi-period distribution estimation under long and volatile lead times with intermittent demand.

We address this by combining bootstrapping with distributional random forest to estimate full lead time demand distributions. Our approach learns across SKUs, remains non-parametric, and improves accuracy in data-scarce environments.

3.3 Problem description

The company is a global manufacturer of highly specialized healthcare equipment, supplying hospitals and clinical institutions with mission-critical systems. To support its maintenance operations, the company manages a network of warehouses holding tens of thousands of spare parts. These parts are essential to keeping equipment operational. When service teams are unable to promptly replace failed components due to stockouts, repair delays occur, and the company incurs contractual penalty costs.

To mitigate such risks while keeping inventory investment low, the company regularly optimizes inventory. The objective is to minimize the total expected cost, which includes both capital-related holding costs and back-order penalty costs. Central to this optimization is the estimation of the lead time demand distribution – the probability distribution of cumulative demand over a stochastic lead time. Inaccurate estimation of this distribution leads to suboptimal decisions: underestimation results in stockouts and penalty payments, while overestimation leads to excess inventory and tied-up capital.

While both parametric and bootstrapping approaches provide a means of estimating the lead time demand distribution, neither method is well suited for reliably determining base-stock levels under sparse and intermittent demand.

Figure 3.2 shows two lead time demand distributions derived from the same historical order data for a typical spare part (Table 3.1). The parametric estimate in Figure 3.2a is based on a fitted negative binomial distribution. To obtain its moments, separate forecasts were made for demand and lead time, with the mean and variance of lead time demand computed through convolution of these forecasts. In contrast, the empirical distribution in Figure 3.2b is constructed by bootstrapping historical order data directly.

Table 3.1: Order data

Order date	Lead time	Demand
2018-10-16	119	7
2019-02-06	119	7
2019-06-18	77	50
2021-07-29	123	17
2023-01-31	108	15

Figure 3.1: Inventory development

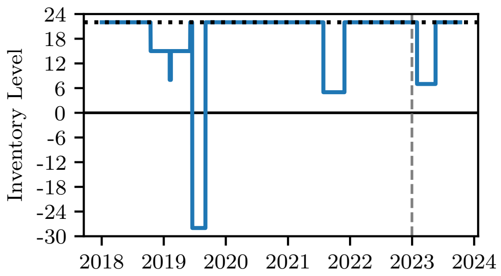
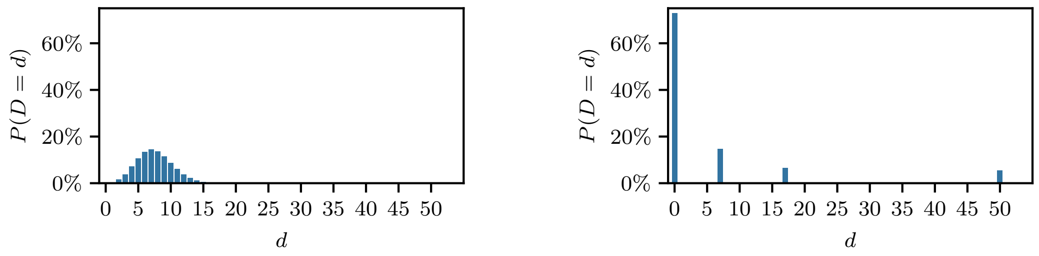


Figure 3.2: Lead time demand distributions under different estimation methods



(a) Lead time demand under parametric assumption

(b) Lead time demand sampled via bootstrapping method

The resulting shapes reflect the challenges imposed with such limitated data. In five years, only one order resulted in non-zero demand during lead time. This extreme sparsity, also visible in the inventory development in Figure 3.1, leads the bootstrapped distribution to concentrate probability mass at a few isolated values. The parametric distribution, by contrast, is smooth and unimodal – driven more by model assumptions than actual data.

Estimating lead time demand distributions based on such historical data is difficult. Parametric methods risk oversmoothing important structural features such as spikes, mass at zero, or multimodality. Non-parametric methods suffer from high variance and poor tail estimation under data sparsity, especially problematic when safety stock is set for percentiles with high penalty costs

This creates a core operational challenge: how can the company estimate realistic lead time demand distributions in the presence of sparse, intermittent demand and volatile lead times?

To address this, we propose a hybrid approach that combines bootstrapping with distributional random forest. This method leverages shared patterns across similar SKUs to improve estimation accuracy, particularly in the tails of the distribution. By integrating machine learning into the estimation process, the approach remains non-parametric, incorporates cross-sectional information, and is robust to the limitations of both classical parametric and non-parametric methods.

3.4 Methodology

This Section details the research methodology, commencing with the formal inventory modeling framework employed. It then elaborates on our proposed approach for lead time demand distribution estimation, which integrates bootstrapping with distributional random forest. Subsequently, we describe the benchmark parametric and non-parametric methods used for comparison. Finally, the evaluation framework for assessing the performance of these different approaches is presented.

3.4.1 Modeling framework

We consider a single-echelon, periodic review inventory system for managing spare parts, operating under an average cost criterion. The system is reviewed at the beginning of each time period $t \in \mathbb{N}$. At the start of each period, a replenishment order placed earlier may arrive. Demand $D_t \in \mathbb{N}_0$ is then realized, assuming that demand follows some probability density function $D_t \sim f_D(\cdot)$. Demand not met from available stock is back-ordered and fulfilled upon receipt of a subsequent replenishment. We assume that no order crossover is possible, as is rarely the case in our setting (cf. Section 2.4)

At the end of each period, a replenishment order is placed to restore the inventory position to a base-stock level $S \in \mathbb{N}_0$. Orders are subject to a stochastic lead time $L \in \mathbb{N}$ with known probability density distribution $f_L(\cdot)$ that is independent of probabilty density function $f_D(\cdot)$.

The inventory position at time t is $IP_t = I_t - B_t + O_t$, where I_t is the on-hand inventory, B_t is the back-order level at the beginning of period t, and O_t is the pipeline inventory. The order quantity Q_t placed at the end of period t is given by $Q_t = \max\{0, S - IP_t\}$.

Let $D^L = \sum_{t=1}^{L+1} D_t$ denote the cumulative demand over the stochastic lead time L, where $L \sim f_L(\cdot)$. We denote realizations of D^L by $d^\ell \in \mathbb{N}$. The distribution of D^L is induced by the convolution of the lead time distribution $f_L(\cdot)$ and the demand distribution $f_D(\cdot)$.

The system incurs linear holding and back-order penalty costs. Let h denote the per-unit holding cost per period, and b the per-unit back-order cost per period. The long-run expected cost per period under a given base-stock level S is

$$C(S) = h \cdot \mathbb{E}\left[S - D^L\right]^+ + b \cdot \mathbb{E}\left[D^L - S\right]^+. \tag{3.1}$$

The first term represents expected holding costs, while the second term represents expected back-order costs. The objective is to find the optimal base-stock level S^* that minimizes the expected cost:

$$S^* = \arg\min_{s \in \mathbb{N}} C(s). \tag{3.2}$$

While the true distribution $f_{D^L}(\cdot)$ is unknown, we have access to a dataset $\mathcal{D}_T = \{(\mathbf{x}_t, d_t^\ell)\}_{t=1}^T$, with observed realizations of lead time demand d_t^ℓ and corresponding covariates in vector $\mathbf{x}_t$ such as SKU characteristics. These observations are used to estimate the conditional probability density function $f_{D^L|\mathbf{X}}(d^\ell \mid \mathbf{x})$, which models the distribution of lead time demand given item-specific or contextual information.

3.4.2 Solution approach

Inspired by Bertsimas and Kallus (2020), we use machine learning to estimate the conditional distribution of lead time demand based on contextual features. Let $\mathcal{J}$ denote the set of all SKUs in our study. Our approach combines a temporal bootstrapping procedure with a distributional random forest to generate empirical estimates of the conditional probability density function $\hat{f}_{D_j^L|\mathbf{X}}(d^\ell|\mathbf{x}_{j,t})$ for each SKU $j \in \mathcal{J}$. Our model is trained on a dataset containing historical data for these SKUs across many time periods, enabling the estimation of these distinct probability functions.

We first construct a pseudo-sample of lead time demand realizations for each SKU j by applying a non-parametric bootstrapping procedure to its historical demand time series. Specifically, for each SKU j and each bootstrap iteration $b = 1, \ldots, B$, we randomly select a start date t_b from the historical demand data of a given SKU. We then draw a lead time realization $l_{j,b}$ from the empirical distribution of historical lead times observed for SKU j. The corresponding lead time demand is computed as $d^\ell_{j,b} = \sum_{i=0}^{l_{j,b}} D_{j,t_b+i}$, that is, the sum of daily demands for SKU j over the sampled lead time $l_{j,b}$. Each sampled lead time demand is paired with covariates $\mathbf{x}_{j,b}$ describing temporal and contextual features (e.g., timing, characteristics specific to SKU j and it's cluster information), forming an empirical training set $\mathcal{D}_B = \Big\{\Big(\mathbf{x}_{j,b}, d^{(l)}_{j,b}\Big) \mid j \in \mathcal{J}, b = 1, \ldots, B\Big\}$.

Next, we employ the distributional random forest approach as follows: we train a random forest regressor on $\mathcal{D}_B$ to model the relationship between the feature vectors $\mathbf{x}_b$ and the corresponding lead time demand realizations d_b^ℓ (Scornet et al. 2015). At the time of prediction, the forest structure provides a notion of local similarity between observations: for a new feature vector $\mathbf{x}$, we identify the set of training observations that fall into the same terminal nodes across the ensemble. Following a random forest conditional density construction adapted from Walzner et al. (2025), we use leaf-wise occurrence weights to form a local empirical distribution. Specifically, let $n_m(\mathbf{x})$ denote the terminal node of tree m to which the feature vector $\mathbf{x}$ is assigned, and let $\mathcal{N}_m(\mathbf{x})$ be the set of training samples that fall into terminal node $n_m(\mathbf{x})$. The aggregated neighborhood of $\mathbf{x}$ is then given by $\mathcal{N}(\mathbf{x}) = \bigcup_{m=1}^{M} \mathcal{N}_m(\mathbf{x})$ (Lin and Jeon 2006).

The weight structure induced by the random forest ensemble can be interpreted as a data-adaptive kernel, which effectively defines a non-parametric estimator (Scornet 2016). For each observation $i \in \mathcal{N}(\mathbf{x})$, we compute a weight $w_i(\mathbf{x})$ that reflects how often this observation co-occurs with $\mathbf{x}$ in the same terminal node across the forest, and the number of observations are in that node. Formally, we define the weight as

$$w_i(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \frac{\mathbb{1}\{i \in \mathcal{N}_m(\mathbf{x})\}}{|\mathcal{N}_m(\mathbf{x})|}, \tag{3.3}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, which evaluates to 1 if the condition inside the braces is true, and 0 otherwise. In this context, $\mathbb{1}\{i \in \mathcal{N}_m(\mathbf{x})\}$ is equal to 1 if training observation i appears in the same terminal node as $\mathbf{x}$ in tree m, and zero otherwise.

Using these weights, we construct an empirical conditional probability mass function over the observed lead time demand values in the neighborhood. Specifically, the estimated probability of observing lead time demand d^ℓ given feature vector $\mathbf{x}$ is computed as

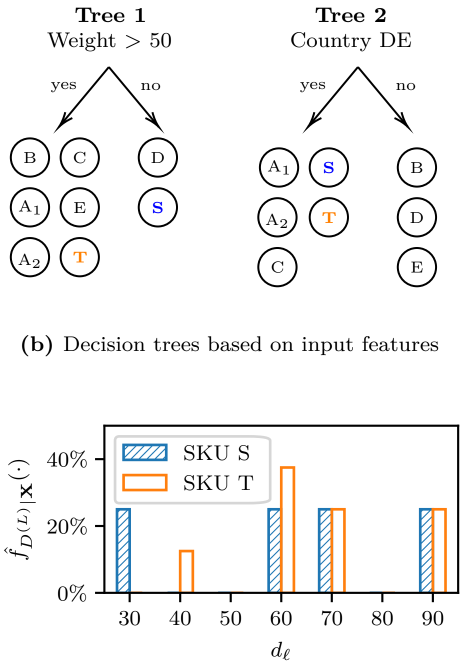
$$\hat{f}^{\mathrm{drf}}_{D_j^L \mid \mathbf{X}}(d^\ell \mid \mathbf{x}) = \sum_{i \in \{\mathcal{N}(\mathbf{x}) \mid d_i^\ell = d^\ell\}} w_i(\mathbf{x}), \tag{3.4}$$

SKU	Weight [kg]	Country	d^ℓ
A	55	DE	70
B	60	CH	40
C	75	DE	90
D	40	EN	30
A	55	DE	60
E	80	IT	60
S	35	DE	62.5
T	65	DE	67.5

(a) Raw input data $\mathcal{D}_B$

Sample	$w(\mathbf{x}_S)$	$w(\mathbf{x}_T)$
A_1	0.25	0.250
A_2	0.25	0.250
B	0.00	0.125
C	0.25	0.250
D	0.25	0.000
E	0.00	0.125

(c) Calculated weights $w_i(\mathbf{x})$



(b) Decision trees based on input features

(d) Estimated conditional proabilty density

Figure 3.3: Illustration of conditional distribution estimation with distributional random forest adapted from Walzner et al. (2025)

where the sum is taken over all training observations i in the neighborhood of $\mathbf{x}$ whose lead time demand value d_i^ℓ exactly matches d^ℓ. The total probability over all possible d^ℓ values sums to one due to the normalization of the weights.

This approach yields a discrete, data-driven approximation of the conditional distribution of lead time demand. The resulting density estimate can be directly inserted into the expected cost function in Equation 3.1 to compute inventory performance metrics and derive the optimal base-stock level S^*.

Figure 3.3 provides an intuitive illustration of how the distributional random forest estimates the conditional distribution of lead time demand. In Figure 3.3a, we see a small dataset where each SKU is described by features such as weight and country of origin, along with its observed lead time demand d^ℓ. Two new SKUs, labeled S and T, are introduced, and we aim to estimate their lead time demand distribution based on the patterns learned from the existing data. Figure 3.3b shows how decision trees split the data based on input features – for example, one tree splits on whether weight is above 50 kilograms, and the other on whether the SKU is beeing sourced from Germany (DE). Each tree groups SKUs into leaf nodes that represent locally similar observations. Figures 3.3c and 3.3d show the result of this grouping: for a new observation like S or T, we identify which training samples land in the same leaves, assign weights based on how frequently they co-occur, and then use those weights to build a weighted empirical distribution over past demand values.

To make this more concrete, consider the new SKU S. This SKU is assigned to the same terminal nodes as four training samples: D, A_1, A_2, and C. Among these, A_1 and A_2 are both instances of SKU A and have observed lead time demands of 70 and 60, respectively; SKU C has an observed lead time demand of 90, and SKU D of 30. Figure 3.3c shows that each of these four observations receives an equal weight of 25% when estimating the distribution for SKU S. Consequently, the estimated probability that the lead time demand of SKU S equals that of SKU D, namely 30, is 25%. The same applies to each of the other observed demand values (60, 70, and 90), resulting in a discrete empirical distribution as visualized in Figure 3.3d.

3.4.3 Benchmark estimators

To assess the performance of our proposed approach, we compare it against two reference approaches: a non-parametric temporal bootstrap (Saldanha et al. 2023) and a classical parametric estimator based on separate forecasting of demand and lead time (Eppen and Martin 1988).

