To the Graduate Council:

I am submitting herewith a dissertation written by M Harshvardhan entitled "From Data to Decisions: Machine Learning for Enterprise Demand Forecasting." I have examined the final paper copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Analytics.

_____

Dr. Chuanren Liu, Major Professor

We have read this dissertation
and recommend its acceptance:

_____

Dr. Paolo Letizia

_____

Dr. Tingliang Huang

_____

Dr. Xiaopeng Zhao

Accepted for the Council:

_____

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

To the Graduate Council:
I am submitting herewith a dissertation written by M Harshvardhan entitled "From Data to Decisions: Machine Learning for Enterprise Demand Forecasting." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Analytics.

Dr. Chuanren Liu, Major Professor

We have read this dissertation
and recommend its acceptance:

Dr. Paolo Letizia

Dr. Tingliang Huang

Dr. Xiaopeng Zhao

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

From Data to Decisions: Machine Learning for Enterprise Demand Forecasting

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

M Harshvardhan

May 2025

Dedicated to my father Rajendra Prasad, my mother Chandra Lata Barnwal, and my motherland India.

Acknowledgements

First and foremost, I would like to thank Dr. Chuanren (Charles) Liu for his constant investment of time and effort in making me a better researcher. When I encountered roadblocks in research, he suggested innovative ways forward. His willingness to grant me significant freedom in research direction and time is deeply appreciated. His thoughtful comments on my writing have improved my research communication significantly, and I'm deeply indebted to him for that.

I extend my gratitude to my dissertation committee members: Dr. Paolo Letizia, Dr. Tingling Huang, and Dr. Xiaopeng Zhao for their words of encouragement and valuable suggestions on my research.

My 16-month internship with the Strategic Planning and Modeling (SPaM) team at HP Inc. has significantly shaped this dissertation. I am grateful to Cara Curtland, Jerry Hwang, Barrett Crane, Dr. Pedro Neto, Chuck VanDam, Frederic Marie, and Adam Ghozeil. Cara has been a driving force in helping my work gain acceptance within HP and achieve external recognition by supporting my research beyond the application at HP. I also look up to her for life and career advice. Jerry taught me best practices for coding and clever Python tricks that have stuck with me ever since. Additionally, I appreciate their support in co-authoring our research, which has been accepted for publishing in the *INFORMS Journal of Applied Analytics* and *Foresight*, helping to bring our forecasting work to a broader academic and industry audience.

Working with Dr. Sean Willems during my first year was a turning point—it laid the foundation for everything that followed in my PhD journey. Through the

process of finding a research topic of mutual interest, I gained a deep understanding of applied research. His work serves as an exemplar of bridging theoretical frameworks with applied technologies. While I may not achieve all aspects of the gold standard he established, it remains the North Star guiding my research. His many adages like "never compare your private self with someone's public self" have contributed greatly to my happiness levels.

Throughout my academic journey at UTK, I had the privilege of taking 15 courses, each contributing uniquely to my development, and I am grateful to the instructors for deepening my understanding across disciplines. Among them, Dr. Emre Demirkaya's Statistical Learning stood out for its rigorous, blackboard-driven approach to the mathematical foundations of machine learning. Equally formative was Dr. Luiz Lima's Time-Series Econometrics—together, these were the most challenging yet rewarding analytical experiences of my PhD.

Dr. Wenjun Zhou has been a great mentor, consistently making time to provide expert guidance whenever needed. I am also grateful to Dr. Michael Galbreth and Dr. Robert Mee for their excellent career advice, and to Terry Higgins for helping me develop as an educator.

I am indebted to Dr. Pritam Ranjan who inspired me to take up research as a career option. During my undergraduate at IIM Indore, my summer research internships with him fueled my passion for coding and statistics. Eventually after my M.B.A., he encouraged me to undertake a Ph.D. — if it weren't for his prodding, I wouldn't have chosen this career path. His guidance continues to shape my academic journey and professional decisions.

Dr. Christine Vossler, Laura Watts, and Janice Wade have made my time at UTK easier by shielding me from most administrative tasks. Mary Kay Mee, Bob Mackey, Lindon Thomas, and the International Navigators group made my stay in Knoxville much more fun with hikes and thoughtful conversations.

The friendships formed during graduate school have been invaluable. Nikhil Narayane, Yu Jiang, and I have strengthened our bond through shared hardships

over the past four years; I'm grateful to have such wonderful friends as my PhD cohort. My life in Knoxville has been brightened by wonderful friends. Greeshma Geetha's company in Fort Sanders Manor was a springboard whenever I felt low; the persisting idealist in her encouraged me to question status quo, in research and in life. Pablo Reboredo-Segovia taught me that learning is more about exploration than education. Sourav Pan's excellent vegetarian cooking always left our mouths watering. Samudra Dasgupta and his dog Henley were delightful friends; I also thank him for introducing me to Jiddu Krishnamurthy. My weekly serendipity talks with Dea Bardhoshi brought moments of calm reflection.

To Meenal Singh, thank you for being my favorite person in the world; life has been a wonderful journey since meeting you. I am also deeply grateful to my parents, Rajendra Prasad and Chandra Lata Barnwal, my brother Shashank, and my sister Shalini Kaushal, for their unwavering support of my work and life decisions.

Also, can I thank the internet? It has been a fundamental utility for me to track new research, exchange messages, and maintain connections; even keeping up with memes for my sanity would have been impossible without it. I am profoundly grateful for modern technology and artificial intelligence tools, including ChatGPT and Claude, which have enhanced my research capabilities and fostered intellectual curiosity. These innovations have significantly streamlined my quest for knowledge and enriched both my current and future research endeavors.

यथा पिण्डे, तथा ब्रह्माण्डे।

As is the microcosm, so is the macrocosm.

Abstract

Accurate demand forecasting is critical for operational efficiency and strategic decision-making in large-scale enterprises. This dissertation presents a machine learning (ML)-driven demand forecasting framework implemented at a Fortune-500 company HP Inc., focusing on three key areas: ML-based predictive modeling, MLOps and deployment scalability, and Human-in-the-loop forecasting integration. Additionally, we explore how predictive optimization enhances decision-making through end-to-end learning.

The first contribution involves the development of a scalable ML-based forecasting system, leveraging tree-based models (LightGBM), feature engineering, and advanced time-series methodologies. The model captures complex demand drivers, including macroeconomic trends, product life cycle effects, and channel inventory dynamics. By transitioning from traditional statistical models to ML-based approaches, the framework improves forecasting accuracy in key metrics while adapting to evolving market conditions.

The second contribution addresses MLOps and enterprise-scale deployment challenges, ensuring model reliability, automation, and reproducibility. The research outlines best practices in model monitoring, version control, model deployment, and continuous learning pipelines, demonstrating how systematic ML deployment reduces technical debt and maintains forecast accuracy over time.

The third contribution integrates 'Human-in-the-Loop' forecasting, ensuring that ML predictions are refined through expert-driven consensus mechanisms. The system

incorporates business intelligence inputs such as sales insights, promotional strategies, and market conditions, balancing data-driven automation with human expertise to enhance interpretability and trust in forecasts. Through this closed-loop process, we are able to improve the overall forecast accuracy by 34% (wMAPE) and reduce inventory by 28% while maintaining same service levels.

Finally, this dissertation presents a predictive optimization framework that transforms ML-based predictions into actionable strategies. We showcase how perfect predictions still don't lead to perfect decisions through a simulation study. Subsequently, we propose an end-to-end learning paradigm that simultaneously addresses demand forecasting, inventory allocation, procurement planning, and production scheduling in the supply chain.

Table of Contents

1 Introduction **1**

1.1 Evolution of Demand Forecasting 2

1.2 Background 4

1.2.1 Demand Forecasting Process 6

1.2.2 Strategic Planning and Modeling Group (SPaM) 7

1.3 Research Challenges 9

1.3.1 Complex Multi-factorial Demand Patterns 9

1.3.2 Dynamic Adaptability Requirements 11

1.3.3 Data Quality and Scale 11

1.3.4 Handling ML Models and Empowering Planners 12

1.4 Research Contributions 13

1.4.1 Scalable ML-based Forecasting Framework 13

1.4.2 MLOps for Enterprise Forecasting 13

1.4.3 Human-in-the-Loop Architecture 14

1.4.4 Practical Contributions 14

1.4.5 Predictive Optimization 15

1.5 Related Publications 15

1.6 Organization of Dissertation 16

2 Literature Review **17**

2.1 Overview of Demand Forecasting 18

2.1.1 Analytical Forecasting Methods . . . 18
2.1.2 Judgmental Forecasting Methods . . . 20
2.1.3 Human-in-the-loop Ensembling: Integrating Analytical and Judgmental Forecasting . . . 22
2.2 Machine Learning Based Demand Forecasting . . . 24
2.2.1 Direct vs Iterative Forecasting . . . 26
2.2.2 External Features . . . 27
2.3 MLOps in Large-Scale Machine Learning Deployments . . . 27
2.3.1 Challenges in Enterprise ML Deployment . . . 28
2.3.2 MLOps Solutions and Best Practices . . . 30
2.3.3 Unified Platforms and Project Management . . . 31
2.4 The Way Forward . . . 33

3 Forecasting Demand with Machine Learning 35
3.1 Problem Definition and Formulation . . . 36
3.2 Iterative Forecasting Algorithm . . . 38
3.3 LightGBM: A Succinct Summary . . . 40
3.4 Model Input Features . . . 41
3.4.1 Feature Selection . . . 42
3.5 Performance Evaluation . . . 46
3.5.1 Evaluation Metrics . . . 46
3.5.2 Results . . . 48
3.6 Conclusion and Future Work . . . 51

4 MLOps: Machine Learning Operations 54
4.1 Ensuring Reproducible Experimentation . . . 56
4.2 MLFlow . . . 58
4.3 Notebook Orchestration and Reproducibility Framework . . . 60
4.4 Data Storage . . . 61
4.5 Experimentation with FLAML . . . 64

4.6 Conclusion 66

5 Operationalizing ML Forecasting with Human-in-the-loop Framework 68

5.1 Implementation Journey 69
5.2 Dashboard 72
5.3 Human-in-the-loop Ensembling 73
5.4 Operational Benefits 75
5.4.1 Inventory Reduction 75
5.4.2 Forecast Accuracy Improvements 77
5.5 Principles and Lessons: Making Analytical Forecasts Actionable . . . 79
5.6 Conclusion 80

6 Bridging Demand Forecasting and Decision Optimization 81

6.1 Good Demand Forecasts Good Production Planning Decision 82
6.1.1 Uncorrelated Random Errors: Theoretical Best Case 84
6.1.2 Correlated Forecast Errors: The Realistic Scenario 86
6.1.3 Implications for Predictive Optimization 87
6.2 Predictive Optimization for Supply Chain Management 89
6.2.1 Integrated Demand Forecasting and Production Optimization 89
6.2.2 Production Planning Formulation 90
6.2.3 Extension to Multiple Products 92
6.2.4 Differentiable Lagrangian Optimization 93
6.2.5 Optimization Regret and End-to-End Training 94
6.3 Conclusion and Future Directions 94