The first benchmark estimates the distribution of lead time demand through an unconditional temporal bootstrap for each SKU independently. Let $\{D_t\}_{t=1}^{T}$

denote the historical demand time series for a given SKU, and let $\{l_k\}_{k=1}^{K}$ be the observed lead time realizations with K being the number of obsered lead times. For each bootstrap iteration $b = 1, \ldots, B$, a calendar date $t_b \in \{1, \ldots, T\}$ and a lead time realization $l_b \in \{l_1, \ldots, l_K\}$ are sampled independently. The corresponding lead time demand is computed as $d_b^{\ell} = \sum_{i=0}^{l_b} D_{t_b+i}$. The resulting sample $\{d_1^{\ell}, \ldots, d_B^{\ell}\}$ defines the empirical probability density function

$$\hat{f}_{D^L}^{\text{boot}}(d) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\{d_b^{\ell} = d\}, \tag{3.5}$$

which assigns probability mass to previously observed demand totals over randomly sampled lead time windows.

The second benchmark models demand and lead time as independent random variables. Demand is assumed to follow a negative binomial distribution, motivated by compound poisson demand processes in which intermittent arrivals and variable order sizes lead to negative binomial lead time demand (Quenouille 1949). This framework has been widely adopted in inventory settings involving spare parts and slow-moving items (Silver et al. 1998, Ritchie and Kingsman 1985, Prak et al. 2018).

To forecast demand $\hat{D}_t$, we apply Croston's method (Croston 1972), which separately models non-zero demand sizes z_t and inter-arrival times p_t. This method is regarded as the standard for intermittent demand forecasting. It is commonly implemented in commercial forecasting systems and is strongly connected to inventory control formulations, where it facilitates explicit safety stock calculation (Willemain et al. 1994, Johnston and Boylan 1996, Levén and Segerstedt 2004, Hyndman 2008, Syntetos et al. 2015). We also consider the Syntetos–Boylan Adjustment (SBA), a bias-corrected variant that modifies Croston's forecast to reduce systematic overestimation.

Both Croston and SBA apply SES to non-zero demand sizes z_t and inter-arrival times p_t, with forecasts updated only when demand occurs:

$$\hat{z}_t = \alpha z_t + (1-\alpha)\hat{z}_{t-1}, \quad \hat{p}_t = \alpha p_t + (1-\alpha)\hat{p}_{t-1}, \tag{3.6}$$

In Croston's method, the per-period demand forecast is given by $\hat{d}_{t+1} = \frac{\hat{z}_t}{\hat{p}_t}$, while the SBA correction adjusts this to $\hat{d}_{t+1} = \left(1 - \frac{\alpha}{2}\right)\frac{\hat{z}_t}{\hat{p}_t}$. These forecasts serve as the mean estimate for the per period demand distribution $f_D(\cdot)$. The variance is estimated using the squared forecast errors $\widehat{\mathbb{V}}[D_t] = \frac{1}{n}\sum_t (d_t - \hat{d}_t)^2$.

Lead time L is modeled as a discrete random variable, independent of period demand. Since lead time is observed only on delivery days it forms a sparse, irregular time series. We forecast lead time using SES, which is well-suited to low-noise, non-seasonal data. Let $\mathcal{T}$ denote the set of time periods at which deliveries occur (i.e., the periods when previously placed orders arrive). Let $\{l_\tau | \tau \in \mathcal{T}\}$ denote the observed lead times corresponding to those delivery periods. SES is applied only at these points:

$$\hat{l}_\tau = \alpha l_\tau + (1-\alpha)\hat{l}_{\tau-1}, \tag{3.7}$$

with the forecast held constant between updates $\hat{l}_t = \hat{l}_\tau$ for $\tau_i < t < \tau_{i+1}$.

Given the forecasts for demand $\hat{d}_t$ and lead time $\hat{l}_t$, and the corresponding squared forecast errors $\widehat{\mathbb{V}}[D_t]$, $\widehat{\mathbb{V}}[L]$, the moments of the lead time demand distribution are calculated using established formulas for the mean and variance of the sum of a random number of random variables (Zipkin 2000, p. 285)

$$\mu_{D^L} = \hat{d}_t\hat{l}_t, \qquad \sigma^2_{D^L} = \hat{d}_t\hat{l}_t + \widehat{\mathbb{V}}[D_t]\hat{l}_t + \hat{d}_t{}^2 \cdot \widehat{\mathbb{V}}[L]. \tag{3.8}$$

These moments define a negative binomial approximation $\hat{f}_{D^L}^{\text{par}}$ with parameters

$$r = \frac{\mu_{D^L}^2}{\sigma_{D^L}^2 - \mu_{D^L}}, \quad p = \frac{r}{r + \mu_{D^L}} \tag{3.9}$$

provided $\sigma_{D^L}^2 > \mu_{D^L}$. In the few cases where this condition fails, the variance is set equal to 1.1 times the mean. Although this may look ad hoc, Sani (1995) shows that it produces robust results.

3.5 Evaluation procedure

To analyze the performance of the proposed approaches, we evaluate each approach within a cost-based inventory simulation. The evaluation focuses on inventory performance across a diverse portfolio of spare parts.

3.5.1 Data description

We use the dataset of Reiners et al. (2025), which consists of transactional purchase order records and master data from the ERP system of the company. Data spans from January 2018 to October 2023 and includes 219,985 purchase orders for 17,728 distinct SKUs.

Each transaction captures historical order quantities, realized lead times, valuation prices, and additional supplier and SKU-level attributes. Table 3.2 provides an overview of the relevant features used in this study. The final feature set includes both raw transactional data and engineered variables such as SKU cluster assignments to support learning across related items.

Details on the cleaning, filtering, and feature engineering steps are documented in Reiners et al. (2025)

Table 3.2: Fields used in this study

Field	Description
Part ID	Internal identifier for each SKU
Order date	Purchase order creation date
Order quantity	Number of units requested in the order
Lead time	Realized procurement lead time in calendar days
Delivery date	Date the item was received at the warehouse
Valuation price	Unit cost of the item in euros
Supplier ID	Internal identifier for the supplier
Supplier country	Country of origin of the supplier
Information record	ERP-level material–supplier reference
Cluster ID	Feature-engineered categorical SKU grouping

3.5.2 Model tuning and forecast optimization

To ensure a fair and meaningful comparison, each estimation method is tuned and validated using a consistent time-based data split: training data spans from January 1, 2018 to December 31, 2021; validation is performed from January 1st, 2022 to December 31, 2022; and test performance is evaluated from January 1st, 2023 to October 11, 2023. This split allows model selection based on a complete seasonal cycle of historical data while maximizing the available training set. Moreover, it approximately reflects a typical 80/20 train-test split when training and validation periods are combined, while preserving the temporal structure essential to time series modeling (Vandeput 2021).

For the proposed distributional random forest approach, model performance is influenced by the structure of the individual trees in the ensemble. We therefore tune the forest using 5-fold cross-validation, optimizing the configuration of hyperparameters (Hastie et al. 2009). To find appropriate hyperparameter values, we use a grid search approach with a maximum entropy grid design with 100 parameter combinations that fill the defined parameter space (Santner et al. 2003). These include ensemble parameters (e.g., number of estimators, minimum leaf size), time-series lags and moving averages, trend and seasonal extraction flags, categorical encoding schemes, and similarity-based clustering thresholds. We then choose the hyperparameter configuration that minimizes the RMSE averaged over the five folds.

For the parametric benchmark, the smoothing parameters α and β (cf. Equations 3.6, 3.7) in the Croston, SBA and SES methods are tuned via grid search over the interval $[0.05, 0.3]$, using RMSE on the validation set as the selection criterion. For each stock keeping unit, the method that achieves the lowest rolling one-step-ahead squared forecast error on the validation set is retained as the final model.

The bootstrap estimator has few tunable parameters. The number of bootstrap iterations B is fixed at a large value (e.g., $B = 10,000$) to ensure stability of the empirical distribution. Sampling is restricted to time points that allow full lead time intervals within the demand history, thereby avoiding truncation artifacts.

A short summary of the hyper-parameters and their optimal values for each algorithm can be found in Table 4.3.

3.5.3 Simulation framework

We simulate a single-echelon periodic review inventory system using historical demand and lead time data from the company. For each SKU, we evaluate how the different estimation methods affect inventory performance when used to determine base-stock levels.

For each estimation method (distributional random forest, bootstrap, parametric), we compute the estimated distribution $\hat{f}_{D^L}(\cdot)$ for a given SKU and derive the cost-optimal base-stock level $\hat{S}$ by solving the optimization problem in Equation 3.2. Using the actual historical realizations of demand and lead time, we then simulate inventory operations over an out-of-sample evaluation window as follows:

At the beginning of each period t, the inventory position $IP_t = I_t - B_t + O_t$

Table 3.3: Tuned hyperparameters and selected values for each estimator

Method	Hyperparameter	Tuning range	Selected value
Distributional random forest	Number of trees	50, 100, 200, 300	200
	Min samples per leaf	5, 10, 25, 50, 100	5
	Bootstrap samples	100, 500, 1000	100
	Multivariate	True, False	True
	Lag periods	{1}, {1,2}, {1,2,3}, {1,3}	{1}
	Rolling window periods*	{2}, {3}, {2,3}	{2,3}
	Rolling window function**	simple, exponential	exponential
	Enable trend	True, False	True
	Enable seasonality	True, False	False
	Categorical encoder	CatBoost, Target, LOO, James-Stein	James-Stein
	Text similarity threshold	0.5, 0.6, 0.7, 0.8, 0.9	0.6
	Minimum cluster size	1, 2, 3, 5, 10, 100	5
Parametric (SES, Croston, SBA)	Smoothing parameter α	$[0.05, 0.3]$ (grid search)	chosen per SKU
	Smoothing parameter β (SBA only)	$[0.05, 0.3]$ (grid search)	chosen per SKU
Bootstrap estimator	Number of bootstrap iterations (B)	Fixed	10,000
	Sampling constraint	—	Full lead time only

* defines window lengths p used to compute static. With $p \in \{2, 3\}$ separate statistics are computed for each p, using data up to time $t-1$ only (the target is lagged one step).

** selects the statistic per window. *Simple* uses the equal–weight mean over the last p values. *Exponential* uses an exponentially weighted mean with span p, which maps to $\alpha = 2/(p+1)$ and weights $w_k = \frac{\alpha(1-\alpha)^{k-1}}{1-(1-\alpha)^p}$, $k = 1, \ldots, p$.

is reviewed. If $IP_t < \hat{S}$, a replenishment order of size $Q_t = \hat{S} - IP_t$ is placed. Deliveries occur after the realized lead times recorded in the historical data. After delivery and potential fulfillment of back-orders, demand d_t is realized and either satisfied from inventory or back-ordered. Lead times and demands are treated as empirically observed and independent.

To account for transient effects, the initial warm-up period is excluded from evaluation. Cost performance is assessed over the remaining horizon, based on (i) average on-hand inventory valued at an annual holding cost rate of 10 % of the SKU's unit price, (ii) average back-orders valued at a per-unit penalty cost, and (iii) total average cost as defined in Equation 3.1.

3.6 Results

This Section presents the empirical results of the simulation study described in Section 3.4. We evaluate and compare inventory performance across the three lead time demand estimation methods – parametric, non-parametric bootstrapping, and the proposed distributional random forest under varying levels of back-order penalty cost. Figure 3.6 summarizes the main findings in terms of back-order penalty cost, inventory holding cost and total cost.

3.6.1 Model analysis

To better understand the distributional random forest, we examine the internal behavior of the model through its feature importance structure. Figure 3.4 presents the relative importance scores of the input variables, as estimated by the trained ensemble. The most influential features are “Part ID”, the most recent demand lag “Lag 1”, and short-term trend estimates.

The prominence of “Part ID” in particular suggests that the model places substantial weight on SKU-specific characteristics. Since “Part ID” id is encoded as a high-cardinality categorical variable, the forest can use it as a proxy for average lead time demand. However, while this indicates a strong reliance on SKU identity, it does not mean the model ignores cross-sectional information.

The forest’s structure ensures that predictions for a given SKU are informed not only by its own history, but also by training examples from other SKUs that fall into similar terminal nodes. This occurs when SKUs exhibit comparable values in the temporal features – especially lagged demand and trend estimates. In other words, even though “Part ID” appears important, the forest performs an implicit form of local averaging across SKUs that are similar in their recent lead time

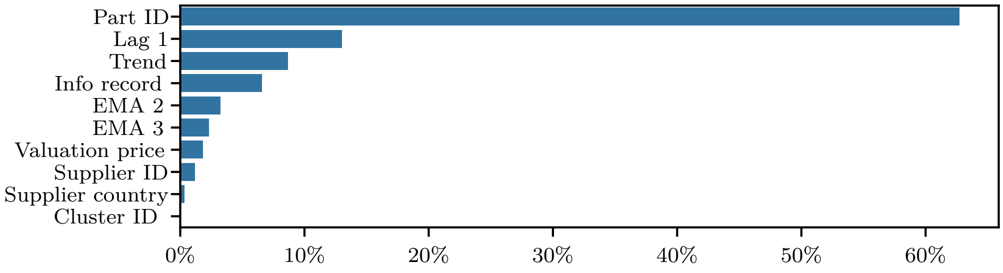


Figure 3.4: Feature importance of the distributional random forest model

demand behavior.

Figure 3.5 shows estimated lead time demand distributions for a representative SKU using three different methods: parametric estimation, bootstrapping, and the distributional random forest. The parametric estimate appears as a smooth, unimodal distribution with a clear central tendency and rapidly declining tails. The bootstrap estimate, by contrast, consists of a small number of discrete spikes, with the majority of probability mass concentrated at a few historical values. The distributional random forest yields a more dispersed shape, with multiple peaks and smoother transitions between demand levels, including non-zero probabilities in regions not covered by the bootstrap.

These shapes reflect how each method handles limited historical information. The parametric method imposes a predefined distributional form including probabilities for unobserved lead time demands, including non-negligible probability mass to the extreme upper tail of the distribution. The bootstrap method, relying exclusively on empirical resampling, reflects only those values previously observed. As a result, when few historical demand realizations exist, the estimate becomes sparse, and the tail ends of the distribution are poorly represented.

The distributional random forest exhibits a broader and smoother estimate, including lead time demand levels not observed for the target SKU. This can be attributed to the way the model aggregates information from other SKUs with

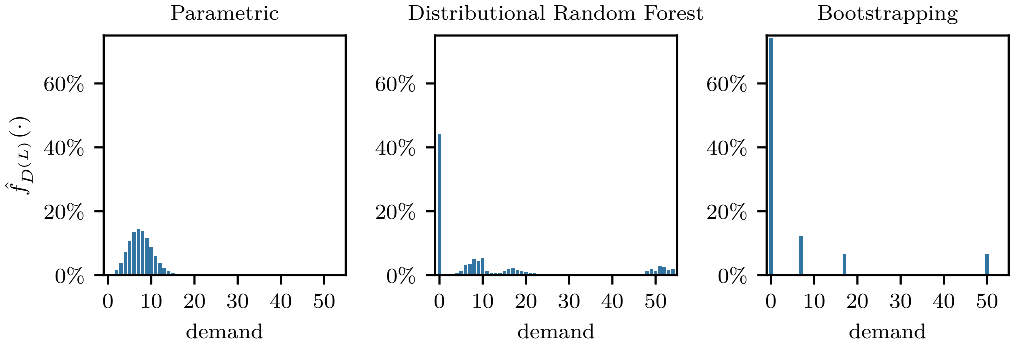


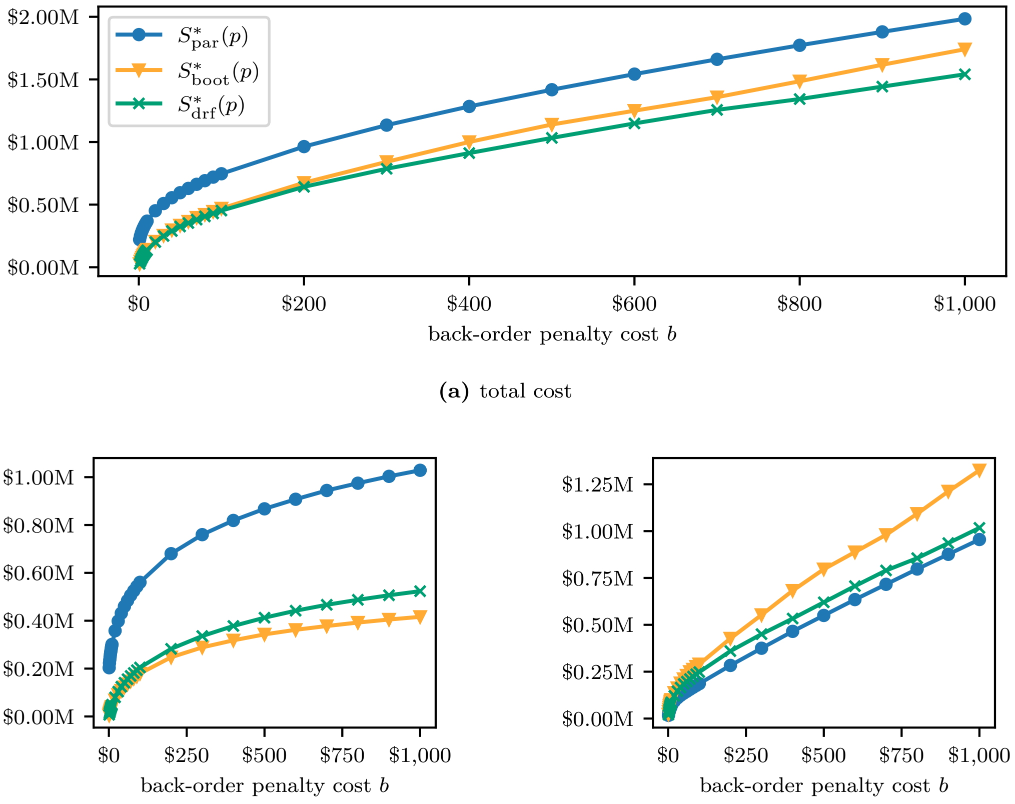
Figure 3.5: Comparison of lead time demand distribution estimates for a representative SKU

similar temporal features, such as lagged lead time demand and trend. When those SKUs have experienced higher demand levels during lead times, the model may assign probability to those outcomes – even if they are not present in the focal SKU's history.

3.6.2 Aggregate results

Figure 3.6a shows the total cost incurred by each estimation method as a function of the unit back-order penalty cost b. The proposed distributional random forest consistently achieves the lowest total cost across all penalty levels. The parametric estimator performs worst overall, while the bootstrapping method yields intermediate cost levels. At low unit back-order penalty cost values, the total cost curves for the bootstrapping and distributional random forest methods are nearly indistinguishable. As the penalty increases, a growing performance gap emerges: the bootstrapping method incurs increasingly higher costs compared to the forest-based approach.

Figures 3.6b and 3.6c provide the decomposition of total cost into inventory holding cost and back-order penalty cost, respectively. The parametric method leads to the highest inventory levels, followed by the forest and then the bootstrap. In terms of back-order cost, the bootstrap method performs worst, particularly at



(a) total cost

(b) inventory cost

(c) back-order penalty cost

Figure 3.6: benchmark simulation results

high penalty levels. The distributional random forest maintains back-order costs that are consistently and substantially lower than those of bootstrapping, despite only slightly higher inventory levels.

These results can be interpreted by examining the differences in how each method estimates the cumulative distribution function of lead time demand. While both the bootstrap and the distributional random forest rely on empirical information, their treatment of the distribution tails differs significantly. At low unit back-order penalty cost levels, inventory policies are more responsive to the distribution's central mass, where both methods provide similar estimates, explaining the near-identical total cost. However, as the penalty cost increases, the inventory model becomes more sensitive to tail risk. The bootstrap method, due to limited sample size, places insufficient probability mass in the tail regions, leading to systematic underestimation of high-demand scenarios and consequently elevated back-order costs.

In contrast, the distributional random forest draws on cross-sectional patterns and smooths over sparse observations, thereby allocating more probability mass to extreme values. This results in slightly higher inventory levels but prevents costly stockouts under high-penalty conditions. The model anticipates heavy-tail risks better than pure bootstrapping, reducing expected penalty costs and improving overall inventory performance.

3.6.3 Stratified analysis by order frequency

To better understand how the performance of each method evolves with data availability, we conduct a stratified analysis based on order frequency. Figure 3.7 presents simulation outcomes for five sets of SKUs, ranging from extremely sparse (only one training observation) to relatively dense (more than four observations per year). Each row in the figure reports total cost, inventory cost, and back-order cost for each estimation method across penalty levels.

In the first row (166 SKUs with only one historical observation), the distributional random forest and bootstrap methods perform nearly identically across all cost components. This is expected: with a single training observation, lag features are undefined and the forest model cannot leverage temporal patterns or cross-sectional learning. In this regime, the distributional random forest essentially collapses to an empirical bootstrap. With only one sample in the train set the parametric estimator cannot fit an arrival rate and therefore collapses to a last observed demand forecast. This yields large base-stock levels resulting in access inventory costs but low back-order costs, showing that fitting a parametric model to a single observation is ill posed.

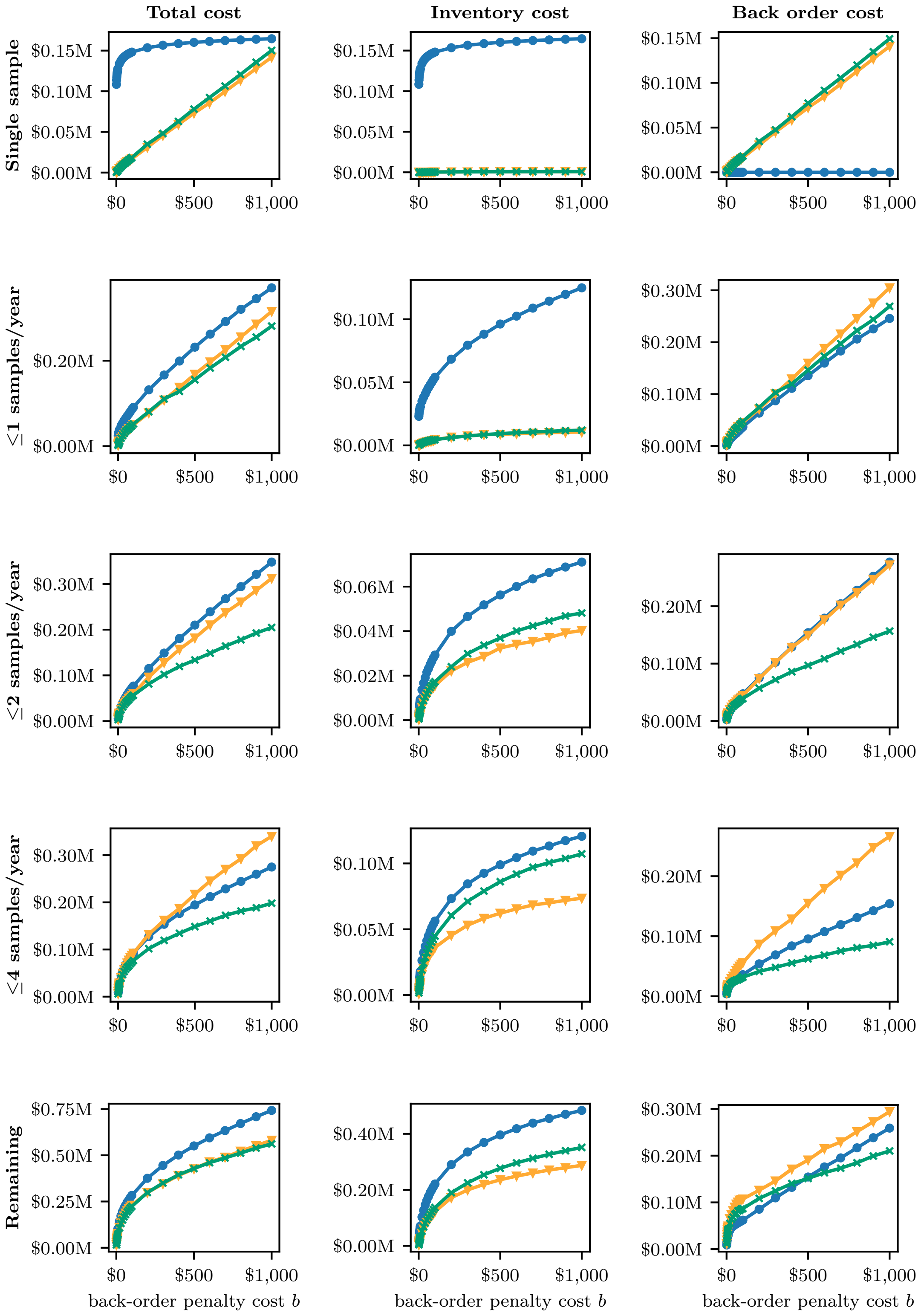


Figure 3.7: Simulation results by order frequency cohort

In the second and third row (788 and 1,521 SKUs with one to two observations per year), the forest begins to outperform the bootstrap method. With even minimal additional data, the random forest can incorporate information from similar SKUs and exploit structure in the feature space. As a result, it maintains comparable or slightly higher inventory levels while significantly reducing back-order costs – especially as penalty levels rise. The cost advantage of the forest approach becomes most pronounced in these low-data settings, where the bootstrap still struggles with tail estimation.

In the fourth and fifth rows (1,931 and 2,267 SKUs with higher observation counts), the performance of all methods begins to converge. The bootstrap method benefits from richer historical data and produces more stable estimates of the lead time demand distribution, including its tails. The forest continues to deliver the lowest total cost, but the marginal advantage over bootstrapping narrows considerably. In high-data regimes, the cost of model misspecification (in the case of the parametric estimator) remains the primary driver of suboptimality.

Across all cohorts, the parametric estimator performs worst because data is too sparse to fit the negative binomial model reliably. The imposed distributional form assigns non-zero probability to unobserved high-demand outcomes used in the base-stock calculation. Parts for which the parametric fit overestimates tail mass receive higher base-stocks, increasing inventory cost relative to both the bootstrap and the distributional random forest. While this additional stock reduces back-orders, the penalty savings are insufficient to compensate for the additional holding cost, so total cost remains highest for the parametric approach. Two factors plausibly amplify this effect: small-sample parameter bias that inflates variance estimates and a concentration of expensive SKUs for which modest stock increases translate into disproportionately higher holding cost. Together, these mechanisms explain the large and persistent total-cost delta of the parametric method across low-data settings.

3.7 Conclusions

This study addresses a central challenge in spare parts inventory management: the accurate estimation of lead time demand distributions under conditions of sparse, intermittent demand and volatile lead times. Traditional parametric models, while analytically convenient, exhibit limited reliability when historical data is sparse. With only a few observations, the data provides almost no reliable information about the true underlying lead time demand behavior, resulting in highly uncertain distribution estimates. Non-parametric bootstrapping methods avoid restrictive functional forms but also produce unstable results when historical data is sparse – particularly in the tails, where inventory decisions are most sensitive.

To overcome these limitations, we propose a novel approach that combines temporal bootstrapping with distributional random forests. This method preserves the non-parametric nature of bootstrapping while leveraging the cross-sectional and temporal structure of the data through machine learning. Our results, based on a large-scale empirical study using data from a global equipment manufacturer, demonstrate that this approach consistently outperforms both classical parametric and non-parametric benchmarks in terms of total cost, particularly as back-order penalties increase.

The performance gains are most pronounced for SKUs with sparse historical data, where traditional methods are least reliable. By enabling cross-learning across SKUs and improving tail estimation, the proposed method reduces costly stockouts without requiring significant increases in inventory levels. In extreme cases, such as when only a single training observation is available for an SKU, the distributional random forest reduces to behavior similar to bootstrapping, as little cross-learning is possible. Conversely, when sufficient historical data is available and bootstrapping performs reliably on its own, the added benefit of the forest structure diminishes. These boundary cases highlight the adaptive nature of the method – it maintains performance in data-poor conditions and converges toward standard bootstrapping when data is abundant.

Practically, our findings suggest that companies can improve spare parts inventory performance by integrating machine learning techniques that move beyond point forecasting and embrace full distributional estimation. From a theoretical perspective, the results highlight the value of combining non-parametric estimation with data-adaptive learning in inventory control applications.

Future research could extend this work by exploring other models for distribution estimation, integrating external covariates such as macroeconomic indicators or supplier performance metrics, and applying the method to multi-echelon inventory systems. Furthermore, there is potential to explore adaptive learning mechanisms that update demand distributions in real time as new data becomes available.

In conclusion, the proposed approach represents a significant advancement in the estimation of lead time demand distributions, offering both theoretical innovation and practical value for inventory management in complex, data-constrained environments.

Appendix of Chapter 3

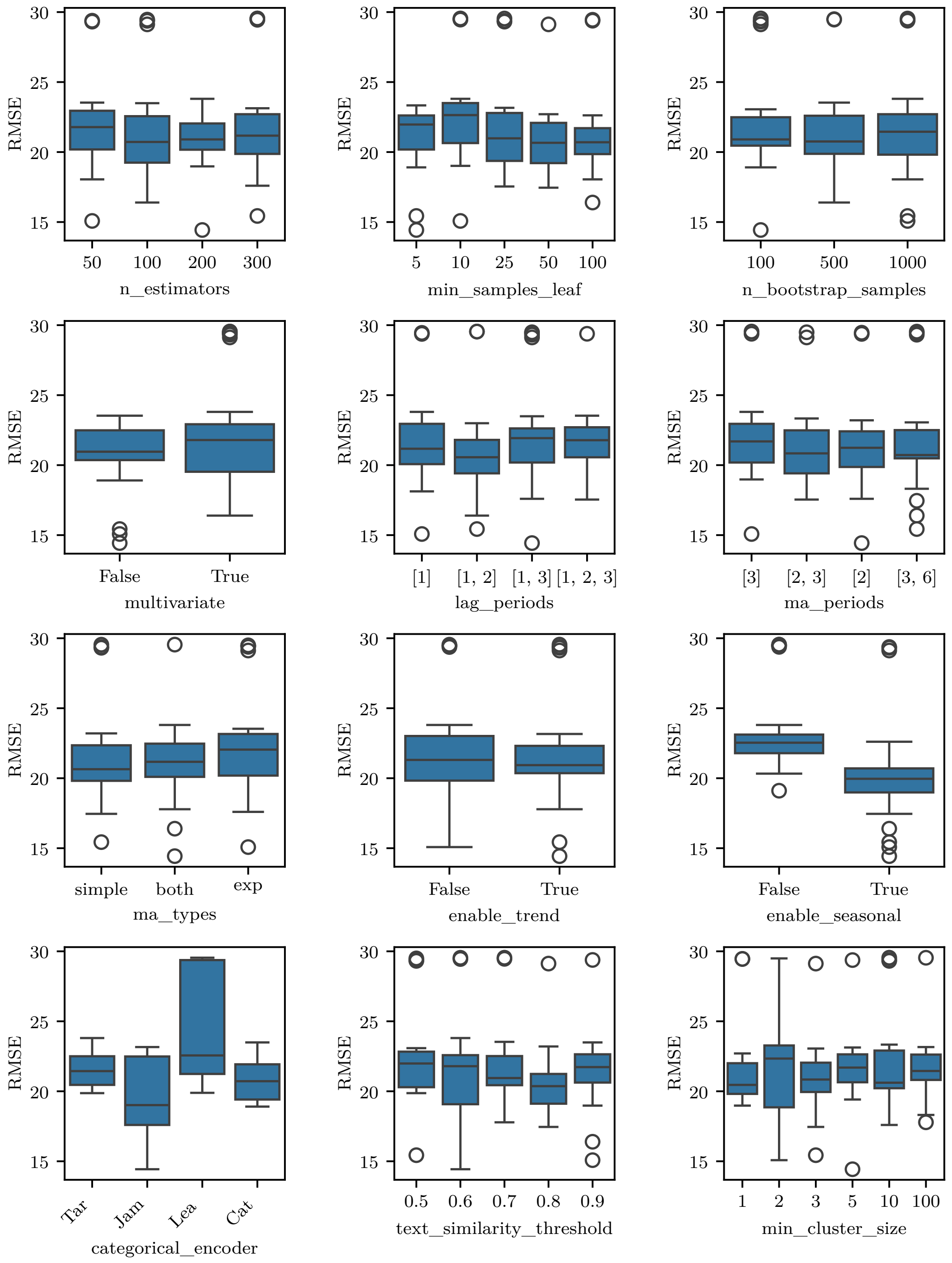


Figure 3.8: Sensitivity of the distributional random forest to hyperparameter settings

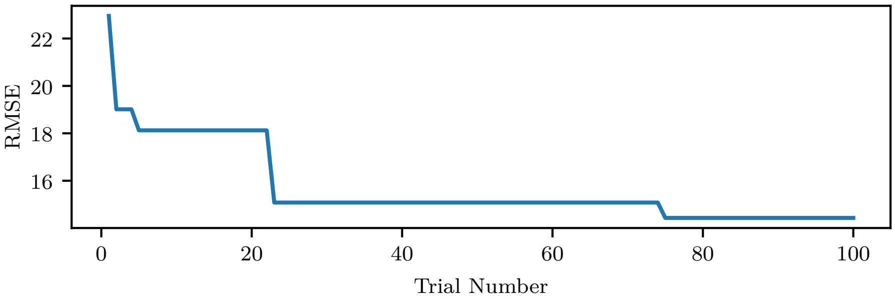


Figure 3.9: Tuning convergence of the distributional random forest

Chapter 4

Cross-learning for lead time demand forecasting

In spare parts supply chains, sparse and intermittent observations complicate lead time demand estimation, which is a critical requirement for inventory management. Despite an extensive body of literature it remains an active area of research (Nikolopoulos 2021). Traditional methods typically estimate each stock-keeping-unit (SKU) independently. In contrast, cross-learning methods share information across related SKUs and have performed well in forecasting competitions (Makridakis et al. 2018a, 2022), learning transferable patterns (Semenoglou et al. 2021).