7 Concluding Remarks 96

Bibliography 99

Vita 112

List of Tables

2.1 Summary of related research papers on ML-based demand forecasting. 25

3.1 Summary of Forecasting Model Input Features and Their Utility . . . 43

3.2 Forecasting accuracy metrics (bias, RMSE, and wMAPE) for cumulative forecast horizons (CM1, CM3, CM6) with Mean (Standard Deviation). 49

3.3 Forecasting Accuracy Metrics: Bias, wMAPE, RMSE Comparison for CONS, ML, and STAT Methods. 50

4.1 Summary of MLOps Principles and Implementation Components . . 57

4.2 File Format Benchmark Results for 1M Rows and 10 columns Synthetic Dataset with Relative Performance 65

List of Figures

1.1 Overview of the forecasting process. Our approach leverages historical and additional data to create robust statistical and machine learning forecasts. These forecasts are then refined by consensus planners, serving as the crucial human element in the loop, to formulate a comprehensive forecast that informs granular supply planning. The focus of this work is 'ML Forecasting'. 8

3.1 Dumbbell plot visualizing the mean (center point) and one standard deviation (vertical lines) of Bias, RMSE, and WMAPE for three forecasting methods (Consensus, Machine Learning, and Statistical) over cumulative forecast horizons of one month (CM1), three months (CM3), and six months (CM6). 49

3.2 Bias, WMAPE, and RMSE metrics over 12 months show that the ML model is consistently among the top performers of the three models. CM1 is point forecast, while CM3 and CM6 are three and six months cumulative forecasts, respectively. 50

4.1 Figure of integrated MLOps framework showing the intersection of Machine Learning, Development, and Operations with a continuous workflow pipeline and key roles and benefits highlighted. 55

4.2 Project management for continuous deployment pipeline of our ML forecasting efforts. 59

4.3 Overview of `papermill` library and its key benefits. 62

5.1 Implementation journey of our ML forecasting project at HP Inc. . . 70
5.2 Illustration of Human-in-the-Loop Forecasting that combines Machine Precision with Human Insight. 74
5.3 Inventory trajectory from 2022 to 2025 showing reduction by 28.5%. . 76
5.4 Performance metrics illustrating forecasting accuracy improvements: (a) wMAPE and (b) Bias from 2022 to 2025. 78

6.1 Decision errors measured as objective cost difference against forecast errors measured with RMSE for purely uncorrelated forecasts with random errors. 85
6.2 Decision errors measured as objective cost difference against forecast errors measured with RMSE when forecasts and true values have correlation $\rho = 0.7$. 88

Chapter 1

Introduction

For our organs of sense, after all, are a kind of instrument. We can see how useless they would be if they become too sensitive. — Erwin Schrödinger

Accurate demand forecasting is fundamental to a company's operational and financial success in today's complex business environment. When organizations can effectively predict future demand for their products and services, they can optimize their entire supply chain — from procurement and production planning to inventory management and distribution. Poor forecasting leads to significant challenges: overforecasting results in excess inventory and increased holding costs, while underforecasting causes stockouts, lost sales, and damaged customer relationships.

The stakes are particularly high for large multinational companies managing diverse product portfolios across global markets, where forecast errors can cascade through the supply chain and significantly impact profitability. Furthermore, with increasing market volatility, shorter product lifecycles, and complex customer preferences, traditional forecasting approaches are often insufficient. Modern companies require sophisticated forecasting systems that can capture nuanced demand patterns, adapt to market changes, and scale across thousands of products and multiple geographies. This challenge of developing and implementing effective demand

forecasting capabilities represents a critical strategic imperative for businesses seeking to maintain competitive advantage in dynamic global markets.

1.1 Evolution of Demand Forecasting

Demand forecasting has rich historical roots across civilizations. Ancient Chinese officials in the Han Dynasty (110 BCE) combined multiple data sources — harvest records, weather patterns, and population data — to forecast grain demand through their 'Ever-Normal Granary' system, a practice that mirrors modern ensemble forecasting approaches. Similarly, ancient Indian texts like Arthashastra (350 BCE) advocated for blending quantitative data with qualitative factors in resource prediction, a principle that remains central to contemporary forecasting systems.

The formalization of demand forecasting methods probably began in the early 20th century United States, with Ford Motor Company and General Motors implementing the first structured production planning systems around 1920s (O'Brien, 1989). Statistical rigor was introduced through exponential smoothing (Brown, 1956) and the Box-Jenkins ARIMA methodology (1970), establishing the mathematical foundations of modern forecasting (Box et al., 2015). Japanese manufacturers, particularly Toyota in the 1950s, contributed significantly by developing Just-in-Time systems that introduced pull-based forecasting concepts.

However, the landscape of demand forecasting has transformed dramatically over the past few decades due to several converging factors. First, the rise of multinational corporations has exponentially increased the complexity of forecasting needs. For instance, while Ford in 1921 managed forecasting for essentially one product (the Model T) in one primary market, modern global companies like HP Inc.* must forecast demand for tens of thousands of SKUs across over 170 countries. Second, the advent of enterprise resource planning (ERP) systems in the 1990s, coupled with

*HP Inc. is an American multinational company manufacturing personal computers, printers and related supplies. See https://www.hp.com/, accessed March 11, 2025.

point-of-sale (POS) data collection, created vast repositories of historical sales and inventory data. The scale of available data expanded further with the emergence of e-commerce in the 2000s, which provided granular visibility into customer behavior and preferences.

The technological enablers for modern ML-based forecasting systems emerged in parallel. The development of powerful tree-based algorithms like Random Forests (Breiman, 2001) and gradient boosting methods like XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017) provided the computational tools needed to handle high-dimensional forecasting problems. Cloud computing platforms, becoming mainstream in the 2010s, solved the infrastructure challenges of processing massive datasets. Additionally, the maturity of MLOps tools and frameworks has made it feasible to deploy and maintain large-scale forecasting systems in production environments (Kreuzberger et al., 2023).

What's particularly significant about this evolution is not just the technological progression, but how it has fundamentally changed the nature of forecasting itself. Traditional forecasting methods treated each product-location combination as an independent time series to forecast. Modern ML-based systems can instead learn patterns across multiple products and markets simultaneously, incorporating diverse data sources like macroeconomic indicators, weather patterns, competitive activities, and social media signals. This shift from 'local' to 'global' forecasting models, combined with the ability to process real-time or rhythmic data streams, has enabled a level of forecast accuracy and adaptability that was previously unattainable.

ML algorithms for demand forecasting, despite their high general accuracy, can make significant and costly errors in real-world applications. For instance, Worten (2003) documented how Nike incurred $400 million in inventory write-offs due to overreliance on purely algorithmic forecasting. These analytical models are currently unable to incorporate qualitative information gained through planner expertise and direct communication between account managers and end customers. Additionally, planners often require control over the final output, as accuracy may not always be the

sole objective; various other managerial targets may take precedence. Consequently, human oversight remains both necessary and warranted.

The current frontier in demand forecasting lies in developing hybrid systems that combine the computational power of ML algorithms with human expertise and domain knowledge. These systems must not only process historical data but also adapt to rapid market changes, account for new product introductions, and handle supply chain disruptions — challenges that are particularly relevant in today's volatile global business environment.

In this dissertation, I present a comprehensive case study of implementing ML-based demand forecasting at HP Inc., one of the world's largest technology companies worldwide. Drawing upon the rich historical foundations of forecasting and leveraging modern computational advances, we demonstrate how traditional forecasting approaches can be enhanced through ML while preserving valuable human expertise. This work represents a significant real-world application of theoretical forecasting principles, addressing challenges that arise when scaling forecasting systems to handle tens of thousands of products across global markets. Our implementation builds upon decades of forecasting research while incorporating contemporary advances in machine learning and data processing capabilities. Before delving into our specific contributions, we present the background of HP's forecasting systems, which provides crucial context for understanding the scale and complexity of the problem we address.

1.2 Background

HP Inc. (HP) is an American multinational information technology company headquartered in Palo Alto, California. It develops, manufactures, and distributes personal computers (PCs), printers, and related supplies, along with 3D printing services. Founded as Hewlett-Packard in 1939, the company split in 2015 into HP Inc. (consumer-focused PCs and printers) and Hewlett Packard Enterprise (enterprise

solutions). HP Inc. is the legal successor of the original Hewlett-Packard. In the fiscal year 2024, HP reported a global revenue of approximately $54 billion, with one-third coming from its Print business.

HP's Print division manufactures and sells over 18,000 Stock Keeping Units (SKUs) of print products that are sold in over 170 countries. They include home printers, office printers, ink, toner, and other services such as 3D and large-format printing. The products are classified into five categories: Home Print Services (HPS), Office Print Services (OPS), Ink, Toner, and Others (e.g, 3D and large-format printing). Specifically, home printers are targeted to consumers looking to buy standalone printers. They're usually sold through channel partners including retailers like Walmart and Amazon. Office printers are usually sold via business contracts through managed account deals. The consumables, Ink and Toner, are sold to existing printer-owners. Inks' customer base is Inkjet users, while Toner's customer base is Laserjet users. 3D Printing offers a portfolio of additive manufacturing solutions and supplies to help customers with unique or experimental demands. Additionally, HP also offers large-format printing solutions and supplies through industrial products, including HP DesignJet, HP Latex, HP Indigo, and HP PageWide Web Presses. Beyond these five top-level categories, products are further classified based on their technology and platform, resulting in over 18,000 SKUs. Building on this portfolio breadth, HP operates on a global scale with markets organized into three world regions: Americas (AMS), Europe, Middle-East and Africa (EMEA), and Asia-Pacific (APAC). Countries in each world region are grouped by geographical proximity and the demand forecasting is needed for each SKU in each Group of Countries (GOC).

Given the diverse product portfolio and extensive global reach, accurate demand forecasting is a crucial component of operational strategy for an international company like HP. Accurate forecasts are critical to planning and operational decisions such as strategically allocating resources, managing inventory, and aligning production schedules with consumer demand (Gardner, 1990; Ritzman and King, 1993; Lee, 2002; Seifert et al., 2015). Furthermore, past studies have highlighted

that effective forecasting can not only support business operations, but can also lead to cost savings and improved efficiency throughout the supply chain (Simatupang and Sridharan, 2005; Seifert et al., 2015; Fildes et al., 2022). With the advancement of machine learning technologies, there's been a significant interest from academics and practitioners in applying ML methods for these forecasting tasks. This paper discusses the challenges and solutions to deploy an ML-based framework to forecast product demand for a Fortune-500 technology company like HP.

1.2.1 Demand Forecasting Process

Before implementing ML-based models, HP relied on *Statistical* and *Consensus* forecasts for demand forecasting. The *Statistical* forecasts leverage historical demand data and use conventional time-series models, such as autoregressive (AR), moving averages (MA), ARMA, ARIMA, and exponential smoothing (ETS) models (Hyndman and Athanasopoulos, 2018). While these models are cost-effective and easy to implement, they often lack the nuance required for accurate forecasting due to oversimplified modeling assumptions. Statistical models are also 'local' in nature, training with a single time-series whereas ML-based models are 'global', incorporating details from multiple time-series. Additionally, statistical models cannot easily incorporate exogenous variables that might influence demand. Local models struggle with short product life cycles whereas a global model gets to learn from similar products. A common attempt to handle this is through predecessor-successor mapping, but such information isn't always readily available to forecasters (Manary et al., 2019).