We study when cross-learning improves estimation of lead time demand. We evaluate a cross-learning estimator (distributional random forest) against two series-by-series benchmarks: a non-parametric bootstrap and a parametric negative-binomial model. In a simulation experiment, we vary demand intermittency as well as alignment between features and true demand behavior. The methods are evaluated based on realized inventory cost in a periodic-review base-stock system against the cost-optimal policy from the true distribution.

Results show that when features are informative, cross-learning achieves near-optimal solutions and yields the largest gains for slow moving SKUs. With misleading features, cross-learning converges toward the bootstrapping estimator, indicating robustness. The parametric method shows sensitivity to demand intermittency. These results inform when to prefer each approach: cross-learning when features are informative and data are scarce, bootstrapping when features are uninformative or intermittency is moderate, and parametric models in data-rich conditions.

4.1 Introduction

Effective spare parts inventory management relies on accurate estimates of lead time demand, that is, the cumulative demand that materializes over a replenishment lead time. In contrast to fast-moving consumer goods, spare parts exhibit intermittent, low-volume, and highly skewed demand with substantial heterogeneity across SKU. These characteristics challenge classical, series-by-series forecasting approaches that rely either on strong distributional assumptions or on sufficient demand data for each SKU. When the demand data per SKU are scarce, point and distributional forecasts of lead time demand become inaccurate, which in turn propagates to high inventories or large back-orders.

For the case of univariate data parametric and non-parametric approaches have been proposed to address this challenge. Parametric approaches assume a specific distributional form for lead time demand (e.g., negative binomial) and estimate the corresponding parameters from past data (Axsäter 2015, Babai et al. 2021). While parametric methods can be efficient when data are rich and distributional assumptions are met, they perform poorly when distributional assumptions are violated. Non-parametric methods make no assumptions about the distributional family. The most common is bootstrapping, which constructs an empirical distribution by resampling observed demand histories (Willemain et al. 2004). Non-parametric bootstrapping is more flexible but struggles when historical data are sparse. For overviews of these approaches, the reader is referred to Syntetos et al. (2016) and Hasni et al. (2019).

Another stream of literature exploits feature information across SKUs. If SKUs share product attributes or usage contexts, pooling data can improve estimation of the predictive distribution of lead time demand. Evidence in support of such cross-learning comes from forecasting competitions (Makridakis et al. 2018a, 2022), from recent work on installed-base forecasting (Van der Auweraer et al. 2019, 2021), and from recent work on data-driven inventory management that leverages pooled data (Reiners et al. 2025). However, the success of cross-learning rests on a tacit assumption: feature similarity aligns with similarity of demand behavior over lead time. When this alignment fails (e.g., SKU look similar but serve different usage contexts), cross-learning can be misleading.

The contribution of this paper is to provide an assessment of these trade-offs through a controlled analysis. We design a simulation experiment that allows us to vary both the demand rate (slow vs. fast movers) and the degree of alignment between feature-based and demand-based similarity. Within this experiment we benchmark three approaches: a cross-learning estimator (distributional random forest) that exploits information on SKUs, a bootstrap estimator, and a parametric estimator based on the negative binomial distribution.

The analysis focuses on three dimensions. First, we examine the size of the efficiency gains that cross-learning can deliver when feature similarity genuinely reflects demand behavior. Second, we investigate the robustness of cross-learning when features are misleading.Third, we examine how these effects vary with the demand rate. The findings provide indications when to use cross-learning methods in spare parts inventory management and when to apply traditional methods.

The remainder of the paper is organized as follows. Section 4.2 reviews related work. Section 4.3 formally describes the problem of estimating lead time demand distributions as well as the inventory system and policy. Section 4.4 discusses the design of our simulation experiment after which the numerical results are presented and discussed in Section 4.5. Finally, Section 4.6 concludes with implications for research and practice.

4.2 Literature review

Prior research on lead time demand estimation can be grouped by the extent to which methods treat SKUs in isolation or exploit cross-SKU information. Classical inventory models typically assume that each SKU is forecasted separately, while more recent approaches aim to pool information across SKU through hierarchical modeling or machine learning. This review first summarizes single-SKU estimation methods, then discusses approaches to cross-SKU learning, before positioning our study in this landscape.

Single-SKU. Early models typically estimate lead time demand distributions independently for each SKU, either through parametric or non-parametric approaches. Parametric models impose a functional form (e.g. normal, gamma, weibull, negative binomial) and estimate parameters from historical observations (Hadley and Whitin 1963, Keaton 1995, Vernimmen et al. 2008, Turrini and Meissner 2019). They often provide closed-form inventory solutions and perform well when demand is fast-moving and assumptions hold. However, they often misestimate distributions of spare parts from industry datasets (Syntetos et al. 2012, Axsäter 2015). Bayesian extensions such as the poisson-gamma and compound poisson bayesian models have shown to improve fit over traditional frequentist models and inventory efficiency by capturing variability in arrivals and batch sizes (Aronis et al. 2004, Babai et al. 2021), though they add complexity.

Non-parametric estimators such as empirical distributions or bootstrapping avoid restrictive assumptions (Bookbinder and Lordahl 1989, Fricker and Goodhart 2000, Zhou and Viswanathan 2011). They can capture skewness in the distribution but rely on sufficient SKU-level demand data to be reliable. When demand data are sparse, resampling methods can suffer from high variance and may lead to inaccurate tail estimates (Syntetos et al. 2015, Hasni et al. 2019, Boylan and Babai 2022). While extensions to bootstrapping incorporate time-series dynamics (Wang and Rao 1992, Willemain et al. 2004, Zhou and Viswanathan 2011) or multivariate sampling (Saldanha et al. 2023), these remain fundamentally single-SKU estimators.

Cross-SKU. Another stream of literature replaces series-by-series fitting with joint learning. Some studies cluster SKU or stratify demand series before applying bootstrap-based resampling, effectively borrowing information across similar SKUs. For example, association-rule mining and hierarchical clustering have been used to group spare parts with correlated consumption, allowing joint replenishment decisions (Tsai et al. 2009, Moharana and Sarmah 2018). Recent work demonstrates that hierarchical clustering combined with joint replenishment policies can yield significant cost reductions compared to ABC classification, k-means clustering, or single-SKU management (Lolli et al. 2025). This approach illustrates how clustering can act as a form of cross-SKU learning, though its effectiveness depends on the chosen similarity metric and whether it reflects true demand dependencies.

Machine learning models are capable of learning from multiple demand series to accurately predict individual ones (Semenoglou et al. 2021). Early applications embed machine learning into the newsvendor setting, employing quantile regression, tree ensembles, or neural networks (Ban and Rudin 2019, Huber et al. 2019, Cao and Shen 2019). These approaches demonstrate improved forecast accuracy from leveraging patterns across SKU, though they often focus on point forecasts or single-period quantiles.

Cross-learning pools information across SKU. In forecasting, this is implemented with global models that are trained on many series jointly rather than local models fitted per SKU. Empirical studies show that global models can improve accuracy when SKU histories are short and patterns recur across series. These gains are strongest for heterogeneous but related parts and when model architectures can extract shared structure at scale (Makridakis et al. 2018b, Semenoglou et al. 2021). A key reference is Semenoglou et al. (2021), who evaluate cross-learning neural networks on the M4 monthly data. They report consistent improvements over series-by-series training when cross-learning is paired with appropriate feature extraction and regularization. They also document regimes where gains diminish, such as highly idiosyncratic series. These findings position cross-learning as a practical remedy for sparse histories. While this tension has been noted in forecasting competitions (e.g., Makridakis et al. 2018a, 2022), it remains underexplored in the spare parts inventory context.

In summary, single-SKU estimators whether parametric or non-parametric are vulnerable to demand data sparsity and heterogeneity. Cross-learning methods, from hierarchical models to modern machine learning, offer a way to pool information, but their success depends critically on the alignment between observed features and latent demand similarity. Our study contributes by systematically evaluating this alignment: we benchmark parametric, bootstrap, and distributional machine learning estimators under conditions of both aligned and misaligned similarity, providing boundary conditions for when cross-learning improves or undermines predictive accuracy.

4.3 Description of the inventory system and policy

We consider the task of estimating lead time demand distributions for spare parts inventory management. The inventory system is modeled as a single-echelon, periodic review base-stock system. The sequence of events in a period is as follows:

At the beginning of each period $t \in \mathbb{N}$, a replenishment order may arrive if it was placed L periods earlier, where L denotes a random lead time. Demand $D_t \in \mathbb{N}$ is then realized. Unmet demand is back-ordered and fulfilled upon receipt of subsequent replenishments. At the end of each period, an order is placed to restore the inventory position to a predetermined base-stock level $S \in \mathbb{N}_0$. Orders are subject to stochastic lead times $L \sim f_L(\cdot)$ that are independent of the per-period demands $D_t \sim f_D(\cdot)$.

Let the cumulative demand over the stochastic lead time be denoted by $D^L = \sum_{t=1}^{L} D_t$, with realizations $d^L \in \mathbb{N}$. The associated distribution $f_{D^L}(\cdot)$ is a random-sum mixture,

$$f_{D^L}(d^L) = \sum_{\ell \in \mathbb{N}} f_L(\ell) f_D^{*\ell}(d^L), \tag{4.1}$$

where $f_D^{*\ell}$ denotes the ℓ-fold convolution of f_D. Inventory costs consist of linear holding and back-order penalty costs. For a given base-stock level S, the long-run expected cost per period is

$$C(S) = h \cdot \mathbb{E}[S - D^L]^+ + b \cdot \mathbb{E}[D^L - S]^+, \tag{4.2}$$

where h denotes the per-unit holding cost per period and b the per-unit back-order penalty cost. The optimal base-stock level S^* minimizes this expected cost:

$$S^* = \arg\min_{S \in \mathbb{N}_0} C(S). \tag{4.3}$$

The distribution $f_{D^L}(\cdot)$ is not directly observable and must be estimated from historical data. Each SKU $j \in J$ is characterized by a limited time series of demand and lead time realizations $\{(d_{j,t}, l_j)\}$, together with a feature vector x_j describing product attributes, supplier characteristics, or usage context. The task is to estimate the conditional distribution

$$f_{D^L|X}(d^L \mid x_j), \tag{4.4}$$

which captures the probability distribution of lead time demand conditional on observable SKU features. These estimates are then used in the Cost Function (4.2) to derive base-stock levels and inventory performance metrics.

In practice the task is difficult. Intermittent demand produces short histories that make parameter estimates in parametric models unstable, while non-parametric methods struggle to capture the distribution tails with so little data. Observable features add further uncertainty because they may or may not reflect actual demand behaviour. Cross-learning across SKUs can be beneficial when features are informative but can degrade performance when they are not. To understand under which conditions each method is appropriate, we construct a simulation experiment with known ground truth.

4.4 Experimental design

This section outlines the design of the simulation experiment. We describe the treatments, how synthetic demand, lead time, and feature data are generated, how the benchmark estimators are implemented, and how performance is measured against the cost-optimal policy.

4.4.1 Treatments

Each treatment combines an arrival-rate level and an alignment level (cf. Table 4.1). At the arrival-rate level, we control the expected number of arrivals over the lead time. We consider very slow moving demand with 0.1 arrivals on average, a medium regime with 0.5 arrivals, a faster regime with one arrival, and a very fast regime with ten arrivals per lead time. These four levels span the range of intermittency observed in the empirical dataset introduced in Chapter 3. For the same reason, we set average lead times to 60 days and the valuation price to $29,000 to match the empirical setting; holding and back-order penalty parameters are introduced in Subsection 4.4.4. We report daily arrival rates by dividing expected arrivals by the average lead time $\lambda \in \{0.0017, 0.0083, 0.0167, 0.1667\}$.

At the alignment level, we control the relationship between feature-based similarity and the true demand structure. We consider perfect alignment, slight misalignment with 10% of SKUs reassigned, moderate misalignment with 25%, high misalignment with 50%, and a random case with 100% reassignment (i.e., no information). The misalignment is implemented by reassigning the specified proportion of SKUs so that they appear closer to other groups, while their demand processes remain unchanged. These five levels cover the range from fully informative to non-informative similarity. Section 4.4.2 describes in detail the data-generating processes and the construction of feature similarity to map these alignment levels. We report alignment by the reassignment fraction $\varphi \in \{0\%, 10\%, 25\%, 50\%, 100\%\}$.

Per treatment we simulate 300 SKUs in three equal-size groups to create controlled heterogeneity in lead time demand patterns for cross-learning. Within each treatment the groups differ only in demand size, with means of 1.5, 10, and 100 units. These levels represent low, medium, and high demand, which creates enough contrast for transfer across SKUs without adding excess complexity to the experiment.

Each treatment spans 1000 years to obtain statistically stable performance measures under strong intermittency. The first four years form the training sample, the next two years are used for validation, and the remaining 994 years constitute the test horizon. Thus, the training and validation set are similar to the real

Table 4.1: Treatments and data regime.

Component	**Symbol / Levels**	**Notes**
Treatment factors		
Daily arrival rate	$\lambda \in \{0.0017, 0.0083, 0.0167, 0.1667\}$	Intermittency control.
Feature demand alignment	$\varphi \in \{0\%, 10\%, 25\%, 50\%, 100\%\}$	Share misassigned; 100% = random.
Fixed settings		
SKUs per treatment	$N = 300$ (3×100)	Groups of different demand-size mean.
Group 1	$N = 100$, demand-size mean 1.5 units	Subgroup within $N = 300$.
Group 2	$N = 100$, demand-size mean 10 units	Subgroup within $N = 300$.
Group 3	$N = 100$, demand-size mean 100 units	Subgroup within $N = 300$.
Lead time	$\mathbb{E}[L] = 60$ days	Used for daily-rate conversion.
Valuation price	\$29,000	Scales holding cost.
Holding cost rate	$h = 0.1 \times$ \$29,000/a	Applied daily.
Back-order penalty	$b \in \{1, \ldots, 1000\}$	Sensitivity; main tables $b = 500$.
Horizon and split	1000 years; train 4 years, val 2 years, test 994 years	Daily periodic-review simulation.

world scenario in Chapter 3. However the large test set enables us to obtain a clear picture of the methods' performance.. A summary is provided in Table 4.1.

4.4.2 Data generating processes

We consider two stylized stochastic processes for generating lead time demand, which together capture the analytical tractability of classical base-stock models and the complexity of real-world spare parts demand.

Compound poisson with logarithmic-series demand sizes. The first process couples demand arrivals N with a poisson process rate λ with a logarithmic distribution of demand sizes Y with parameter $p \in (0, 1)$ and deterministic replenishment lead times of length L. Poisson demand arrivals coupled with logarithmically distributed demand sizes result in a closed-form negative binomial distribution for lead time demand, a canonical assumption in the inventory literature (Quenouille 1949, Babai et al. 2021, Syntetos et al. 2015). This setting allows exact evaluation of the lead time demand distribution $f_{D^L}(\cdot)$. Logarithmic sizes were generated under three experimental conditions: logarithmic parameter $p = 0.5$ ($\mu = 1.5, \sigma = 0.89$), $p = 0.97$ ($\mu = 10, \sigma = 1.41$) and $p = 0.99845$ ($\mu =$

Table 4.2: Data generating processes (DGPs).

Component	**Symbol / Levels**	**Notes**
DGP A	$N_t \sim \text{Poisson}(\lambda)$; $Y \sim \text{LogSeries}(p)$; $L = 60$	Closed-form lead time demand:
Size parameters	$p \in \{0.5, 0.97, 0.99845\}$	$D^L \sim \text{NegBin}$.
DGP B	$N_t \sim \text{Poisson}(\lambda)$; $Y \sim \text{Geom}(q)$; $L \sim \Gamma(k = 3.0, \theta = 20.0); \mathbb{E}[L] = 60$	Panjer recursion
Size parameters	$q \in \{0.40, 0.10, 0.01\}$	
Feature similarity	cluster ID and similarity matrix	Louvain clustering on synthetic similarities; misalignment via reassignment share φ.

$100, \sigma = 233.07)$. In the former case, slow moving demand with mostly unitsized demand is simulated whereas in the latter case demand patterns are lumpy.

Poisson–gamma mixture with geometric demand sizes. The second process is less restrictive and intended to reflect spare parts demand more realistically. Here both lot sizes and lead times vary. Demand arrivals also follow a poisson process with rate λ but with a geometric distribution of demand sizes with success probability q and stochastic replenishment lead times following a gamma distribution with shape parameter k and scale parameter θ.

Unlike the poisson–logarithmic case, this setting does not yield a closed-form distribution for D^L. We can compute the lead time demand distribution using panjer recursion (Panjer 1981). Geometric sizes were generated under three experimental conditions: $q = 0.40$ ($\mu = 1.5, \sigma = 1.94$), $q = 0.10$ ($\mu = 10, \sigma = 9.49$), and $q = 0.01$ ($\mu = 100, \sigma = 99.5$). In the first case, demand is dominated by unit-sized orders, whereas in the last case demand is highly lumpy. Lead times were generated as $L \sim \Gamma(k = 3.0, \theta = 20.0)$, which implies an average lead time of 60 periods consistent with the empirical dataset introduced in Chapter 3.

A summary of the data generating processes is provided in Table 4.2.

Feature similarity and cluster construction. As described in Chapter 2, SKUs can be associated with a feature vector x_j capturing product descriptors or usage attributes. To mimic cross-learning in the synthetic data, we construct a matrix of pairwise similarity scores (cf. Section 2.4.2). These scores are sampled from specified ranges: when two SKUs originate from the same demand group, their similarity is sampled from an interval close to one, whereas when they originate from different demand groups, their similarity is sampled from an interval closer to zero. This similarity matrix serves as input to a clustering algorithm, so that SKUs are assigned to feature-based clusters in addition to their underlying demand pattern.

In the aligned case, the similarity structure ensures that an algorithm such as Louvain (Blondel et al. 2008) can recover the true demand groups. In our treatments misalignment φ is introduced by reassigning a fixed share of SKUs so that their similarity scores are shifted toward other groups, while their demand process remains unchanged. In the slight, moderate, and high treatments, 10%, 25%, and 50% of SKUs are misassigned in this way. These SKUs are therefore connected more strongly in feature space to SKUs with different demand behaviour, which induces feature-based methods to transfer information across heterogeneous groups. This reflects the practical concern that observable features do not always align with demand. In the random case ($\varphi = 100\%$), all similarity scores are drawn independently of demand, so clustering provides no information about the underlying structure.

4.4.3 Benchmarked methods

Parametric approach. We forecast demand using Croston's method (Croston 1972) with the Syntetos–Boylan adjustment (SBA) (Syntetos and Boylan 2005) and the Teunter–Syntetos–Babai (TSB) formulation (Teunter et al. 2011). Both are widely used in the intermittent demand literature and are regarded as reference methods for spare parts forecasting, as they employ recursive exponential smoothing updates and correct the bias and variance problems that arise under sparsity (Hasni et al. 2019, Teunter and Duncan 2009).

Following established practice, smoothing constants are kept small to reflect the assumed stationarity of the process. For SBA we select values of $\alpha \in [0.05, 0.3]$ for both demand sizes and inter-arrival times, consistent with prior recommendations (Syntetos and Boylan 2001, Eaves and Kingsman 2004, Teunter and Duncan 2009). Following Teunter et al. (2011), in TSB we set the smoothing constant for demand-occurrence probability smaller than that for demand size and tune $\beta \in [0.01, 0.05]$. Candidate values are applied uniformly across SKUs within a treatment, and the best configuration is selected using a grid search with step size 0.05 for α and 0.01 for β with MSE as selection criterion. For demand we employ a rolling one-step-ahead scheme across the entire validation horizon, so that each day's demand is predicted using only past data.

Lead times, which are observed only at receipt events, are modeled independently of demand and smoothed with SES at each receipt, with the forecast held constant in between. The combination of smoothed means and error-based variance estimates from demand and lead time provides the two moments of lead time demand. These moments are then matched to a negative binomial distribution, yielding a closed-form predictive distribution (Zipkin 2000).

Bootstrapping. We apply a temporal bootstrap independently to each SKU. From the observed demand history and empirical lead time realizations, pseudo-samples of lead time demand are generated by repeatedly drawing a random calendar date and pairing it with a lead time sampled with replacement. The corresponding demand over the sampled horizon is aggregated to form a synthetic realization. Repeating this procedure $B = 10,000$ times yields an empirical distribution of lead time demand, which serves directly as the estimator.

The method reproduces the empirical shape of the distribution, capturing skewness and lumpiness without parametric assumptions. Unlike the two-state Markov scheme of Willemain et al. (2004), the temporal bootstrap accommodates stochastic lead times without auxiliary assumptions, consistent with the arguments of Teunter and Duncan (2009). With short demand histories the resulting distribution may be discrete and irregular, since it is limited to linear combinations of the observed demand sizes. As the history lengthens and the number of bootstrap replications increases the distribution stabilizes, but under strong intermittency it remains sensitive to sampling variability unless further structure is imposed (Willemain et al. 2004, Boylan and Babai 2022).

Distributional random forest. The estimator we propose in Chapter 3 extends the bootstrap procedure by exploiting feature-based similarity across SKUs. The procedure first constructs a pseudo-sample of lead time demands using the bootstrap scheme described above. Each realization is paired with the feature vector of its SKU. Features include a one hot encoding of the cluster identifier to enable cross learning across similar SKUs and an encoded part identifier to preserve part identity. This lets the random forest place weight on the SKUs own pseudo samples when they exist while still drawing on information from feature-similar SKUs. No temporal features are used since the design is stationary. This data is then used to fit an ensemble of decision trees. At prediction time, the random forest identifies a neighborhood of training observations through terminal-node assignments. This yields a discrete, data-driven approximation of the lead time demand distribution for an SKU with features, combining local evidence from its own history with cross-SKU information.

Hyperparameters are tuned by time-series cross-validation. Candidate settings are generated by Latin hypercube sampling of a maximum entropy grid. The search space includes the number of trees ($\text{n\_trees} \in \{50, 100, 200, 300\}$), the maximum

Table 4.3: Estimators and tuning.

Estimator	**Main setting**	**Notes**
Parametric (Croston SBA; TSB)	$\alpha \in [0.05, 0.30]$ $\beta \in \{0.01, \dots, 0.05\}$	Rolling one-step CV. SES for mean lead time. Moment match to NegBin for f_{D^L}.
Bootstrap (per-SKU temporal)	$B = 10{,}000$	Random start day with resampled lead time. Aggregate over horizon. Empirical f_{D^L}.
Distributional Random Forest	n_trees $\in \{50, 100, 200, 300\}$ max_depth $\in \{5, 10, 20, \text{None}\}$ min_leaf $\in \{5, 10, 25, 50, 100\}$ $B \in \{10, 100, 200, 300\}$	Five-fold expanding-window CV. Cluster threshold in $[0.5, 0.9]$ and minimum cluster size in $[1, 1000]$. Features: cluster ID and encoded part ID.

depth of the trees (max_depth $\in \{5, 10, 20, \text{None}\}$), the minimum number of samples per leaf (min_leaf $\in \{5, 10, 25, 50, 100\}$), and the number of bootstrap samples ($B \in \{10, 100, 200, 300\}$). Clustering parameters are also tuned, with similarity thresholds between 0.5 and 0.9 and minimum cluster sizes between 1 and 300. Cross-validation here follows a five-fold expanding-window scheme: in each fold the training sample is extended forward in time and the next fixed block of days is used for validation (Hyndman 2014).

The tuning procedure allows the model to decide whether to exploit cross–sectional information or to rely on a part's own history. When cluster identifiers carry predictive value the random forest may assign them high importance and draws strength from neighboring SKUs. When they do not, the model may reduce their weight and falls back on the individual identifier. In this way the search over alternative hyperparameters allows that similarity is exploited when it improves accuracy and that negative transfer can be limited when cluster information is weak. A summary of the estimators and their main tuning parameters is provided in Table 4.3.

4.4.4 Evaluation procedure

To assess the competing estimators we follow a practice that is well established in the intermittent demand literature (e.g., Syntetos and Boylan 2008, Teunter and Duncan 2009, Babai et al. 2022, Teunter et al. 2011, Syntetos et al. 2015), namely to evaluate predictive distributions through their induced inventory performance rather than through purely statistical accuracy metrics. This choice is consistent with our setting where the decision maker's objective is to minimize expected inventory cost in the base-stock system described in Section 4.3.

The evaluation proceeds in two stages. In the first stage, each estimator produces a predictive distribution of lead time demand for every SKU. Given this discrete distribution we compute the expected holding and back-order costs for each candidate base-stock level by applying the same cost function as in Section 4.3. The base-stock level is then chosen as the minimizer of expected total cost. Holding costs are parameterized as ten percent of the part valuation price per year, scaled to the daily simulation horizon. Back-order penalty costs are treated as an exogenous parameter and are varied across a wide grid from 1 to 1000 to assess sensitivity.

In the second stage, these base-stock levels are applied in a discrete-event simulation of the inventory system. The evaluation excludes a warm-up period (train and evaluation sets as described in Section 4.4.2) to remove initialization effects, after which inventory trajectories are recorded. From these trajectories we compute average holding and back-order costs, the corresponding service levels, and the realized total cost.

Because the synthetic environment provides the true distribution of lead time demand, the cost-optimal policy can be derived exactly by solving the same optimization problem with the true distribution rather than an estimated one. This policy provides a benchmark against which realized performance is evaluated.

Reported results include absolute cost levels as well as relative efficiency defined as the ratio of realized cost to the true optimal cost.

4.5 Results

This section presents the empirical results. We evaluate the inventory performance of the three estimators under varying data regimes and alignment conditions. Performance is reported as average total cost and its deviation from the theoretical optimum. Two sets of experiments are analyzed: the poisson–gamma mixture with geometric lot sizes and stochastic lead times, and the compound poisson with logarithmic-series lot sizes implying a negative binomial distribution (cf. Tables 4.4 and 4.5). Rows vary the daily arrival rate. Row blocks vary the degree of feature–demand alignment. Within each cell the first number is cost in \$100,000 and the second is the percent deviation from optimal cost.

For the poisson–gamma mixture with geometric lot sizes and stochastic lead times (Table 4.4), in the aligned case with very slow arrivals at $\lambda = 0.0017$, the parametric estimator yields a cost of \$465,000 which is 54% above optimal, while the bootstrap yields \$325,000 at 7.5% above optimal. The distributional random forest attains \$307,000, almost indistinguishable from the optimum at only 1.7% deviation. Reading across the row, this demonstrates the relative inefficiency of the parametric method under sparsity and the improvement gained through cross-learning.

Increasing arrival rates reduce all deviations. When $\lambda = 0.1667$, costs across methods are close to the optimum, with deviations below 12% even for the parametric estimator. These results align with literature on data-rich settings, which support the use of parametric estimation. Across misalignment levels we observe how the random forest gradually loses its advantage but never deteriorates

Table 4.4: Total cost (in $100,000) and deviation from the optimum at $b = 500$ by misalignment level φ and daily arrival rate λ under poisson–gamma mixture with geometric demand sizes.

		Methods		
φ	λ	$S^*_{\text{par}}(b)$	$S^*_{\text{boot}}(b)$	$S^*_{\text{drf}}(b)$
0%	0.0017	(4.65, 54.0%)	(3.25, 7.5%)	(3.07, 1.7%)
	0.0083	(6.36, 33.4%)	(5.29, 10.9%)	(4.67, -2.1%)
	0.0167	(7.74, 31.1%)	(6.29, 6.6%)	(6.04, 2.2%)
	0.1667	(20.83, 11.5%)	(18.89, 1.2%)	(19.07, 2.2%)
10%	0.0017	(4.65, 54.0%)	(3.25, 7.5%)	(3.18, 5.3%)
	0.0083	(6.36, 33.4%)	(5.29, 10.9%)	(5.06, 6.1%)
	0.0167	(7.74, 31.1%)	(6.29, 6.6%)	(6.11, 3.5%)
	0.1667	(20.83, 11.5%)	(18.89, 1.2%)	(18.75, 0.4%)
25%	0.0017	(4.65, 54.0%)	(3.25, 7.5%)	(3.28, 8.7%)
	0.0083	(6.36, 33.4%)	(5.29, 10.9%)	(5.62, 17.9%)
	0.0167	(7.74, 31.1%)	(6.29, 6.6%)	(6.18, 4.6%)
	0.1667	(20.83, 11.5%)	(18.89, 1.2%)	(18.75, 0.4%)
50%	0.0017	(4.65, 54.0%)	(3.25, 7.5%)	(3.28, 8.6%)
	0.0083	(6.36, 33.4%)	(5.29, 10.9%)	(5.28, 10.7%)
	0.0167	(7.74, 31.1%)	(6.29, 6.6%)	(6.30, 6.7%)
	0.1667	(20.83, 11.5%)	(18.89, 1.2%)	(18.75, 0.4%)
100%	0.0017	(4.65, 54.0%)	(3.25, 7.5%)	(3.32, 9.8%)
	0.0083	(6.36, 33.4%)	(5.29, 10.9%)	(5.40, 13.2%)
	0.0167	(7.74, 31.1%)	(6.29, 6.6%)	(6.30, 6.7%)
	0.1667	(20.83, 11.5%)	(18.89, 1.2%)	(18.75, 0.4%)

Note: The entry for $\varphi = 0\%$ and $\lambda = 0.0083$ reports a deviation of -2.1% for $S^*_{\text{drf}}(b)$. This does not indicate performance better than the true optimum. The benchmark cost is computed with panjer recursion with discretization and truncation, so it only approximates the lead time demand distribution and can shift the computed optimum slightly. At $\lambda = 0.0083$ demand is still very sparse, so simulation variance is material. Together these effects can produce small negative deviations for some b. Figure 4.3 shows the simulation results over the full range of b and demonstrates that such cases are rare and close to zero. We therefore treat -2.1% as essentially zero and indistinguishable from the optimum.

below bootstrap. Even under 100% randomization the method yields cost levels similar to bootstrapping, showing that the random forest reverts to local estimation in the absence of informative features.