In contrast, the Consensus forecasts incorporate quantitative information such as historical demand and current inventory levels, as well as qualitative demand signals and contextual information, with the Statistical (Analytical) forecast also serving as an input. Particularly, the Consensus forecasters heavily leverage 'soft data' like customer demand sentiments and deal progress. Soft data includes qualitative

knowledge on upcoming promotions offered by channel partners to their customers, deal stage for bulk corporate orders, subjective opinions from market insiders and experts, and networking insights through deep business relationships, among others (Fildes et al., 2009; Petropoulos et al., 2018). Though soft data is challenging to include and maintain, its strategic advantages in capturing transient market conditions make it invaluable to forecasting, especially contributing to robustness of planner forecasts. Figure 1.1 depicts and compares the different demand forecasting solutions, where our focus is to develop the new ML forecasts as shown in the orange box.

Moreover, the superiority of data-based method compared to human judgemental forecasts isn't always obvious. Zellner et al. (2021) surveyed literature on human judgement and quantitative forecasting as well as hybrid methods that involve both humans and algorithmic approaches. They found that while quantitative methods have gotten popular over time, they aren't universally superior to human judgement; the better method is subject to availability, quality, extent and format of data. Indeed, the two approaches can complement each other to yield more accurate and resilient models. Recent research also shows that human-based forecasts struggle to effectively filter out noise in the inputs. In fact, forecasters tend to reproduce the noise in a time-series in their forecasts rather than filter it out (Petropoulos and Siemsen, 2023). Khosrowabadi et al. (2022) evaluate AI-generated forecasts for a major European retailer, revealing that product attributes like price, freshness, and discounts play a crucial role in adjustment decisions. They find that while large positive adjustments are more common, they tend to be less accurate. In contrast, large negative adjustments, though less frequent, are generally more precise.

1.2.2 Strategic Planning and Modeling Group (SPaM)

Formed in 1994, SPaM is a team of OR specialists, data scientists, and external collaborators who provide internal support to HP product divisions to improve

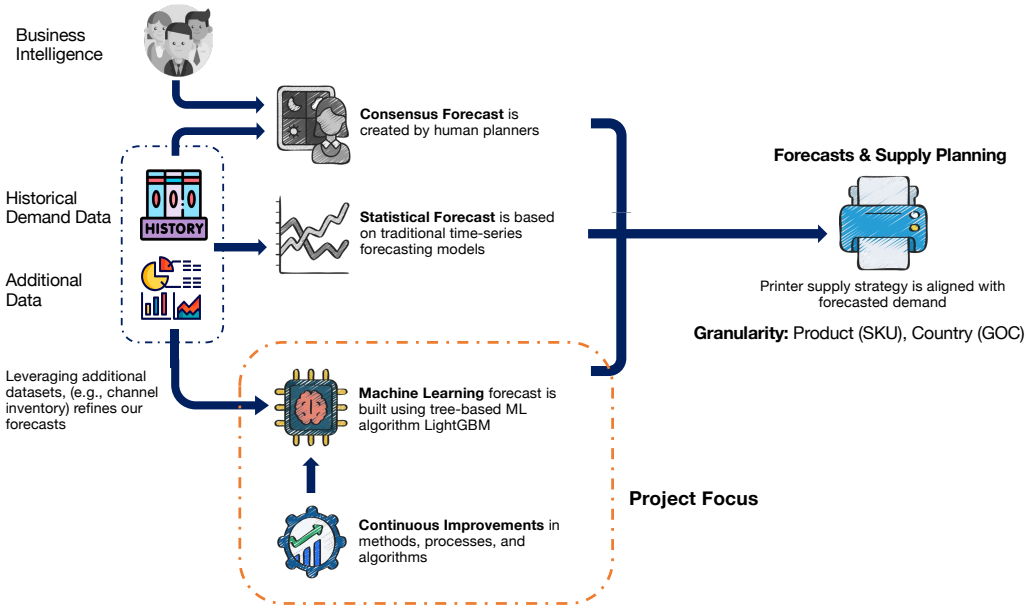


Figure 1.1: Overview of the forecasting process. Our approach leverages historical and additional data to create robust statistical and machine learning forecasts. These forecasts are then refined by consensus planners, serving as the crucial human element in the loop, to formulate a comprehensive forecast that informs granular supply planning. The focus of this work is 'ML Forecasting'.

their efficiency, cost-effectiveness, and profitability (Laval et al., 2005). SPaM has developed and adapted many supply chain models for specific applications at HP (Cargille and Branvold, 2000). For example, Ward et al. (2010) documents the team's work in transforming product portfolio management: developing a new framework for screening new products using custom return-on-investment calculators, and a revenue-coverage-optimization tool to manage product variety after introduction. Similarly, Billington et al. (2004) documents how efforts from SPaM helped HP create a standard process for analyzing and designing supply-chain networks. In 2019, SPaM was charged with the work on building the demand forecasting system and model at HP. More details on the staged implementation are provided in Section 5.1.

1.3 Research Challenges

The implementation of machine learning for demand forecasting at an enterprise scale presents several fundamental challenges that this dissertation addresses. These challenges span multiple dimensions - technical, operational, and organizational - each requiring careful consideration and novel solutions. Now, we examine these challenges in detail and outlines the contributions this research makes toward addressing them.

1.3.1 Complex Multi-factorial Demand Patterns

The first major technical challenge lies in capturing and modeling the intricate web of factors that influence product demand across different dimensions. Unlike traditional time-series problems where patterns might be dominated by seasonality or trends, enterprise-scale demand forecasting must account for a complex interplay of various inputs. Economic conditions vary significantly across markets, with regional economic indicators, currency fluctuations, and market-specific economic cycles all playing crucial roles in determining demand patterns. These economic factors interact with

regional variations in consumer behavior and preferences, creating unique demand signatures for each market.

The complexity extends further when considering seasonality patterns, which differ not only by geography but also by product category. For instance, consumer printer demand shows strong seasonal patterns aligned with academic calendars and festive seasons, but these patterns vary globally across countries. Additionally, commercial printing solutions may follow entirely different seasonal patterns tied to business cycles and fiscal years, which also vary by country.

Competitive dynamics add another layer of complexity to demand patterns. Market share fluctuations, competitor product launches, and promotional activities create short-term demand variations that must be captured by the forecasting model. Additionally, new products get launched in the market which often cannibalize existing products of the company. These competitive effects are often localized, requiring the model to understand and account for market-specific competitive landscapes while maintaining a coherent global perspective.

Supply chain constraints and their cascading effects further complicate the demand patterns. Manufacturing capacity limitations, transportation bottlenecks, and inventory holding constraints can create artificial demand patterns that must be distinguished from genuine market demand. The model must understand these supply-side constraints to avoid confounding supply limitations with reduced demand signals. This can be sensed through the amount of inventory that retailers already have stockpiled of a specific model.

All of these challenges are further magnified by the need to develop a single versatile model that can effectively handle these variations while maintaining computational efficiency across HP's extensive portfolio of 18,000+ SKUs and 170+ countries. This requires sophisticated feature engineering and model architecture decisions that can capture complex interactions between multiple factors while scaling efficiently across diverse product categories.

1.3.2 Dynamic Adaptability Requirements

The second critical challenge involves creating systems that can adapt to rapid market changes while maintaining forecast stability. Real-time integration of new data streams presents significant technical challenges, requiring sophisticated approaches to continuous data incorporation and dynamic feature importance adjustment. The system must detect and adapt to concept drift in demand patterns while maintaining sufficient stability to support operational planning.

Supply chain disruptions, which have become increasingly common in the global economy, require special handling within the forecasting system. The model must be able to distinguish between temporary anomalies and structural changes in demand patterns, adapting its predictions accordingly without overfitting to transient events. This becomes particularly challenging when dealing with major disruptions like the global pandemic, where historical patterns may provide limited guidance for future predictions. Product transitions present another significant adaptability challenge. As new products are introduced and older ones phased out, the system must smoothly handle these transitions while maintaining forecast accuracy. This requires sophisticated approaches to transfer learning, where patterns learned from existing products can inform predictions for new introductions, while accounting for changing market conditions and evolving consumer preferences.

1.3.3 Data Quality and Scale

The third technical challenge revolves around data management at scale. The sheer volume of data involved in enterprise-scale forecasting creates significant computational and storage challenges. Missing data, inconsistent reporting formats across regions, and varying data quality levels must be handled robustly by the system. The challenge extends beyond mere data cleaning to understanding the business context of data anomalies and developing appropriate correction strategies.

Data integration across multiple source systems presents additional complications. Different systems may use varying definitions, granularities, and update frequencies, requiring sophisticated harmonization approaches. Historical data preservation becomes crucial for model training and validation, while real-time processing capabilities are needed for operational deployment.

1.3.4 Handling ML Models and Empowering Planners

Beyond modeling considerations, implementing ML forecasting systems at scale presents significant technical and operational challenges. Integration with existing business processes requires careful coordination across multiple stakeholder groups. The system must align with established workflows while introducing new capabilities and insights. Change management becomes crucial as users become accustomed to using ML-based forecasts in addition to traditional forecasting approaches. As detailed in Subsection 1.3.1 and Subsection 1.3.2, we also need our forecasting system to be adaptable to various upcoming developments in terms of growing input features as well as upgrading of model itself. To that end, we created a bedrock framework for continual updates and resilient pipeline for creating and deploying ML models using MLOps. While machine-based forecasts can consume huge amounts of data, human expertise shines in handling unstructured information such as conversations with sales teams and other qualitative information. We do this by creating a Human-in-the-Loop framework that empowers planners to use ML forecasts when needed without forcing its adoption and thus enabling the optimal combination of machine precision and human insight. This framework, detailed in Chapter 4 and Chapter 5, allows forecasting models to interact with each other and integrate domain expertise of planners with technical capabilities of sophisticated models.

1.4 Research Contributions

This dissertation advances both the theoretical understanding and practical implementation of enterprise-scale forecasting systems through several significant contributions to the academic literature and industry practice.

1.4.1 Scalable ML-based Forecasting Framework

The first theoretical contribution of this work is the development and validation of a comprehensive framework for implementing tree-based models at enterprise scale. Built around LightGBM but flexible to future model change, this framework introduces novel approaches to model architecture design that effectively handle diverse product portfolios while maintaining computational efficiency. The framework incorporates advanced feature engineering strategies that capture complex demand patterns across different product categories and geographical regions. We develop efficient training and inference pipelines specifically designed for large-scale deployment, addressing the computational challenges of processing thousands of SKUs across multiple regions. The framework also introduces new approaches to handling product hierarchy and geographical variations, enabling effective demand forecasting across different organizational levels.