Turning to the compound poisson with logarithmic-series sizes in Table 4.5, the qualitative ranking is the same but contrasts are more pronounced. Under full alignment the random forest is essentially optimal across all arrival rates, with deviations between 0.0% and 0.4%, whereas the bootstrap remains 15 to 27% above the benchmark even at $\lambda = 0.1667$ and much higher at very low λ. The parametric estimator is again far from optimal at very low λ. Unlike in the poisson–gamma mixture, these gains persist at higher arrival rates. This reflects the absence of lead time uncertainty and the tractable negative binomial target, which make

Table 4.5: Total cost (in $100,000) and deviation from the optimum at $b = 500$ by misalignment level φ and daily arrival rate λ under compound poisson with logarithmic-series demand sizes.

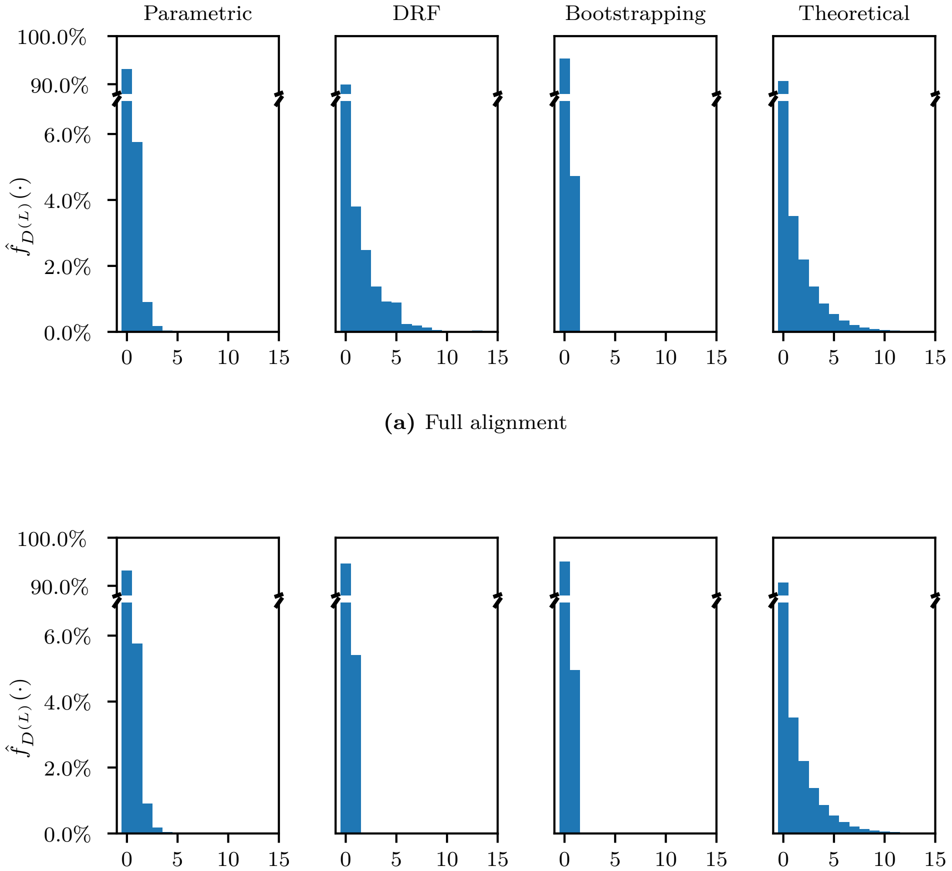
		Methods		
φ	λ	$S^*_{\text{par}}(b)$	$S^*_{\text{boot}}(b)$	$S^*_{\text{drf}}(b)$
0%	0.0017	(0.50, 281.7%)	(0.16, 24.7%)	(0.13, 0.0%)
	0.0083	(0.38, 48.1%)	(0.33, 27.3%)	(0.26, 0.1%)
	0.0167	(0.49, 51.3%)	(0.40, 24.4%)	(0.32, 0.0%)
	0.1667	(0.97, 30.8%)	(0.85, 15.3%)	(0.74, 0.4%)
10%	0.0017	(0.50, 281.7%)	(0.16, 24.7%)	(0.14, 6.1%)
	0.0083	(0.38, 48.1%)	(0.33, 27.3%)	(0.29, 14.6%)
	0.0167	(0.49, 51.3%)	(0.40, 24.4%)	(0.36, 9.7%)
	0.1667	(0.97, 30.8%)	(0.85, 15.3%)	(0.85, 15.7%)
25%	0.0017	(0.50, 281.7%)	(0.16, 24.7%)	(0.15, 12.8%)
	0.0083	(0.38, 48.1%)	(0.33, 27.3%)	(0.33, 27.9%)
	0.0167	(0.49, 51.3%)	(0.40, 24.4%)	(0.40, 24.8%)
	0.1667	(0.97, 30.8%)	(0.85, 15.3%)	(0.85, 15.6%)
50%	0.0017	(0.50, 281.7%)	(0.16, 24.7%)	(0.16, 17.9%)
	0.0083	(0.38, 48.1%)	(0.33, 27.3%)	(0.33, 27.6%)
	0.0167	(0.49, 51.3%)	(0.40, 24.4%)	(0.40, 23.4%)
	0.1667	(0.97, 30.8%)	(0.85, 15.3%)	(0.85, 15.7%)
100%	0.0017	(0.50, 281.7%)	(0.16, 24.7%)	(0.16, 24.0%)
	0.0083	(0.38, 48.1%)	(0.33, 27.3%)	(0.33, 29.3%)
	0.0167	(0.49, 51.3%)	(0.40, 24.4%)	(0.40, 24.8%)
	0.1667	(0.97, 30.8%)	(0.85, 15.3%)	(0.85, 15.7%)

Notes: Misalignment (φ) shown as the share of parts randomly reassigned across clusters. The daily arrival rate is computed from arrivals per lead time divided by the 60-day lead time representing 0.1, 0.5, 1, and 10 arrivals per lead time. Each cell reports (total cost in $100,000, deviation in % vs. theoretical optimum).

cross-SKU pooling more effective and expose tail noise in per-SKU resampling. As misalignment increases, the random forest converges toward the bootstrap and from 25% to 100% misassignment they are essentially indistinguishable across arrival rates. The random forest does not fall below the bootstrap by more than a few percentage points, consistent with tree-based localization. Sensitivity to misalignment is larger than in the poisson–gamma mixture, with deviations reaching 24 to 29% at $\lambda \in \{0.0083, 0.0167\}$ under full randomization, compared to mostly single-digit deviations in Table 4.4. Overall, the compound poisson results confirm the value of cross-learning when features align and show a faster loss of advantage under misalignment.

The estimated distributions in Figure 4.1 illustrate these dynamics. When feature alignment is high, the random forest produces predictive distributions that closely

Figure 4.1: Example of estimations for different methods for $\lambda = 0.0017$ and geometric demand sizes with $q = 0.01$.



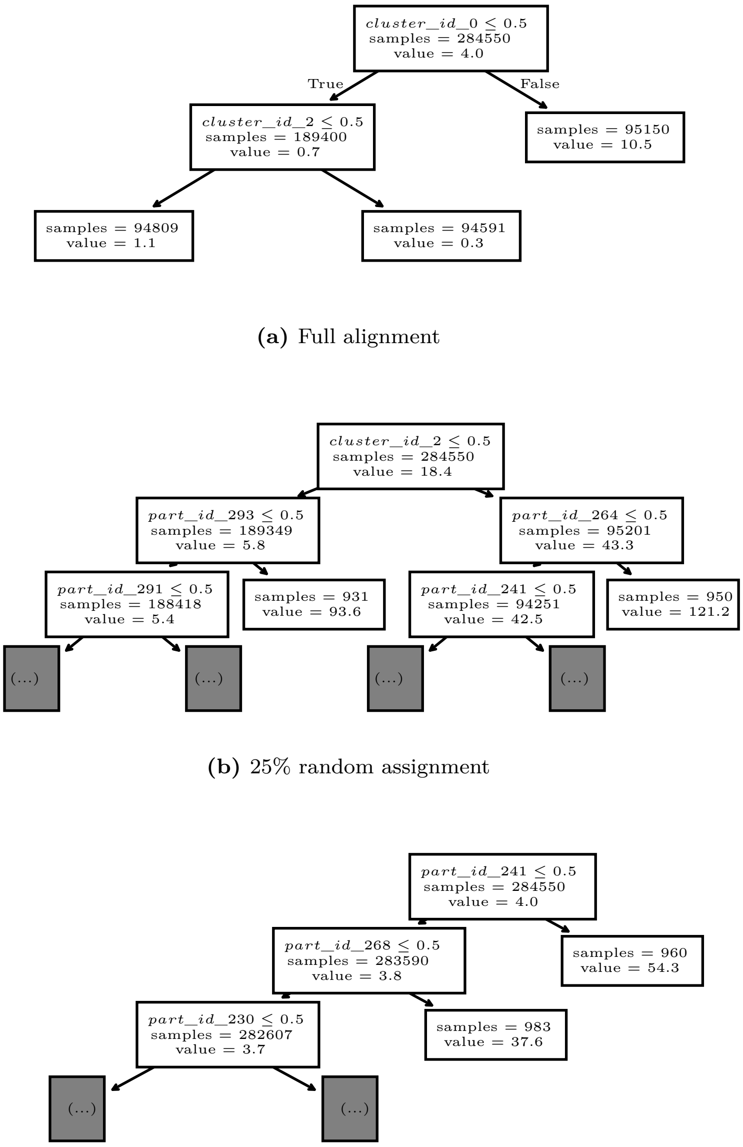
(a) Full alignment

(b) Random assignment

replicate the true theoretical distribution we sampled the data from. This is visible in the smooth, symmetric shapes resembling the true negative binomial or panjer recursed distributions, in contrast to the irregularity of the bootstrap which reflects sampling noise. Under misalignment the random forest distributions become less regular but remain centered around the correct mean, whereas the bootstrap increasingly exhibits spikes and mass shifts due to data sparsity. The parametric estimator appears smoother but systematically biased, with mass displaced toward the center and underweighting of the tails, explaining its persistent cost penalties.

The decision rules of one tree from the random forest further illustrate the mechanism in Figure 4.2 (showing the first three levels). Under perfect alignment

Figure 4.2: Example of tree structure under different alignment levels with $\lambda = 0.0017$ and geometric demand sizes.



(a) Full alignment

(b) 25% random assignment

(c) Random assignment

the random forest exploits cluster identifiers to partition SKUs cleanly by their true demand group, thereby transferring information across genuinely similar SKUs. With partial misalignment, splits show both cluster features and part identifiers being used, indicating that the algorithm learns across SKUs with local correction. Under random alignment the random forest relies almost exclusively on individual identifiers, effectively collapsing to a local estimator. This adaptive weighting of cross-sectional information explains why the random forest captures gains when available and avoids severe losses when similarity misleads.

4.6 Conclusions

We studied cross-learning for estimating lead time demand distributions in spare parts inventory and assessed its value under controlled variation in data sparsity and alignment between observable features and true demand behavior. Using a simulation with compound poisson and poisson–gamma mixture demand, stochastic lead times, and long test horizons, we compared a distributional random forest to a per-SKU temporal bootstrap and a per-SKU parametric negative binomial approach. The results show that when features align with demand, cross-learning yields material cost reductions, particularly for slow movers where within-SKU histories are short. When features mislead, performance deterioration is limited relative to non-parametric bootstrapping because tree-based localization shifts weight back to the focal SKU and curbs negative transfer. Parametric estimation performs well only when data are abundant and assumptions match the data generating process; under sparsity or misspecification it remains far from the inventory optimum.

These findings provide practical guidance. Cross-learning is most useful when part attributes or usage context carry information about demand and when intermittent arrivals constrain local estimation. In such settings, distributional machine learning can recover stable lead time demand distributions and improve stock decisions without strong distributional assumptions. When alignment is uncertain or weak, local estimators remain competitive and the proposed method reverts toward them, preserving robustness. The study also clarifies that evaluation through realized inventory costs is essential, since small distributional improvements at the tails can translate into meaningful changes in holding and back-order costs.

The work has limitations. The modeling is deliberately stylized to isolate the mechanics of cross-learning. We assume a single-echelon, periodic-review base-stock system with stationary, independent demand and lead time, a convex separable cost structure, and full backlogging. The learning problem is framed with clean feature representations, fixed similarity metrics, and controlled alignment noise, a condition that is unlikely to hold in practice. The simulation uses synthetic data from known distributions, which do not capture all aspects of real-world demand. The evaluation focuses on average cost performance, without analyzing service level distributions or other risk measures. These reductions make the dynamics tractable and the claims precise, but they narrow external validity to settings with stable structure and well-behaved features.

Future research can extend to nonstationary and hierarchical settings, incorporate causal or experimental features derived from usage telemetry, and study joint learning of demand and lead time. The empirical study in Chapter 3 already demonstrates performance on one firm's industrial data within a cost-based inventory evaluation. Future work could extend this work to further external replication. A natural next step is to re-run the same evaluation on two established contexts: the royal air force dataset, the jewellery dataset and the automotive aftermarket dataset used in intermittent-demand literature (Teunter and Duncan 2009, Teunter et al. 2010, Hasni et al. 2019, Babai et al. 2022).

Overall, the results support cross-learning as a principled and robust approach for distributional estimation in spare parts inventory, with clear benefits when informative similarity exists and tempered risks when it does not.

Appendix of Chapter 4

4.A Full range of back-order penalty cost

Figure 4.3: Details of simulation for $\lambda = 0.0017$ and geometric demand sizes.

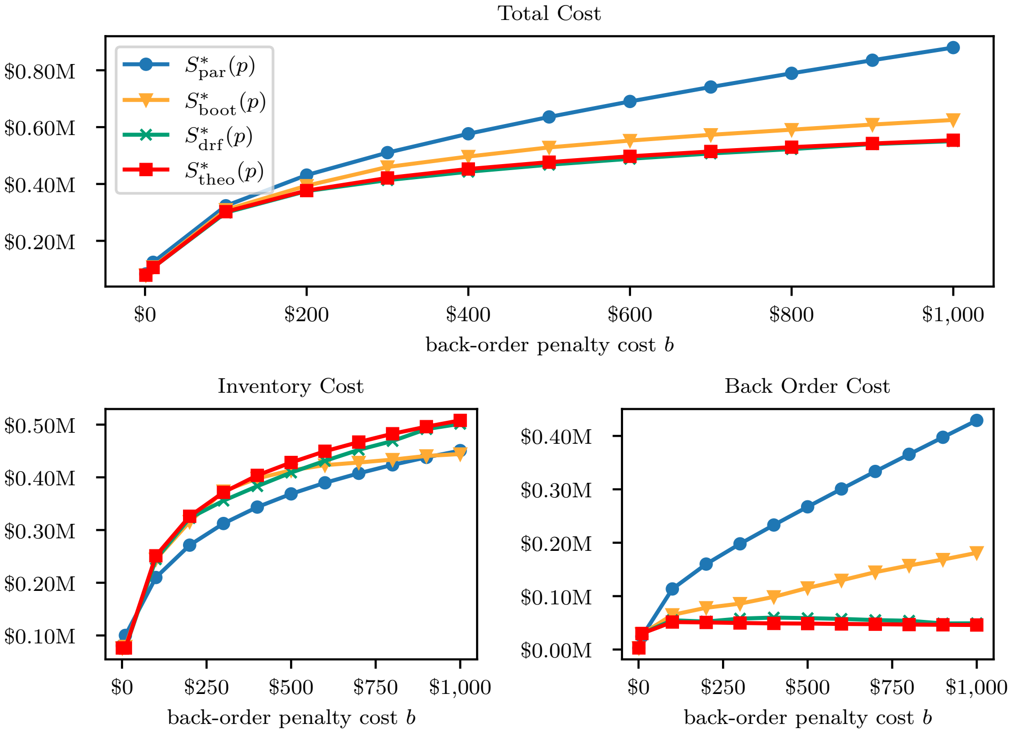
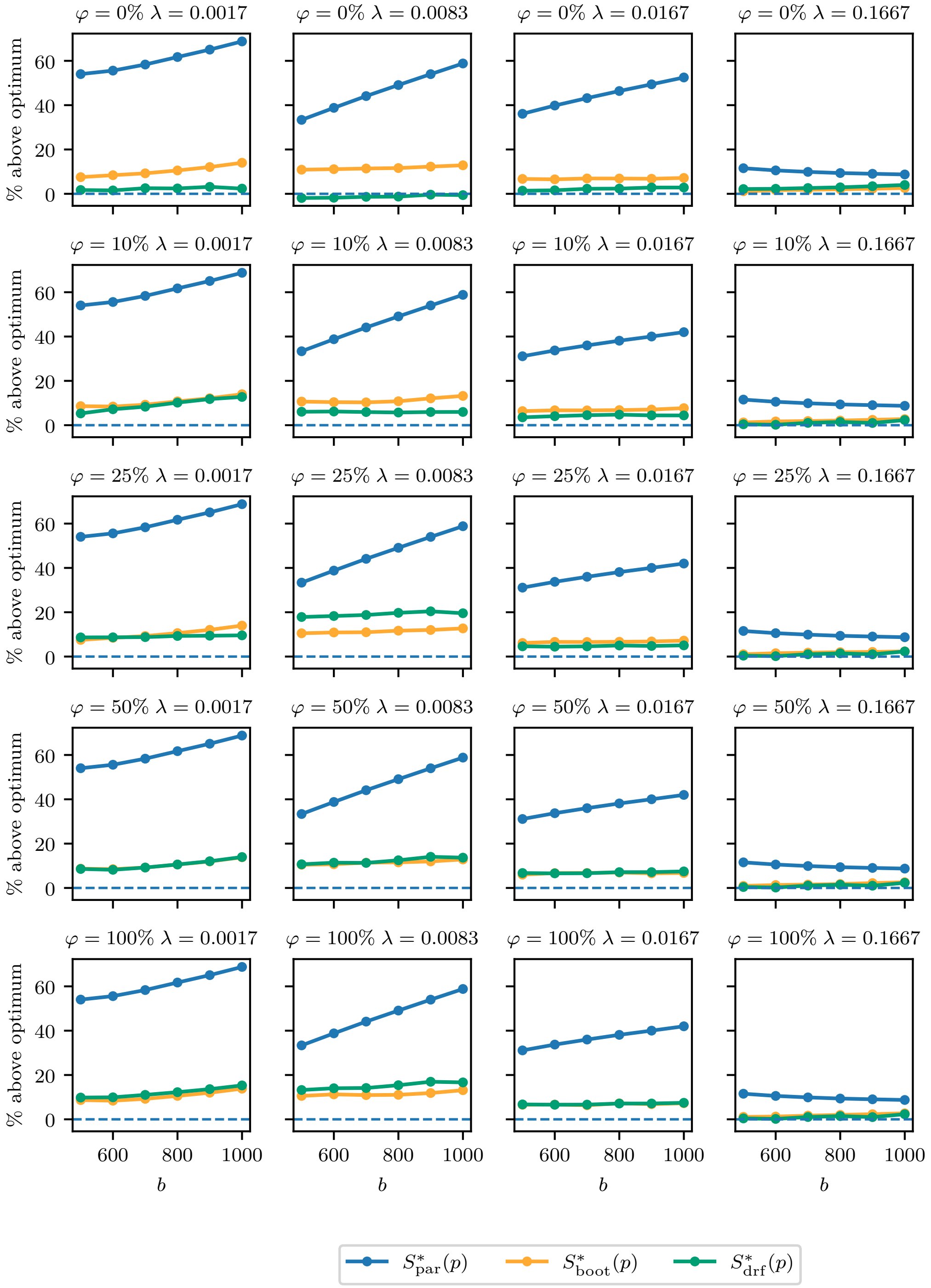


Figure 4.4: Details of deviation grid for poisson-gamma mixture.



Chapter 5

Conclusion

This dissertation examined how data-driven methods can improve spare-parts planning when demand is intermittent, lead times are volatile, and histories are sparse. Across three studies, it addressed the accuracy of plan lead times, the estimation of full lead time demand distributions, and the conditions under which cross-learning across parts is beneficial. The results show that better information improves inventory outcomes and that the value of specific methods hinges on data availability, tail risk, and the informativeness of observable similarity.

5.1 Summary of key results

The results of this thesis provide three main recommendations when putting inventory research to practice: Consider the data, consider the distribution, not only the point and consider the boundaries.

Consider the data. Chapter 2 showed that plan lead times derived with supervised learning from purchase-order and master data are materially more accurate than static ERP values and simple time-series baselines. In our case company, machine-learning regressors improved MSE by over 30% versus ERP entries. The gains persist across frequency segments and are especially relevant for infrequently purchased items and new SKUs. In a simulation study, replacing ERP plan lead times with learned values reduced capital lock-up while maintaining a 95% service target, with an estimated 7% reduction in required inventory investment for that service level. Feature engineering, the use of categorical encoders and text-based SKU similarity were decisive. These results demonstrate that firms can raise planning quality with the data they already collect.

Consider the distribution, not only the point. Chapter 3 moved from point forecasts to full predictive distributions of lead time demand. A distributional random forest that blends temporal bootstrapping with cross-sectional learning produced lower total cost than both a per-SKU bootstrap and a classical parametric benchmark, with the largest advantages when back-order penalties emphasize tail risk. Cost decompositions explain the pattern. The distributional random forest holds slightly more inventory than the bootstrap but cuts back-order penalties substantially by allocating more probability mass to extreme outcomes that standard bootstraps understate when histories are short. As data become abundant, differences narrow and the non-parametric methods converge.

Consider the boundaries. Chapter 4 tested when cross-learning helps for distributional forecasting. In controlled experiments, cross-learning via a distributional random forest delivered material cost reductions when observable features aligned with true demand similarity, and showed limited downside when similarity signals were weak. (With weak alignment the forest naturally reverts toward local estimation and tracks the bootstrap; with strong alignment it uses pooling to smooth tails). As histories grow, the estimates from all methods converge and the inventory cost curve flattens near its optimum. Parametric estimation is weak in sparse data but becomes competitive once arrivals are frequent. The upshot is a practical rule to employ cross-learning when informative similarity exists and expect neutral outcomes otherwise.

Jointly, the chapters connect improved inputs to realized outcomes: more accurate plan lead times lower capital-lockup for a given service level; distributional forecasting cuts penalty exposure without excessive stock; and cross-learning offers a robust pathway through sparsity. Together they form an implementable pipeline that starts from raw operational data and ends in improved inventory decisions.

5.2 Critical review and future research

In Chapter 2 the evidence shows that machine learning can correct inaccurate plan lead times using information already present in operational systems, yet the external validity of the specific model choice is limited. Random forests performed well because they tolerate skewness and capture nonlinearity, but no single algorithm will dominate across all supply settings. The decisive lever is careful feature engineering that embeds domain knowledge, including text based similarity and supplier performance signals. The study focused on point estimates that feed existing planning tools and did not quantify uncertainty, although inventory optimization often requires dispersion measures. The proposed path forward is to extend the same data pipeline to conditional density estimation so that full predictive distributions of lead time or lead time demand can be used directly. Continuous improvement efforts on problematic parts and suppliers can complement modeling by stabilizing inputs and creating routines that absorb residual uncertainty. Together these directions would connect the demonstrated gains in accuracy to decision models that account for variability rather than only central tendency.

In Chapter 3 the move from point forecasts to distributional estimation improves cost outcomes under sparse histories, but the design leaves room for broader scope. The distributional random forest with temporal bootstrapping exploits cross learning to sharpen tail behavior, which is where service penalties concentrate. The method adapts as data accumulate and converges toward standard non-parametric baselines, which supports robust deployment. Open issues include the range of distributional learners that could be substituted without strong assumptions, the role of external covariates such as macro conditions and supplier metrics, and the extension from single location control to multi echelon settings where upstream and downstream nodes interact. Another frontier is adaptive updating that refreshes predictive distributions as new transactions arrive, turning the approach into a live component of planning rather than a periodic batch procedure.

In Chapter 4 controlled simulations inform when cross learning adds value and when it should defer to local estimators, but the abstraction that makes these mechanisms transparent also narrows generalizability. The setting assumes a single echelon system with periodic review, stationary and independent demand and lead time, convex separable costs, and full backlogging. Feature representations are clean and similarity is governed by exogenous alignment noise, while data are generated from known processes. We evaluate models by average cost under a back-order penalty. We do not model fill rate or service distributions, which limits direct transferability to practice. The most direct next steps are to relax stationarity, to model hierarchical and networked structures, and to incorporate causal features derived from usage telemetry that carry information about demand formation. Joint estimation of demand and lead time would align with practice where both processes evolve together. Replication on established intermittent demand datasets such as the royal air force, jewellery, and automotive aftermarket collections would further test robustness and aid comparison across studies.

Taken together the chapters indicate a staged research agenda that deepens modeling of uncertainty, widens operational scope, and broadens empirical grounding. Chapter 2 motivates richer outputs for lead time by moving from points to distributions. Chapter 3 introduces a distributional learner that is effective under sparsity and can leverage on external signals. Chapter 4 specifies the conditions under which cross learning is reliable and sets out the relaxations and replications needed for wider adoption. The result is a coherent roadmap from data inputs to informed decisions that remain stable when data are thin, tails matter, and observable similarity is only partially informative.

References

Aronis, K. P., Magou, I., Dekker, R., and Tagaras, G. (2004). Inventory control of spare parts using a bayesian approach: A case study. *European Journal of Operational Research*, 154:730–739.

Athey, S. and Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, 11(1):685–725.

Axsäter, S. (2015). *Inventory Control*, volume 225. Springer International Publishing, 3 edition.

Babai, M. Z., Chen, H., Syntetos, A. A., and Lengu, D. (2021). A compound-poisson bayesian approach for spare parts inventory forecasting. *International Journal of Production Economics*, 232:107954.

Babai, M. Z., Dai, Y., Li, Q., Syntetos, A. A., and Wang, X. (2022). Forecasting of lead-time demand variance: Implications for safety stock calculations. *European Journal of Operational Research*, 296(3):846–861.

Ban, G. Y. and Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67:90–108.

Banerjee, A. G., Yund, W., Yang, D., Koudal, P., Carbone, J., and Salvo, J. (2015). A hybrid statistical method for accurate prediction of supplier delivery times of aircraft engine parts. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, page V01BT02A037, Boston, MA, USA. American Society of Mechanical Engineers.

Barbour, W., Samal, C., Kuppa, S., Dubey, A., and Work, D. B. (2018). On the data-driven prediction of arrival times for freight trains on U.S. railroads. In *21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2289–2296.

Bender, J. and Ovtcharova, J. (2021). Prototyping machine-learning-supported lead time prediction using autoML. In *Procedia Computer Science*, volume 180, pages 649–655.

Berlec, T., Govekar, E., Grum, J., Potočnik, P., and Starbek, M. (2008). Predicting order lead times. *Strojniški vestnik – Journal of Mechanical Engineering*, 54(5):308–321.

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Bookbinder, J. H. and Lordahl, A. E. (1989). Estimation of inventory re-order levels using the bootstrap statistical procedure. *IIE Transactions*, 21:302–312.

Boylan, J. E. and Babai, M. Z. (2022). Estimating the cumulative distribution function of lead-time demand using bootstrapping with and without replacement. *International Journal of Production Economics*, 252:108586.

Boylan, J. E. and Syntetos, A. A. (2010). Spare parts management: a review of forecasting research and extensions. *IMA Journal of Management Mathematics*, 21(3):227–237.

Branco, P., Torgo, L., and Ribeiro, R. P. (2017). Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, volume 74, pages 36–50. PMLR.

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

Brownlee, J. (2020). *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python.* Machine Learning Mastery.

Brynjolfsson, E. and McAfee, A. (2017). Artificial intelligence, for real. *Harvard Business Review*, 2017:1–30.

Burggräf, P., Wagner, J., Koke, B., and Steinberg, F. (2020). Approaches for the prediction of lead times in an engineer to order environment: A systematic review. *IEEE Access*, 8:142434–142445.

Cachon, G. and Terwiesch, C. (2008). *Matching supply with demand: An introduction to operations management.* McGraw-Hill, New York, NY,USA.

Cao, Y. and Shen, Z. J. M. (2019). Quantile forecasting and data-driven inventory management under nonstationary demand. *Operations Research Letters*, 47:465–472.

Carling, K. (2000). Resistant outlier rules and the non-gaussian case. *Computational Statistics & Data Analysis*, 33(3):249–258.

Cavalieri, S., Garetti, M., Macchi, M., and Pinto, R. (2008). A decision-making framework for managing maintenance spare parts. *Production Planning and Control*, 19(4):379–396.

Choi, S., Kim, Y. J., Briceno, S., and Mavris, D., editors (2016). *Prediction of weather-induced airline delays based on machine learning algorithms.* IEEE.

Chopra, S., Reinhardt, G., and Dada, M. (2004). The effect of lead time uncertainty on safety stocks. *Decision Sciences*, 35(1):1–24.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Operational Research Quarterly*, 23:289–303.

Dalmasso, N., Pospisil, T., Lee, A. B., Izbicki, R., Freeman, P. E., and Malz, A. I. (2020). Conditional density estimation tools in python and R with applications to photometric redshifts and likelihood-free cosmological inference. *Astronomy and Computing*, 30(1):100362.

de Oliveira, M. B., Zucchi, G., Lippi, M., Cordeiro, D., Da Silver, N. R., and Iori, M. (2021). Lead time forecasting with machine learning techniques for a pharmaceutical supply chain. In *Proceedings of the 23rd International Conference on Enterprise Information Systems*, volume 1, pages 634–641. SCITEPRESS - Science and Technology Publications.

Dolgui, A., Ammar, O. B., Hnaien, F., and Louly, M. (2013). A state of the art on supply planning and inventory control under lead time uncertainty. *Studies in Informatics and Control*, 22(3):255–268.

Duffie, N., Bendul, J., and Knollmann, M. (2017). An analytical approach to improving due-date and lead-time dynamics in production systems. *Journal of Manufacturing Systems*, 45:273–285.

Eaves, A. H. and Kingsman, B. G. (2004). Forecasting for the ordering and stock-holding of spare parts. *Journal of the Operational Research Society*, 55:431–437.

Efron, B. (1992). *Bootstrap Methods: Another Look at the Jackknife*, pages 569–593. Springer New York, New York, NY.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54–75.

Eppen, G. D. and Martin, R. K. (1988). Determining safety stock in the presence of stochastic lead time and demand. *https://doi.org/10.1287/mnsc.34.11.1380*, 34:1380–1390.