1.4.2 MLOps for Enterprise Forecasting

This research advances the emerging field of MLOps through several methodological contributions. We develop reproducible analysis workflows using parameterized notebooks, enabling consistent and repeatable experimentation across different product categories and geographical regions. The research establishes advanced experiment tracking methodologies that maintain transparency and reproducibility in the model development process. We introduce systematic approaches to model

performance monitoring that ensure sustained forecast accuracy over time. Additionally, we develop automated retraining and deployment pipelines that maintain model freshness while ensuring stability in production environments.

1.4.3 Human-in-the-Loop Architecture

A key methodological contribution is our novel human-in-the-loop architecture that creates synergy between ML capabilities and human domain expertise. The architecture implements a sophisticated ensembling framework allowing planners to combine predictions from different models based on their domain expertise, provides interactive dashboards for interpretable predictions, and establishes a closed feedback loop where planner decisions inform future model improvements. This approach solves a critical challenge in enterprise forecasting by balancing automation with human judgment, enabling the system to handle routine forecasts while empowering planners to focus on complex cases requiring human insight. Ultimately, our system augments the capabilities of the planners and doesn't replace them. With this implementation, we observe forecast accuracy improvement of 34% (wMAPE) and inventory reduction by 28% while maintaining similar customer service levels over three year period.

1.4.4 Practical Contributions

Through implementation at HP Inc., this research provides a comprehensive blueprint for enterprise-scale ML forecasting that demonstrates broad adaptability across different organizational contexts. The framework accommodates various industry settings, product portfolio structures, and geographical configurations, making it relevant for both smaller enterprises and large multinational corporations. Our implementation documents practical strategies for stakeholder management, provides performance comparisons between traditional and ML-based approaches, and establishes best practices for large-scale ML deployment.

1.4.5 Predictive Optimization

While advanced analytics systems can help create automated forecasts that aid human decision making, it would be valuable to supplement model building with the final decision-making process. We seek to explore how forecasting predictive modeling methods can be integrated with optimization-based decision making, a field called *Predictive Optimization*. To that end, we first demonstrate through a simulation study that when a decision-making paradigm can be formally defined as an optimization problem in production planning, making the best forecast doesn't necessarily lead to the best decisions. Building on that intuition and motivated by Mao et al. (2023), we present an end to end predictive optimization framework for supply chain where demand forecasting and final decision making processes are merged to perform end-to-end model learning.

1.5 Related Publications

In addition to the research presented in this dissertation, several aspects of this work have been accepted or published elsewhere. The *INFORMS Journal of Applied Analytics* paper focuses on the methodological aspects of the forecasting framework, providing a comprehensive account of the development, validation, and deployment of the machine learning models at HP Inc. (Harshvardhan et al., 2025b). In contrast, the *Foresight* paper—an invited work following our recognition as a Top-5 team in the IIF Forecasting Practice Competition at the *2025 Foresight Practitioners Conference*—is intended for a managerial audience and serves as an executive summary of the implementation journey, emphasizing strategic and organizational insights (Harshvardhan et al., 2025a). Additionally, concepts discussed in Chapter 6, particularly those related to predictive optimization are motivated from a similar work on end-to-end advertisement inventory management at Alibaba, presented at *ACM SIGKDD Conference* in Long Beach, California and later published in the KDD 2023

Proceedings (Mao et al., 2023). These prior works serve as foundational contributions that are further expanded and integrated into this dissertation, providing a holistic view of forecasting at scale in an enterprise environment.

1.6 Organization of Dissertation

These contributions lay a robust foundation for future research on enterprise-scale ML forecasting systems and offer practical insights for organizations undertaking similar digital transformations. The rest of the dissertation is organized as follows. Chapter 2 surveys existing literature on applied demand forecasting techniques, integration of human judgement in forecasting, apparatus involved in MLOps. Chapter 3 delves into the technical components of the scalable ML forecasting framework implemented at HP. Chapter 4 outlines the MLOps infrastructure and the project management strategies critical for deploying the forecasting pipeline at scale. Chapter 5 illustrates how planner buy-in was secured and details the integration of ML forecasts into their existing workflows through an empowering human-in-the-loop system. Chapter 6 discusses extending demand forecasting to inform end-to-end decision-making using predictive optimization, supported by a detailed simulation study and a proposed end-to-end learning paradigm for supply chain decision making. Finally, Chapter 7 provides concluding remarks and recommendations for future researchers and practitioners.

Chapter 2

Literature Review

There are these three kinds of wisdom — wisdom acquired through study of literature, wisdom from contemplation, and wisdom from direct experience.
— Visuddhimagga by Buddhaghosa, trans. Bhikkhu Ñāṇamoli

Forecasting can be defined as "the art of predicting the occurrence of events before they actually take place", according to Archer (1980). It enables policy makers to make decision s before the advent of predicted happenings which affect, or are affected by, their actions. Thus, demand forecasting is at the heart of planning and decision making.

In this section, I review multiple sets of works from the literature around overview of time-series forecasting including traditional methods of demand forecasting and machine learning methods which utilize advanced computational methods. Followed by that, I note that ML methods of forecasting often require specialized infrastructure in organizations and study the literature around Machine Learning Operations (MLOps). Then, I study some case studies which demonstrate various forecasting methods and their implementation in practice. Finally, I present the research gap fulfilled by this dissertation.

2.1 Overview of Demand Forecasting

Demand forecasting models serve as cornerstones in production and inventory management systems (Gardner, 1990). These models enhance efficiency by streamlining operations and improving customer satisfaction (Heikkilä, 2002). Accurate forecasts reduce inventory holding costs, guide timely procurement of raw materials, and improve responsiveness to demand fluctuations (Gardner, 1990). Ultimately, businesses benefit through optimized resource utilization and enhanced customer satisfaction (Heikkilä, 2002).

There is a large body of literature on demand pattern recognition and prediction. Broadly speaking, there are two primary approaches to demand forecasting: *analytical forecasting methods* (often statistical or quantitative) and *judgmental forecasting methods* (qualitative, relying on expert input). The analytical methods typically leverage historical data, extrapolating observed trends into the future, while judgmental forecasting captures expert insights or domain-specific knowledge not easily quantified.

In the following subsections, we describe both methods in detail and explain how their integration, through human-in-the-loop ensembling, leverages their respective strengths to produce superior forecasts.

2.1.1 Analytical Forecasting Methods

Analytical forecasting methods, also known as time-series methods, predict future values based on historical data without relying on subjective heuristics. These methods assume that past patterns, such as trends and seasonality, will persist into the future, making them effective when historical data accurately reflects underlying demand dynamics. Common analytical approaches include linear and exponential trend analyses, cyclical adjustments, and quantitative extrapolation techniques.

Traditional statistical methods such as the Autoregressive Integrated Moving Average (ARIMA) are among the most widely utilized. ARIMA models integrate

autoregressive (AR), moving average (MA), and differencing (I) components to handle non-stationary data, thereby making the series stationary before forecasting. The Box-Jenkins methodology remains a cornerstone in identifying optimal ARIMA configurations (Box and Jenkins, 1970; Box et al., 2015). Other traditional methods include state-space models such as Kalman filters and Structural Time-Series models, which dynamically adjust forecasts by incorporating latent variables, making them suitable for complex and evolving scenarios like financial markets (Harvey, 1990).

However, traditional statistical time series models, such as ARIMA and Exponential Smoothing (ETS), come with notable limitations. Most critically, they assume linear relationships in the data, which restricts their ability to capture complex, nonlinear dynamics often present in real-world applications such as financial markets or web traffic (Hyndman and Athanasopoulos, 2018). They also typically require the data to be stationary, which may necessitate differencing that can distort meaningful patterns or introduce instability (Box et al., 2015). Moreover, these models struggle to incorporate exogenous variables like macroeconomic indicators, marketing campaigns, product launches, or supply chain disruptions without cumbersome extensions (e.g., ARIMAX or dynamic regression), limiting their flexibility (Hyndman and Athanasopoulos, 2018).

Other drawbacks include the lack of native support for multivariate forecasting, inability to leverage metadata or contextual features without explicit feature engineering, and poor scalability to large datasets with many time series. Additionally, traditional models do not generalize across related series, treating each time series in isolation and missing opportunities for transfer learning (Makridakis et al., 2018). They also tend to underperform on sparse, intermittent, or irregularly spaced data, and exhibit increasing error with longer forecast horizons due to cumulative prediction errors (Hyndman and Athanasopoulos, 2018; Taylor and Letham, 2018). Finally, their rigid structural assumptions—such as predefined forms for trend and seasonality in ETS or fixed (p,d,q) configurations in ARIMA—can make them ill-suited for dynamically evolving systems.

To overcome these limitations, machine learning (ML) methods have gained prominence in recent decades (De Gooijer and Hyndman, 2006). Tree-based models like XGBoost and Random Forests excel at capturing complex feature interactions and handling high-dimensional data, including external predictors. Similarly, neural network-based approaches—particularly Long Short-Term Memory Networks (LSTMs)—have been widely adopted for their ability to learn long-term dependencies and model intricate temporal patterns (Hochreiter and Schmidhuber, 1997; Bai et al., 2018).

2.1.2 Judgmental Forecasting Methods

Judgemental forecasting methods are particularly valuable when historical data are limited or qualitative insights are more informative than quantitative data alone. Scenarios demanding judgemental forecasting typically involve unprecedented events, such as launching entirely new products or navigating regulatory changes. A prominent example occurred when the Australian government introduced plain packaging for cigarettes in December 2012, forcing forecasters to rely entirely on judgemental forecasting due to the absence of relevant historical data (Hyndman and Athanasopoulos, 2018; Francis, 2012).

Despite their practical relevance, judgemental forecasts have been criticized for their vulnerability to cognitive biases. Over four decades ago, Hogarth and Makridakis (1981) highlighted significant limitations, noting biases such as overconfidence, illusion of control, and seeing patterns in randomness. These biases have been shown to systematically undermine the accuracy of human judgement, primarily based on psychological laboratory experiments. Nonetheless, as Lawrence et al. (2006) observed, these laboratory findings might exaggerate actual biases occurring in practical forecasting contexts.

Real-world experiences illustrate the essential role of judgemental forecasting despite these criticisms. Reliance solely on automated analytical forecasts without

human oversight can lead to significant failures. For example, Nike's extensive reliance on purely algorithmic forecasting systems resulted in $400 million in inventory write-offs due to inadequate human intervention (Worten, 2003). Reported in the same article, Goodyear faced significant challenges with purely statistical forecasting methods, underscoring the necessity of managerial judgement to complement analytical techniques.

Judgemental methods are especially valuable when forecasters possess extensive domain expertise and have access to timely, relevant qualitative information. Economic forecasting and sales forecasting, for example, regularly incorporate judgemental insights such as anticipated promotional effects or competitive behaviors (Fildes and Stekler, 2002). Particularly in unstable or unpredictable scenarios, judgemental forecasting can surpass purely analytical models (Syntetos et al., 2016; Franses and Legerstee, 2011). Thus, judgemental forecasting tends to outperform analytical methods in the following conditions: (1) when historical data are sparse or non-existent, (2) when domain-specific expertise or contextual knowledge provides a clear advantage, and (3) in highly uncertain, unstable, or novel forecasting scenarios.