Escudero, L. F., Galindo, E., Garcıia, G., Gomez, E., and Sabau, V. (1999). Schumann, a modeling framework for supply chain management under uncertainty. *European Journal of Operational Research*, 119(1):14–34.

Fetter, R. B. and Dalleck, W. C. (1961). *Decision models for inventory management*. R. D. Irwin.

Fortuin, L. (1980). Five popular probability density functions: A comparison in the field of stock control models. *Journal of the Operational Research Society*, 31(10):937–942.

Fricker, R. D. and Goodhart, C. A. (2000). Applying a bootstrap approach for setting reorder points in military supply systems. *Naval Research Logistics (NRL)*, 47:459–478.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Grout, J. R. and Christy, D. P. (1993). An inventory model of incentives for on–time delivery in just–in–time purchasing contracts. *Naval Research Logistics*, 40(6):863–877.

Gupta, H. and Asha, V. (2020). Impact of encoding of high cardinality categorical data to solve prediction problems. *Journal of Computational and Theoretical Nanoscience*, 17(9-10):4197–4201.

Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W., and Monostori, L. (2018). Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *International Federation of Automatic Control*, 51(11):1029–1034.

Hadley, G. and Whitin, T. M. (1963). *Analysis of inventory systems.* Prentice-Hall. Prentice-Hall quantitative methods series. Includes bibliography.

Hall, P. (1992). On bootstrap confidence intervals in nonparametric regression. *The Annals of Statistics*, 20(2):695–711.

Hasni, M., Babai, M. Z., Aguir, M. S., and Jemai, Z. (2019). An investigation on bootstrapping forecasting methods for intermittent demands. *International Journal of Production Economics*, 209:20–29.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hofleitner, A., Herring, R., and Bayen, A. (2012). Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. *Transportation Research Part B: Methodological*, 46:1097–1122.

Huber, J., Müller, S., Fleischmann, M., and Stuckenschmidt, H. (2019). A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, 278:904–915.

Huh, W. T., Levi, R., Rusmevichientong, P., and Orlin, J. B. (2011). Adaptive data-driven inventory control with censored demand based on kaplan-meier estimator. *https://doi.org/10.1287/opre.1100.0906*, 59:929–941.

Huiskonen, J. (2001). Maintenance spare parts logistics: Special characteristics and strategic choices. *International Journal of Production Economics*, 71(1-3):125–133.

Hyndman, R. J. (2008). *Forecasting with exponential smoothing.* Springer, Berlin.

Hyndman, R. J. (2014). *Business forecasting: Practical problems and solutions.* John Wiley & Sons.

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

Johnston, F. and Boylan, J. (1996). Forecasting intermittent demand: A comparative evaluation of croston's method. comment. *International Journal of Forecasting*, 12:297–298.

Keaton, M. (1995). Using the gamma distribution to model demand when lead time is random. *Journal of Business Logistics*, 16:107–131.

Kuhn, M. and Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, Boca Raton, FL, USA.

Lawrenson, J. (1986). Effective spares management. *International Journal of Physical Distribution & Materials Management*, 16(5):3–111.

Levi, R., Perakis, G., and Uichanco, J. (2015). The data-driven newsvendor problem: New bounds and insights. *https://doi.org/10.1287/opre.2015.1422*, 63:1294–1306.

Levi, R., Roundy, R. O., and Shmoys, D. B. (2007). Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32:821–839.

Levén, E. and Segerstedt, A. (2004). Inventory control with a modified croston procedure and erlang distribution. *International Journal of Production Economics*, 90:361–367.

Li, S., Ragu-Nathan, B., Ragu-Nathan, T. S., and Subba R., S. (2006). The impact of supply chain management practices on competitive advantage and organizational performance. *Omega*, 34(2):107–124.

Lin, Y. and Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101:578–590.

Lingitz, K., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., and Monostori, L. (2018). Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. In *Procedia CIRP*, volume 72, pages 1051–1056.

Liu, J., Hwang, S., Yund, W., Boyle, L. N., and Banerjee, A. G. (2018). Predicting purchase orders delivery times using regression models with dimension reduction. In *38th Computers and Information in Engineering Conference*, volume 1B, page V01BT02A034, Quebec City, QC, Canada. American Society of Mechanical Engineers.

Lolli, F., Babai, M. Z., Coruzzolo, A. M., and Zanetti, L. (2025). Hierarchical clustering for joint replenishment: an application to spare parts. *International Journal of Production Research*.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018a). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34:802–808.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):e0194889.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2022). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, 38:1325–1336.

Min, H. (2010). Artificial intelligence in supply chain management: theory and applications. *International Journal of Logistics Research and Applications*, 13(1):13–39.

Moharana, U. C. and Sarmah, S. P. (2018). Joint replenishment of associated spare parts using clustering approach. *International Journal of Advanced Manufacturing Technology*, 94:2535–2549.

Muthuraman, K., Seshadri, S., and Wu, Q. (2015). Inventory management with stochastic lead times. *Mathematics of Operations Research*, 40(2):302–327.

Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, 291:549–559.

Oeztürk, A., Kayalıgil, S., and Özdemirel, N. (2006). Manufacturing lead time estimation using data mining. *European Journal of Operational Research*, 173(2):683–700.

Panjer, H. H. (1981). Recursive evaluation of a family of compound distributions*. *ASTIN Bulletin: The Journal of the IAA*, 12:22–26.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Pospisil, T. and Lee, A. B. (2018). Rfcde: Random forests for conditional density estimation. *arXiv preprint.*

Prak, D., Teunter, R. H., Babai, M. Z., Syntetos, A. A., and Boylan, J. E. (2018). Forecasting and inventory control with compound poisson demand using periodic demand data. *Research Reports*, 2018.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2017). Catboost: unbiased boosting with categorical features. *arXiv preprint.*

Quenouille, M. H. (1949). A relation between the logarithmic, poisson, and negative binomial series. *Biometrics*, 5(2):162–164.

Reiners, R., Haubitz, C. B., and Thonemann, U. W. (2025). Lead time prediction for inventory optimization with machine learning. *Production and Operations Management.*

Ritchie, E. and Kingsman, B. G. (1985). Setting stock levels for wholesaling: Performance measures and conflict of objectives between supplier and stockist. *European Journal of Operational Research*, 20(1):17–24.

Robinson, L. W., Bradley, J. R., and Thomas, L. J. (2001). Consequences of order crossover under order-up-to inventory policies. *Manufacturing and Service Operations Management*, 3:175–188.

Rodríguez, P., Bautista, M. A., Gonzalez, J., and Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31.

Saldanha, J. P., Price, B. S., and Thomas, D. J. (2023). A nonparametric approach for setting safety stock levels. *Production and Operations Management*, 32:1150–1168.

Sani, B. (1995). *Periodic inventory control systems and demand forecasting methods for low demand items.* PhD thesis, University of Lancaster.

Santner, T. J., Williams, B. J., Notz, W. I., and Williams, B. J. (2003). *The design and analysis of computer experiments*, volume 1. Springer, New York, NY.

Scornet, E. (2016). Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62:1485–1500.

Scornet, E., Biau, G., and Vert, J. P. (2015). Consistency of random forests. *https://doi.org/10.1214/15-AOS1321*, 43:1716–1741.

Semenoglou, A. A., Spiliotis, E., Makridakis, S., and Assimakopoulos, V. (2021). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 37:1072–1084.

Sherbrooke, C. C. (2004). *Optimal Inventory Modeling of Systems: Multi-Echelon Techniques*. Springer, 2nd edition.

Shore, H. (1986). General approximate solutions for some common inventory models. *Journal of the Operational Research Society*, 37:619–629.

Sieke, M. A., Seifert, R. W., and Thonemann, U. W. (2012). Designing service level contracts for supply chain coordination. *Production and Operations Management*, 21(4):698–714.

Silver, W. A., Pyke, D. F., and Peterson, R. (1998). *Inventory management and production planning and scheduling*. John Wiley & Sons, New York, NY, US, 3 edition.

Silver, W. A., Pyke, D. F., and Thomas, D. (2016). *Inventory and Production Management in Supply Chains*. CRC Press, Boca Raton, FL, USA, 4 edition.

Song, J. S., Zhang, H., Hou, Y., and Wang, M. (2010). The effect of lead time and demand uncertainties in (r, q) inventory systems. *Operations Research*, 58(1):68–80.

Syntetos, A. A., Babai, M. Z., and Altay, N. (2011). *Distributional Assumptions for Parametric Forecasting of Intermittent Demand*, pages 31–52. Springer London.

Syntetos, A. A., Babai, M. Z., and Altay, N. (2012). On the demand distributions of spare parts. *International Journal of Production Research*, 50(8):2101–2117.

Syntetos, A. A., Babai, M. Z., and Gardner, E. S. (2015). Forecasting intermittent inventory demands: simple parametric methods vs. bootstrapping. *Journal of Business Research*, 68:1746–1752.

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

Syntetos, A. A. and Boylan, J. E. (2001). On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71:457–466.

Syntetos, A. A. and Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2):303–314.

Syntetos, A. A. and Boylan, J. E. (2008). Demand forecasting adjustments for service-level achievement. *IMA Journal of Management Mathematics*, 19:175–192.

Syntetos, A. A., Boylan, J. E., and Disney, S. M. (2009). Forecasting for inventory planning: A 50-year review. *The Journal of the Operational Research Society*, 60:149–160.

Tadikamalla, P. R. (1984). A comparison of several approximations to the lead time demand distribution. *Omega*, 12:575–581.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4):437–450.

Teunter, R. H. and Duncan, L. (2009). Forecasting intermittent demand: a comparative study. *Journal of the Operational Research Society*, 60(3):321–329.

Teunter, R. H., Syntetos, A. A., and Babai, M. Z. (2010). Determining order-up-to levels under periodic review for compound binomial (intermittent) demand. *European Journal of Operational Research*, 203:619–624.

Teunter, R. H., Syntetos, A. A., and Zied Babai, M. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, 214(3):606–615.

Torgo, L., Ribeiro, R., Pfahringer, B., and Branco, P. (2013). Smote for regression. In *Portuguese conference on artificial intelligence*, volume 8154, pages 378–389.

Tsai, C. Y., Tsai, C. Y., and Huang, P. W. (2009). An association clustering algorithm for can-order policies in the joint replenishment problem. *International Journal of Production Economics*, 117:30–41.

Tukey, J. W. (1992). *Breakthroughs in Statistics: Methodology and Distribution*. Springer, New York, NY.

Turrini, L. and Meissner, J. (2019). Spare parts inventory management: New evidence from distribution fitting. *European Journal of Operational Research*, 273(1):118–130.

Tyworth, J. E. and O'neill, L. (1997). Robustness of the normal approximation of lead-time demand in a distribution setting. *Naval Research Logistics*, 44:165–186.

Urban, T. (2009). Establishing delivery guarantee policies. *European Journal of Operational Research*, 196(3):959–967.

Van der Auweraer, S., Boute, R. N., and Syntetos, A. A. (2019). Forecasting spare part demand with installed base information: A review. *International Journal of Forecasting*, 35:181–196.

Van der Auweraer, S., Zhu, S., and Boute, R. N. (2021). The value of installed base information for spare part inventory control. *International Journal of Production Economics*, 239:108186.

Vandeput, N. (2021). *Data Science for Supply Chain Forecasting*. De Gruyter, Berlin, Boston.

Vernimmen, B., Dullaert, W., Willeme, P., and Witlox, F. (2008). Using the inventory-theoretic framework to determine cost-minimizing supply strategies in a stochastic setting. *International Journal of Production Economics*, 115:248–259.

Viswanathan, S. and Zhou, C. (2008). A new bootstrapping based method for forecasting and safety stock determination for intermittent demand items. Working paper, Nanyang Business School, Nanyang Technological University Singapore.

Walzner, D., Fügener, A., and Denz, C. (2025). Combining humans and artificial intelligence in data-driven surgery scheduling. Working paper, University of Cologne, October 8, 2025.

Wang, M. C. and Rao, S. S. (1992). Estimating reorder points and other management science applications by bootstrap procedure. *European Journal of Operational Research*, 56:332–342.

Wang, W. (2012). A stochastic model for joint spare parts inventory and planned maintenance optimisation. *European Journal of Operational Research*, 216(1):127–139.

Webster, J. J. and Kit, C. (1992). Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on Computational linguistics*, volume 4, pages 1106–1110, Nantes, France. Association for Computational Linguistics.

Weisberg, S. (2005). *Applied linear regression*, volume 528. John Wiley & Sons.

Wiley, M. and Wiley, J. F. (2019). *Advanced R statistical programming and data models: Analysis, machine learning, and visualization.* Apress Media LLC, Berkeley, CA.

Willemain, T. R. and Smart, C. N. (2001). System and method for forecasting intermittent demand. US Patent 6,205,431.

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

Willemain, T. R., Smart, C. N., Shockor, J. H., and DeSautels, P. A. (1994). Forecasting intermittent demand in manufacturing: a comparative evaluation of croston's method. *International Journal of Forecasting*, 10:529–538.

Yang, B. and Geunes, J. (2007). Inventory and lead time planning with lead-time-sensitive demand. *IIE Transactions*, 39(5):439–452.

Zhang, Y., Jin, R., and Zhou, Z. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1):43–52.

Zhou, C. and Viswanathan, S. (2011). Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. *International Journal of Production Economics*, 133:481–485.

Zipkin, P. H. (2000). *Foundations of inventory management.* McGraw-Hill.

Curriculum Vitae: Robin Reiners

Personal Information

Address:	Graeffstr. 43, 50823 Cologne, Germany
Date of Birth:	January 1st, 1994
City of Birth:	Willich, Germany
Nationality:	German

Educational Background

Since 10/2021	Doctoral Studies (Dr. rer. pol.) at the Department of Supply Chain Management and Management Science, University of Cologne, Germany
10/2018 – 09/2021	Master Studies (M.Sc.) in Business Administration - Supply Chain Management, University of Cologne, Germany
09/2017 – 01/2018	DCU Business School, Dublin City University, Ireland
10/2015 – 09/2018	Bachelor Studies (B.Sc.) in Business Administration, University of Cologne, Germany
08/2004 – 07/2012	University Entrance Diploma (Abitur) at Bischöfliche Maria Montessori Gesamtschule Krefeld, Germany

Work Experience

Since 10/2021	Research Assistant at the Department of Supply Chain Management and Management Science, University of Cologne, Germany
Since 04/2021	Freelancing Supply Chain Consultant & Software Engineer
01/2019 – 01/2020	Working Student, Deloitte Consulting GmbH, Düsseldorf, Germany
02/2020 – 04/2021	Student Assistant, Area of Supply Chain Management, University of Cologne, Germany
10/2016 – 04/2021	Tutor for Micro Economics, University of Cologne, Germany

Cologne, September 2025

Eidesstattliche Erklärung nach § 8 Abs. 3 der Promotionsordnung vom 17.02.2015

Hiermit versichere ich an Eides Statt, dass ich die vorgelegte Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Aussagen, Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise unentgeltlich geholfen:

Laura Poreschak – Diskussionen, Korrekturlesen, Editing
Chantale Köster – Diskussionen, Korrekturlesen, Editing
Dominik Walzner – Diskussionen, Korrekturlesen, Editing
Jana Haßdenteufel – Korrekturlesen

Weitere Personen, neben den ggf. in der Einleitung der Arbeit aufgeführten Koautorinnen und Koautoren, waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Ich versichere, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Die Strafbarkeit einer falschen eidesstattlichen Versicherung ist mir bekannt, namentlich die Strafandrohung gemäß § 156 StGB bis zu drei Jahren Freiheitsstrafe oder Geldstrafe bei vorsätzlicher Begehung der Tat bzw. gemäß § 161 Abs. 1 StGB bis zu einem Jahr Freiheitsstrafe oder Geldstrafe bei fahrlässiger Begehung.

Köln, 07.10.2025

Robin Reiners