However, judgemental forecasting also faces inherent limitations, notably vulnerability to cognitive biases and variability in accuracy as Lawrence et al. (2006) observed. Evidence suggests that judgmental adjustments must be informed by genuine informational advantages over analytical methods; otherwise, such adjustments may introduce biases and degrade accuracy (Gupta, 1994; Sanders and Ritzman, 1995). Consequently, research increasingly advocates for integrating judgemental and statistical methods to achieve optimal forecasting performance.

2.1.3 Human-in-the-loop Ensembling: Integrating Analytical and Judgmental Forecasting

Given these findings, the optimal approach to forecasting leverages the complementary strengths of analytical and judgmental methods through (what we call) human-in-the-loop ensembling. Early research by Lawrence et al. (1985, 1986) supports this finding, demonstrating that integrating judgmental inputs with statistical models frequently yields improved forecast accuracy and reduced variability. Human-in-the-loop ensembling refers to systematically integrating expert judgment with analytical forecasts, creating a hybrid approach that combines data-driven rigor with context-aware human insights.

Brau et al. (2023) report that according to a research report by Association of Supply Chain Management, 83.6% of respondents indicated that they rely on integration of human judgement and analytical models for their forecasts. Blattberg and Hoch (1990), who first proposed equal weighted average between models and humans, summarized this well: "when models are weak (strong), humans are strong (weak)." Special events often require adaptability, personal judgment, and supplementary insights—conditions under which the human mind operates most effectively (Goodwin, 2002; Ibrahim et al., 2021). In contrast, analytical techniques outperform human judgment in low-variability settings, where consistent patterns and the systematic processing of data enhance forecasting accuracy (Lawrence et al., 2006; Sanders, 1992).

Various integration strategies have emerged within the literature (Arvan et al., 2019):

1. *Judgmental Adjustments*: Human experts adjust model-generated forecasts based on specialized domain knowledge, experiential insights, or qualitative information not adequately captured by models (Fildes and Petropoulos, 2015; Ibrahim et al., 2021).

2. *Quantitative Correction*: Systematically identifies and removes recurring biases from judgmental forecasts, enhancing reliability by isolating and quantifying these human biases (Fildes, 1991).

3. *Forecast Combination*: Simply averaging judgmental forecasts and statistical predictions, although empirical support for this method remains mixed and its application is limited (Blattberg and Hoch, 1990; Franses and Legerstee, 2013).

4. *Input to Model-Building*: Employs human judgment during the model development phase, influencing parameter selection or feature engineering, thus enriching analytical models with domain-specific intuition (Sanders and Ritzman, 2004; Green and Armstrong, 2012).

The effectiveness of these integration methods depends critically on the forecaster's expertise and the availability of contextual information (Arvan et al., 2019). For example, experienced forecasters making judgmental adjustments typically produce forecasts superior to simple moving averages (Alvarado-Valencia et al., 2017). Additionally, judgmental forecasting shows particular promise when predicting unstable or long-term demand patterns, precisely the conditions where statistical models alone often falter (Syntetos et al., 2016; Franses and Legerstee, 2011).

Therefore, this dissertation employs human-in-the-loop ensembling, particularly judgmental adjustments and integrative judgment learning, in the final stages of forecasting (as elaborated in Chapter 5). Through extensive application at a large-scale organization, we provide empirical evidence demonstrating that integrating human judgment with analytical forecasts markedly improves operational decision-making and performance. In doing so, this research contributes novel insights into how organizations can effectively utilize human-in-the-loop methods to maximize forecasting accuracy and strategic planning efficacy.

2.2 Machine Learning Based Demand Forecasting

With the proliferation of advanced computation hardware and ML technologies, organizations can harness these advanced tools to enhance both the accuracy and efficiency of their demand forecasting frameworks. The efficacy of such implementations depends on three fundamental components: (1) ensuring the quality and integrity of input data; (2) optimal selection and calibration of algorithms; and (3) comprehensive evaluation of forecasting accuracy, coupled with enabling planners to effectively utilize these sophisticated model outputs (Agrawal et al., 2020). While extensive research exists on various aspects of forecasting, particularly in model development and selection methodology, there remains a notable gap in the literature regarding practical implementation strategies. Table 2.1 provides a comprehensive overview of relevant literature in ML-based demand forecasting.

The methodology frameworks for demand forecasting have significantly evolved over the last few decades. Today, ML models can accommodate nonlinearity and handle a broader range of inputs, such as unstructured and high-dimensional data of various types. In recent years we have seen huge potential of ML algorithms in demand forecasting tasks due to their better data fitting capabilities.

Recent notable implementations include the following works that align with our objectives: Deng et al. (2023) outlined a comprehensive omnichannel retail infrastructure by Alibaba, which was the 2022 INFORMS Franz Edelman Award finalist. The infrastructure integrates demand forecasting with inventory management and price optimization, driven by product recommendations. Their implementation leverages deep learning models like DeepAR (Salinas et al., 2017), Prophet (Taylor and Letham, 2018), Wavenet (Oord et al., 2016), and N-BEATS (Oreshkin et al., 2019) to generate demand forecasts. Also, Dodin et al. (2023) showcased a pragmatic application of LightGBM models in forecasting the demand of parts at Bombardier. Similarly, Ferreira et al. (2016) utilized a regression tree-based model for demand forecasting in the pipeline for price optimization.

Table 2.1: Summary of related research papers on ML-based demand forecasting.

Reference	Input	Model	Evaluation Metric
Dodin et al. (2023)	Lagged demands, demand statistics, seasonality components, region and month index, average age of shipped products	Improved LightGBM, Elastic Net	RMSSE
Qi et al. (2023)	Lagged demand, inventory	End-to-end Model (Dynamic Programming, RNN, MLP)	Stockout rate, turnover rate, total inventory management, holding, and stockout costs
Deng et al. (2023)	Lagged demand, inventory, among others	DeepAR, N-BEATS, Prophet	WMAPE
Makridakis et al. (2018)	M-3 data	MLP, BNN, RBF, GRNN, KNN, CART, SVR, GP, RNN, LSTM, SES, ETS	sMAPE, MASE
Sagaert et al. (2018)	Lagged demand, macroeconomic indicators	LASSO Regression	MAPE
Hamzaçebi et al. (2009)	Lagged demand	Artificial Neural Networks (ANN)	SAE, SSE
Marcellino et al. (2006)	Lagged demand	Linear models	MSFE
Gardner (1990)	Lagged demand	Exponential-smoothing Model (ETS)	Investment and Delay Time

The Makridakis (M-series) competitions have served as a crucial test bed for evaluating diverse forecasting models, including support vector regression, classification and regression trees, Gaussian processes, Bayesian neural networks, K-nearest neighbor regression, generalized regression (kernel) networks, multilayer perceptrons, and radial basis function models (Makridakis and Hibon, 2000; Ahmed et al., 2010; Makridakis et al., 2018, 2021). LightGBM (Ke et al., 2017), which is an advanced tree-based model, is notable for its fast and efficient training and prediction, and was used in some form by all of the top-50 performers in the M-5 competition (Makridakis et al., 2022). LightGBM's accuracy has been validated by several other research studies for predictive modeling (Deng et al., 2021; Bandara et al., 2020; Zhang et al., 2020). Motivated by these studies, results from M-5 competition and our own experiments, we adopted the LightGBM algorithm for our task.

2.2.1 Direct vs Iterative Forecasting

Conventionally, two methods exist for regression-based time series prediction: (i) direct and (2) iterated forecasting method. The direct method uses separate models for each forecast horizon, while the iterated method predicts the next period and uses that estimate for subsequent forecasts. The choice between methods involves a bias-variance trade-off and depends on the unknown population projection (Findley, 1983). Theoretically, the direct method yields lower mean squared error, but its superiority in practice isn't guaranteed (McElroy, 2015). Empirical evidence in literature is conflicting: Marcellino et al. (2006) found the iterative method superior for long-lag specifications and longer horizons, while Hamzaçebi et al. (2009) observed better performance with the direct method using artificial neural networks. I direct interested readers to their literature reviews for more related works. With experimentation, we discovered superiority of iterated method in our case and thus use forecasted demand as a lagged input for subsequent predictions.

2.2.2 External Features

Incorporating additional data into ML-based forecasting models is beneficial to improve forecasting performance. For instance, Sagaert et al. (2018) leverage a broad set of macroeconomic indicators from the Federal Reserve Economic Data (FRED) in a LASSO regression model to improve tactical forecasting accuracy. In supply chain, private data creates information asymmetry; lack of information sharing hinders abilities to adequately harmonize manufacturer's activities to align with customers (Simatupang and Sridharan, 2002).

Information shared by suppliers and customers can also improve accuracy of demand forecasting. Hartzel and Wood (2017) show that demand forecasts benefit heavily from point-of-sale reporting. Kurtuluş et al. (2012) show that such forecast (called 'collaborative forecast') can be helpful for customers as well as suppliers, depending on the contractual obligations of both parties. Under the Newsvendor model setting, Taylor and Xiao (2010) show that the manufacturer benefits from selling to a better-forecasting retailer if and only if the retailer is already a good forecaster.

These studies guide us to use demand and inventory information reported by our supply chain partners as part of input to our forecasting model to further improve the forecasting performance.

2.3 MLOps in Large-Scale Machine Learning Deployments

Machine Learning Operations (MLOps) has emerged as a discipline to streamline the end-to-end lifecycle of ML models — from development and deployment to monitoring and maintenance — especially in enterprise settings. As organizations deploy large-scale ML models, they face unique challenges in infrastructure, integration, and project management. Research over the past decade highlights that building a

model is often the easy part; deploying and managing it at scale is far more complex (Sculley et al., 2015; Baylor et al., 2017). In this section, we examine research from the last decade when ML-based systems applications proliferated, specifically identifying *infrastructure challenges* and *MLOps solutions*, as well as project management strategies that help ensure these deployments succeed. We also discuss key case studies around ML demand forecasting applications documented in industry literature.

2.3.1 Challenges in Enterprise ML Deployment

Deploying ML models in production at enterprise scale introduces infrastructure and engineering challenges that extend beyond what data scientists typically encounter during development. In addition to selecting appropriate algorithms and necessary data inputs, significant implementation challenges exist in the deployment of ML algorithms in enterprise settings (Sinha and Lee, 2024). Researchers at Google discovered that while complex prediction systems can be useful, they often lead to massive ongoing maintenance costs, referred to as 'technical debt' (Sculley et al., 2015). They identified 'anti-patterns' resulting from 'glue code' (external dependencies leading to fixed downstream patterns, which freeze a system to peculiarities of a specific package) and 'pipeline jungles' (as new information sources are added incrementally, data preparation becomes a convoluted series of scrapes, joins, and sampling steps with intermediate outputs). In mature systems, machine learning code might constitute only 5% of the total codebase, with the remaining 95% being glue code (Sculley et al., 2015). In such cases, it may be less costly to create a clean native solution rather than reuse a generic package.

A recent survey of case studies by Paleyes et al. (2022) identified challenges in deploying ML projects at all stages: (a) data management, (b) model learning, (c) model verification, and (d) model deployment. They found that for ML models in production, integrating available infrastructure and implementing the model itself is

resource-intensive. Although some studies report on ML-based implementation of demand forecasting models in companies (Dodin et al., 2023; Ferreira et al., 2016), there are few detailed discussions on project management, deployment pipelines, and continuous performance monitoring specifically in demand forecasting.

To summarize, major challenges for deployment of ML projects identified in the literature include:

1. *Scalability of computing and data*: Scaling a prototype model to handle massive data volumes and high throughput in production is non-trivial. Handling high-volume data pipelines (streaming or batch from different sources) and ensuring timely processing demands careful system design (Baier et al., 2019). A common challenge is 'data drift': changes in data distributions or external trends that degrade model performance over time. This requires continuous model monitoring for drift and regular retraining to avoid stale predictions, which necessitates infrastructure for continuous data collection and model updating.

2. *Emerging complexity and technical debt*: ML systems incur "hidden technical debt" in ways traditional software with one-time deployment does not. Sculley et al. (2015) describe how seemingly simple ML pipelines can transform into "pipeline jungle" of glue code, entangled dependencies, data-quality issues, and configuration problems. Over time, maintaining these systems becomes expensive as data changes or external conditions shift, requiring continual adjustments. In practice, this means that without proper architecture, an ML model that was quick to develop can be difficult to reliably productionize and sustain.

3. *Heterogeneous Tools and Environments*: Large organizations use a variety of tools for data preparation, modeling, and serving, leading to fragmented environments. Amou Najafabadi et al. (2024) note that due to the large variety of available tools—from data extract, transform, and load operations and

Python packages to visual dashboard development options—organizations face a severe lack of consolidated architecture knowledge on integration strategies.

2.3.2 MLOps Solutions and Best Practices

While challenges have been identified, several solutions have been presented to the industry for addressing them. As expected, Google, due to its sheer scale of data and model management, has been at the forefront of this development (Sculley et al., 2015). One of the first innovations from Google's research was their seminal work on TensorFlow Extended (TFX) (Baylor et al., 2017). The basic premise of TFX is to replace ad-hoc scripts that have single objectives with a cohesive platform. In other words, TFX replaces glue code with an integrated system, significantly reducing duplicated efforts and technical debt, and cutting model deployment times from "months to weeks." Now, many more tools exist for pipeline orchestration such as Kubeflow (George and Saha, 2022), Apache Airflow (Haines, 2022), and MLFlow (Zaharia et al., 2018), that ensure data preprocessing, model training, and deployment steps are reproducible and automated.

Adapting development operations (DevOps) principles from software engineering, MLOps pipelines incorporate automated testing, integration, and deployment tailored to ML. This includes version control for datasets and models, unit and integration tests for data and model quality, and automated deployment triggers. Automating model retraining and redeployment (Continuous Training) is also advocated for scenarios where data evolves. Google's MLOps framework emphasizes CI/CD/CT pipelines that retrain models on fresh data and continuously roll out updates in a controlled manner (Tabassam, 2023; Alla et al., 2021).

Robust model and data versioning is a core part of MLOps solutions. This involves tracking dataset versions, model binaries, and hyperparameters used, often via a model registry or repository (Amou Najafabadi et al., 2024). Amou Najafabadi et al. (2024) provide examples showing that storing models with their metadata

(training data used, metrics, etc.) allows teams to trace results and revert to prior versions if needed. ML metadata stores log experiment details (like MLflow's experiment tracking database or Google's metadata store) to ensure reproducibility and auditability of ML experiments.

Once deployed, models require ongoing monitoring for performance and drift. MLOps best practices include setting up monitoring dashboards for prediction accuracy, data drift detection alerts, and capturing feedback. If a model's accuracy in production drops or data characteristics deviate from training data, alerts can trigger an investigation or an automatic retraining pipeline (Baier et al., 2019). Some architectures include a feedback loop where user or expert feedback on predictions is logged to a feedback database for use in model improvement for the next cycle (Amou Najafabadi et al., 2024).

2.3.3 Unified Platforms and Project Management

A trend in industry and research is creating unified MLOps platforms that address multiple stages of the lifecycle under one roof. Besides TFX, other platforms (often internal to tech companies) like Uber's Michelangelo (Uber Engineering, 2017) or Facebook's FBLearner Flow (Hazelwood et al., 2018) were built to provide end-to-end support—from data ingestion to model deployment—thereby lowering the barrier for large-scale deployments. The goal is to provide self-service infrastructure where data scientists can easily push models to production without reinventing infrastructure each time. By standardizing infrastructure and providing common services (feature stores, model serving endpoints, etc.), such platforms tackle the challenge noted by Baier et al. (2019) of needing standardized ML infrastructure across projects.

In our work, we use MLflow (Zaharia et al., 2018), an open-source platform for managing the ML lifecycle. MLflow's ability to be self-hosted is critical for us to ensure that sensitive data and model artifacts are not shared with external services.

In addition to privacy, it offers experiment tracking, model versioning, reproducibility, and seamless integration with multiple ML frameworks and deployment environments.

According to Mäkinen et al. (2021), successful large-scale ML deployments benefit from iterative methodologies that follow an evolutionary path. Rather than a long single development cycle, teams should use short iterations to continuously update models based on new data and feedback. Most ML applications start with experimental models (proof-of-concepts) and then evolve into frequent retraining and deployment cycles as they mature. They categorize organizations into stages: (1) exploring data, (2) building first models, and (3) managing many models with frequent updates—and note that full MLOps pipelines become essential only at stage (3) when continuous updates are needed.

Human and process factors are as crucial as technical ones in ML deployments. Large-scale ML deployments often involve many stakeholders (business leaders, domain experts, end users). Baier et al. (2019) found "appropriate communication and expectation management" to be an overarching challenge in ML projects. Project managers need to ensure that business stakeholders understand the iterative nature of ML (that models may need refinement and won't be 100% accurate initially). Setting the right success criteria and regularly demonstrating progress through reports or dashboards can help maintain support for the project. This approach of reporting incremental improvements through interactive dashboards has proven supremely helpful in gaining buy-in from planners.

Operationalizing forecasts presents its own set of challenges beyond model development. Forecasts by themselves don't create value until they inform decisions in procurement, production, staffing, and other operational areas. Many successful applications therefore link forecasting with optimization or simulation modules to directly drive action. The experience of Bombardier (Dodin et al., 2023) demonstrates the importance of an effective dashboard interface in this context. Their forecasting system's dashboard allows inventory planners to see the recommended forecasts from the ML/time-series ensemble along with the model's confidence levels. Because the

system was co-developed with the planning team, it outputs forecasts in formats and hierarchies the business uses (e.g., by part category, region), and updates predictions as new data arrive. This tight integration into Bombardier's ERP and inventory management system enables planners to act on forecasts with minimal additional effort, effectively transforming the system into a decision-support tool for inventory control.

Alibaba's evolution in inventory management illustrates the potential for fully automated decision-making systems (Liu et al., 2023). The e-commerce giant historically relied on human buyers (planners) who used algorithmic forecasts merely as recommendations. Their transition to a system where AI forecasts and decisions are fully automated effectively removed the final human override. Experimental results showed the algorithmic system consistently outperformed human planners by reducing out-of-stock rates while simultaneously cutting excess inventory. During the unpredictable demand spikes of the COVID-19 pandemic, human planners tended to overreact with panic-buying, exacerbating the bullwhip effect up the supply chain. The AI system responded more optimally by detecting changes in demand and supplier reliability without over-ordering, thereby mitigating bullwhip dynamics. Achieving this level of automation required significant trust in the system, extensive testing, and capabilities to handle exceptions such as supplier lead time variability. Alibaba's experience suggests that while fully automated forecasting and decision-making can excel in stable environments, organizations must carefully manage the transition, retraining staff for oversight roles and preparing for scenarios the AI wasn't trained to handle.

2.4 The Way Forward

Collectively, the literature underscores several central insights: (1) the hardest parts of ML in industry are often infrastructure and process, not algorithms; (2) solutions require a mix of technical tooling and organizational practices; and (3) as ML becomes

integral to business, the line between software engineering and data science blurs, demanding new hybrid approaches—which is essentially what MLOps embodies. Based on our own previous works (Curtland et al., 2022) and concepts of MLOps (Zaharia et al., 2018), establishing a comprehensive *Project Management Strategy* is valuable, as such issues are non-trivial in practice. Thus, Chapter 4 and Chapter 5 of this dissertation would be useful to researchers and practitioners looking to implement such systems at their organizations as we demonstrate how we achieved this with our demand forecasting system at HP.

Chapter 3

Forecasting Demand with Machine Learning

All models are wrong, but some are useful. — George Box

In this chapter, we define our key problem in demand forecasting, present our iterative forecasting algorithm and its components, detail the variety of inputs to our model and how we select them, and complete the results with various evaluation metrics and our model's performance. We show that our ML-based forecasts perform better than existing models in many cases, and as well as existing models in other cases, which is ideal for our use case of automating forecast creation for eventual selection between models by planners.

Demand forecasting is a critical component of supply chain management, directly influencing inventory levels, production planning, and ultimately customer satisfaction (Fildes et al., 2008). Traditional approaches to demand forecasting have relied heavily on statistical time series methods and human judgment, but these approaches often struggle with the increasing complexity and volatility of modern supply chains (Syntetos et al., 2016). Machine learning offers a promising alternative by leveraging multiple data sources and capturing complex non-linear relationships that statistical methods might miss (Carbonneau et al., 2008).

Our work focuses on developing and implementing a machine learning-based demand forecasting system that can complement existing statistical and consensus-based approaches. The key innovation lies in our ability to incorporate diverse features beyond historical demand, including product lifecycle information, channel metrics, and geographically specific indicators, while maintaining computational efficiency through careful algorithm selection and feature engineering. By iteratively forecasting demand and continuously updating our models, we create a robust system that adapts to changing market conditions while providing valuable insights to planners.

The detailed methodology, including the supervised learning algorithm, feature engineering strategies, model selection, and hyperparameter optimization, is based on our prior work accepted for publication in *INFORMS Journal of Applied Analytics* (Harshvardhan et al., 2025b). While this chapter expands those, the core methodological framework remains consistent with that publication.

3.1 Problem Definition and Formulation

We address the problem of predicting demand for a product p in a specific country c at time t. Given a dataset of historical demand data and other relevant information, our goal is to train a model that can forecast the demand for future time periods. The historical data includes information about the actual demand $y_{t,c,p}$ and a set of associated features $X_{t,c,p}$. These features represent various aspects of the time, market, and product, as well as lagged demand for up to 15 months prior to the forecasting month.

We formulate our forecasting problem as a supervised learning task where we aim to minimize the prediction loss over the dataset D, consisting of pairs of input features $X_{t,c,p}$ and corresponding demand values $y_{t+1,c,p}$:

$$D = \{(X_{t,c,p}, y_{t+1,c,p}) : \forall c, p, t_{first} \leq t < t_{now}\}, \tag{3.1}$$

where t_{first} is the first period when we have enough observations to create all features, especially the lagged features. The training process minimizes the forecasting loss (RMSE):

$$\ell(f|D) = \sqrt{\mathbb{E}_{X,y\in D}\left(f(X) - y\right)^2}, \tag{3.2}$$

in addition to necessary regularization terms.

In this context, our model $f(\cdot)$ learns to predict future demand based on the input features. Once trained, the model can be applied to forecast demand for future time periods $t \geq t_{now}$. We use $F_{t,c,p} \in \mathbb{R}^T$ to represent forecasts for T periods starting with t_{now}:

$$F^T_{t,c,p} = (\hat{y}_{t+1,c,p}, \cdots, \hat{y}_{t+T,c,p}). \tag{3.3}$$

To select the model for supervised learning, we rigorously evaluated many algorithms including XGBoost, LightGBM, Prophet, ARIMAX, ETS, and multilayer perceptrons, utilizing the Python `darts` library for unified and methodologically consistent comparisons (Herzen et al., 2022).* Our empirical evaluations, emphasizing predictive accuracy and computational efficiency, demonstrated clear superiority of tree-based models, specifically LightGBM (Ke et al., 2017). These models excel at capturing nonlinear relationships and complex data structures, making them effective for demand forecasting, while offering interpretability that outperforms other algorithms, with straightforward parameter optimization and a reduced memory footprint that simplify generalization and expedite training at scale. The results of the M5 competition reinforced our decision to use LightGBM, demonstrating its effectiveness on datasets with structural and computational complexities similar to ours (Makridakis et al., 2021, 2022). Given that the dataset in competition originated from Walmart and represented actual product data, LightGBM's applicability to real-world scenarios was further validated.

*Darts in Python: `https://unit8co.github.io/darts/`, accessed March 14, 2025.

3.2 Iterative Forecasting Algorithm

Our *Iterative Forecasting Algorithm* is outlined in Algorithm 1. It employs the LightGBM model as its core predictive engine, although it's adaptable to other algorithms. The model begins by preprocessing the data, which includes data cleaning and feature engineering. It is designed to forecast demand iteratively over a time window T, which allows for dynamically updating forecasts.

For each time step t_α, the algorithm constructs a training dataset D_α using all available data up to that point in time. Identified hyperparameters are used with D_α to train the LightGBM model $f(\cdot)$, which is optimized to minimize the Root Mean Squared Error (RMSE). Once trained, the model generates T future forecasts for each time step t_α. The LightGBM model is then either incrementally updated (i.e., warm started from best results from the previous month) or retrained from scratch, providing flexibility in handling significant changes in underlying data distribution.

We optimize our LightGBM model's hyperparameters using Hyperopt, a library that efficiently explores both discrete and continuous parameter spaces (Bergstra et al., 2013). Using the last month's data for validation, we employ Hyperopt's Tree of Parzen Estimators (TPE) algorithm to navigate this parameter space. This Bayesian hyperparameter optimization allows for faster convergence to optimal configurations by focusing on hyperparameter values that maximize performance on the validation set. By leveraging Hyperopt's capabilities, we can balance exploration of the search space with the exploitation of promising configurations, ensuring our LightGBM model is finely tuned for optimal performance.

Specifically, we tune the following key parameters in LightGBM:

1. **Learning Rate:** Controls how much to adjust the model with each step, with a range between 0.1 and 1.

2. **Maximum tree depth:** Dictates the maximum depth of each decision tree, explored between 10 and 100.

Algorithm 1 Enhanced training and forecasting algorithm with LightGBM

1: **Preprocess the data:** Data cleaning and feature engineering.
2: **Determine optimal hyperparameters:** Use Hyperopt for the LightGBM model.
3: **Initialize forecast horizon** T (e.g., 7).
4: **for** t_α in $(t_{\text{first}} : t_{\text{now}})$ **do**
5: Create the training data:

$$D_\alpha = \{(X_{t,c,p}, y_{t,c,p}) : \forall c, p, t_{\text{first}} \leq t \leq t_\alpha\}$$

6: Perform time-series cross-validation on D_α and train the LightGBM model $f(\cdot)$ with optimal hyperparameters, minimizing loss (RMSE):

$$\ell(f|D) = \sqrt{\mathbb{E}_{X,y\in D}\left(f(X) - y\right)^2}$$

7: With the fitted model, create T forecasts for $t_\alpha + 1$ to $t_\alpha + T$:

$$F^T_{t_\alpha,c,p} = \left(f(\hat{X}_{t_\alpha+1,c,p}), f(\hat{X}_{t_\alpha+2,c,p}), \cdots, f(\hat{X}_{t_\alpha+T,c,p})\right)$$

8: Update the LightGBM model incrementally by warm starting from last month's best results if possible, or retrain it from scratch.
9: **end for**
10: **Perform Backtesting:** Apply the trained model to a historical dataset $D_{\text{historical}}$ to simulate past predictions. Evaluate its performance using appropriate metrics.
11: **Store Forecasts:** Save the generated forecasts $F^T_{t_\alpha,c,p}$ to a dedicated database or file storage for future evaluation, comparison, or direct usage.
12: **Log Model:** Serialize the LightGBM model, hyperparameters, and performance metrics for future reference or retraining using MLFlow.

3. **Regularization parameters:** L_1 and L_2 regularization terms help prevent overfitting, with values explored between 0 and 1.

4. **Minimum child weight:** Specifies the minimum sum of instance weights needed in a child, ranging from 1 to 50.

5. **Subsample and column-sample proportion:** Controls the fraction of samples and features used per tree, ranging from 0.5 to 1.

Our approach allows us to capture both the seasonality and trends in the demand while benefiting from the efficiency and scalability of LightGBM. Moreover, the iterative nature of this algorithm allows for frequent model updating, leveraging the most recent one-month data for cross-validation. This ensures that the model stays responsive to any significant changes in the underlying data patterns. Storing the serialized model in MLFlow, we are able to ensure repeatability and continuity for future efforts, as detailed later in Chapter 4.

3.3 LightGBM: A Succinct Summary

In this section, we provide a concise overview of the LightGBM model, developed by Microsoft. For comprehensive details, see Ke et al. (2017).

LightGBM represents an advanced implementation of Gradient Boosting Machines (GBM), which build ensembles of weak models to create strong predictive models (Friedman, 2001). Each subsequent model is trained to correct errors made by its predecessors by minimizing the loss function's gradient, i.e. boosting on weak learners on eventually have a strong learner. The final model takes the form $F(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$, where $h_m(x)$ represents the m-th weak learner, γ_m its weight, and M the total number of learners. At each iteration, a new tree $h_m(x)$ is added to minimize the loss: $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$.

The exceptional efficiency of LightGBM stems from two key innovations in its boosting algorithm: Gradient-based One-Side Sampling (GOSS) and Exclusive

Feature Bundling (EFB). GOSS prioritizes instances with larger gradients while randomly sampling those with smaller gradients, preserving accuracy while reducing computational load. EFB combines mutually exclusive features, particularly effective for high-dimensional sparse datasets.

LightGBM employs decision trees as base learners with each internal node representing a decision point and each leaf a prediction value. Unlike traditional level-wise growth algorithms, LightGBM implements leaf-wise growth, selecting the leaf with maximum delta loss for expansion. This approach creates asymmetric trees that can be deeper for complex cases while remaining shallow for simpler ones, balancing accuracy and overfitting prevention.

The model is particularly well-suited for demand forecasting as it naturally handles heterogeneous features (categorical, numerical, and time-based variables), captures complex non-linear relationships, provides native feature importance assessment, and efficiently processes missing values without extensive preprocessing. Its lightweight architecture enables rapid training and frequent updates, while various regularization options ensure generalization to unseen demand patterns.

3.4 Model Input Features

Our ML models distinguish themselves from conventional time series models by their ability to incorporate a diverse set of features including binary, categorical, and numeric — most traditional time-series models cannot use exogenous features. These features are carefully selected to not only capture historical demand data but also to offer insights into the multifaceted nature of demand generation and fulfillment. The features are listed below and their summary is provided in Table 3.1 for ease of reference. In total, we had around 147 input features to use with our model.

1. *Lag Demands:* Demand from the previous m months are factored in, with $m = 15$ for products with intermittent demand and annual buying cycles.

2. *Rolling Demand Features:* These are statistical measures—mean, coefficient of variation, and outlier counts—computed over rolling windows of 3, 6, and 12 months, capturing both recency and variability in demand.

3. *Product and Geography-based Statistics:* Summary statistics are categorized by product and geography to model unique trends and attributes within these dimensions.

4. *Seasonal Fluctuations:* Binary indicators for each fiscal quarter are included to capture seasonal demand patterns. Additionally, a monthly integer representing month of the quarter is also included.

5. *Product Life Cycle (PLC):* Calculated as $(M - m)/M$, where M is the total expected lifetime of product, and m is the current forecasting month, this feature considers a product's remaining lifespan, enriching the model's temporal context. Typically, products introduced to the market experience a surge in demand initially, attributable to their innovative features and promotional efforts, followed by a gradual decline in sales as they progress through their product life cycle.

6. *Channel Metrics:* Features such as 'Channel Partner Inventory' and 'Sell-through' provide a nuanced understanding of real-time market demand and potential future orders with direct inputs from our distribution channel partners (customers in B2B setting). Channel partner inventory refers to the SKU-level inventory that our channel partners report monthly, while Sell-through represents the sales by our partners to their customers.

3.4.1 Feature Selection

We conducted a comprehensive evaluation of two advanced feature selection algorithms for our machine learning pipeline: the hierarchical clustering approach

Table 3.1: Summary of Forecasting Model Input Features and Their Utility

Feature Name	**Description**	**Granularity**	**Utility for Forecasting**
Lagged Demand	Size of demand from previous m months, m varies per product group	Month (t)	Captures influence of past trends on future demand
Rolling Demand Features	Statistics of demand within an n-month rolling window (mean, coefficient of variation, outliers)	Month (t)	Assesses recent trend and variability
Product-based Statistics	Mean and coefficient of variation of lagged demand and rolling features, per product category	SKU (p)	Identifies category-specific demand trends
Geography-based Statistics	Mean and coefficient of variation of lagged demand and rolling features, per country	Country (c)	Captures location-specific demand patterns
Seasonal Fluctuation	Binary indicator for each fiscal quarter and integer month within a quarter	Month (t)	Accounts for seasonal variations in demand
Product Life Cycle	Proportion of product life cycle left, calculated as $(M - m)/M$	SKU, Country (p, c)	Determines stage of the product in its life cycle
Channel Inventory	Inventory levels reported by distribution channel partners	SKU, Country, Month (p, c, t)	Indicates potential reordering needs
Sell-through	Sales to distribution channel partners	SKU, Country, Month (p, c, t)	Reflects downstream demand at distribution level

popularized by Howard (2019) and the Quadratic Programming Feature Selection (QPFS) technique introduced by Rodriguez-Lujan et al. (2010).

FastAI Method of Feature Selection

The Fast AI feature selection methodology follows a structured process beginning with the computation of a Pearson correlation matrix across all 147 initial features in our dataset, resulting in a comprehensive 147×147 similarity matrix. Using the correlation coefficients as similarity metrics, we constructed a hierarchical cluster dendrogram of features using Ward's minimum variance method (Ward Jr, 1963). This dendrogram visually represents feature clusters based on their interdependencies. From each major branch of the dendrogram, we selected the feature with the highest univariate importance score (measured using mutual information with respect to the target variable), while pruning highly correlated features (correlation coefficient > 0.85) within the same cluster. We employed recursive feature elimination with cross-validation (RFECV) as a final refinement step, using our base model to verify the optimal feature subset size. When two features were comparable in their impact to accuracy, we valued expert judgement on what contributed better to the overall model. This methodology effectively handles multicollinearity while preserving important predictive signals across feature clusters.

Quadratic Programming Feature Selection

The QPFS approach of feature selection by Rodriguez-Lujan et al. (2010) formulates feature selection as a quadratic optimization problem. Feature selection is expressed as a minimization problem:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - F^T\alpha,$$

subject to: $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i = 1$, where Q represents the feature similarity matrix, F denotes the feature relevance vector, and α is the weight vector to be optimized. The matrix Q can be interpreted as a mutual information matrix or correlation matrix

between features. The vector F represents the correlation of input features with the dependent variable, or mutual information between the features and the dependent variable. We employed the interior-point method to solve the constrained quadratic programming problem. A regularization parameter $\mu = 0.5$ was used to balance feature relevance against redundancy equally.

Comparative Analysis between Feature Selection Methods

For our comparative analysis, we designed experiments where we ran prediction forecasting across multiple monthly cycles and different geographical regions to evaluate both the performance and stability of selected features. We tracked model accuracy alongside the consistency of feature selection for each algorithm. This temporal and spatial cross-validation approach provided insights into how feature selection would behave in real-world deployment scenarios where models are retrained periodically with evolving data distributions.

Our stability analysis involved tracking the selected feature sets across different time periods and regions. The Fast AI method demonstrated significantly higher stability compared to QPFS. QPFS exhibited high variance in feature importance rankings, while the Fast AI method produced more consistent rankings.

In terms of model performance, both methods achieved comparable predictive accuracy when evaluated using our LightGBM based Algorithm 1, particularly when we replaced the underlying model from FLAML (used for experiments) to LightGBM. While QPFS provides theoretical guarantees on the accuracy trade-off when reducing feature dimensionality, our experiments revealed a critical practical limitation. Furthermore, there were business concerns that the QPFS method selected substantially different feature sets for each time period and geographic region. This inconsistency would make model interpretation challenging for business stakeholders who need to understand and trust the forecasting model's decision factors. Our team specifically noted that having a stable set of features across different forecasting cycles was essential for building confidence in the model and incorporating its insights

into their operational decisions. Thus, based on our comprehensive analysis, we implemented the Fast AI method in our production code primarily due to its feature selection stability across different forecasting periods and regions. Importantly, our pipeline incorporates an expert review phase where planners validate and refine the algorithmically selected features based on their domain knowledge.

3.5 Performance Evaluation

The ultimate adoption of our new ML forecasting pipeline hinges on its accuracy. We validate the performance of ML-based forecasts against existing Statistical and Consensus forecasts, serving two critical purposes. Firstly, before enterprise-wide deployment across products and geographies, we must demonstrate that the ML pipeline's accuracy and reliability meets or exceeds that of current methods. Secondly, we must also evaluate the judicious use of the additional project management machinery which requires significant investment. Successfully achieving the first goal justifies the allocation of these additional resources.

3.5.1 Evaluation Metrics

Bias Bias measures the weighted percentage error in forecasts, signified by a positive or negative value indicating over or underforecasting, respectively. Bias is calculated using the formula:

$$\text{Bias} = \sum_{i=1}^{n} \frac{y_i - \hat{y}_i}{y_i} \tag{3.4}$$

Weighted Mean Absolute Percentage Error (wMAPE) wMAPE represents the weighted mean of absolute percentage errors, a metric easily understood even by non-technical stakeholders as percentage deviation from actuals. It is expressed as:

$$\text{wMAPE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} y_i} \tag{3.5}$$

Root Mean Squared Error (RMSE) RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{3.6}$$

RMSE, our preferred metric for ML model training, is symmetric and continuously differentiable. It balances sensitivity to larger errors with scale dependency, making it valuable for emphasizing significant deviations. However, due to RMSE's sensitivity to outliers, models trained with this metric may prioritize minimizing larger errors which, like in our case, can occasionally result in underforecasting. Planners and managers primarily use Bias and wMAPE as key performance indicators (KPIs) due to their ease of interpretation and actionability. Bias is easiest to fix in planning system; high (low) bias implies requirement to reduce (increase) forecasts and inventory. For a comprehensive comparison of these and other accuracy metrics, including their application in M-3 forecasting, we refer readers to Hyndman and Koehler (2006).

These metrics are calculated over a specified number of months, denoted as CM-k, where k represents the number of months. For a given month t, the k-month cumulative actuals are calculated as $\sum_{i=t}^{t+k-1} y_i$, while the cumulative forecasts are $\sum_{i=t}^{t+k-1} \hat{y}_i$. For example, three-month cumulative forecast (CM-3) starting in January would sum the forecasts for January, February, and March.

Measuring and improving the forecast over different lead time horizons is important due to practical business reasons. Supply chains have specific lead times for manufacturing and shipping products, and businesses maintain inventory close to customers to manage demand variability during these periods. Cumulative forecasts are preferred over point forecasts because they more effectively manage lead time variability. In an optimized supply chain, this approach allows for better inventory pooling and more accurate adjustment of factory capacity based on appropriate lead times and forecast performance. The choice of cumulative forecast horizons depends on specific supply chain lengths and decision-making requirements. CM-1, CM-3,

and CM-6 forecasts are commonly reported, with CM-3 often being the most critical due to its alignment with the typical three-month production lead time. On the other hand, CM-1 provides immediate feedback on short-term operations, while CM-6 offers a longer-term outlook.

3.5.2 Results

We present forecasting performance for a select business segment (1,484 products) from all three methods: Consensus (ConsFcst), Statistical (StatFcst), and ML (MLFcst), evaluated at cumulative horizons of one (CM1), three (CM3), and six (CM6) months. Although the scales have been adjusted for anonymity, the observed trends remain the same. Results from all product lines are not presented due to data sensitivity, and accuracy results vary across business segments.

A summary of accuracy results are provided in Table 3.2. These metrics are also presented as a dumbbell plot in Figure 3.1 with center point as 12-month averages and whiskers indicating one standard deviation. Additionally, Figure 3.2 visualizes these metrics over all 12 months, highlighting the monthly accuracy trends for each method. Finally, a statistical comparison of metrics over 12 months using paired t-test is presented in Table 3.3.

The ML forecast method demonstrates considerable strengths in its forecasting accuracy as compared to the statistical method, particularly in the metrics of wMAPE and RMSE. We observe that wMAPE for ML forecast is better than the other two in all three cumulative periods. In fact, at CM3 and CM6, that is for longer range forecasts, our model has wMAPE almost half of the other two methods. When looking at statistically significant differences, we find statistically significant difference between ML and STAT models with positive t-statistic and p-values less than 0.05. These findings strongly suggest the statistical superiority of the ML forecast in wMAPE, further demonstrating the model's alignment with HP's business objectives, since wMAPE is a business KPI. The higher accuracy of ML model in wMAPE is

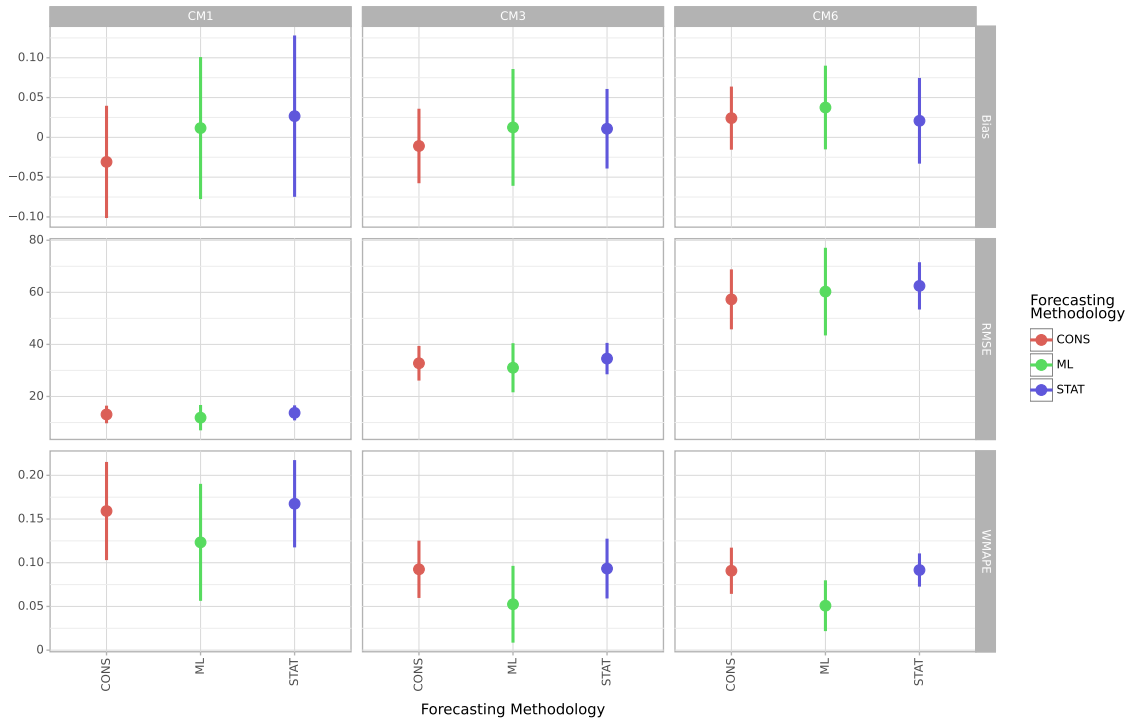


Figure 3.1: Dumbbell plot visualizing the mean (center point) and one standard deviation (vertical lines) of Bias, RMSE, and WMAPE for three forecasting methods (Consensus, Machine Learning, and Statistical) over cumulative forecast horizons of one month (CM1), three months (CM3), and six months (CM6).

Table 3.2: Forecasting accuracy metrics (bias, RMSE, and wMAPE) for cumulative forecast horizons (CM1, CM3, CM6) with Mean (Standard Deviation).

Model	CM1			CM3			CM6		
Metric	Bias	RMSE	wMAPE	Bias	RMSE	wMAPE	Bias	RMSE	wMAPE
Consensus	-3.08%	13.09	15.92%	-1.08%	32.76	9.25%	2.42%	57.29	9.08%
	(7.05%)	(3.38)	(5.62%)	(4.68%)	(6.66)	(3.28%)	(3.96%)	(11.51)	(2.64%)
ML	1.17%	11.87	12.33%	1.25%	31.03	5.25%	3.75%	60.28	5.08%
	(8.92%)	(4.87)	(6.69%)	(7.34%)	(9.43)	(4.39%)	(5.26%)	(16.83)	(2.91%)
Statistical	2.67%	13.71	16.75%	1.08%	34.55	9.33%	2.08%	62.47	9.17%
	(10.14%)	(2.89)	(4.99%)	(5.00%)	(6.03)	(3.42%)	(5.38%)	(9.08)	(1.90%)

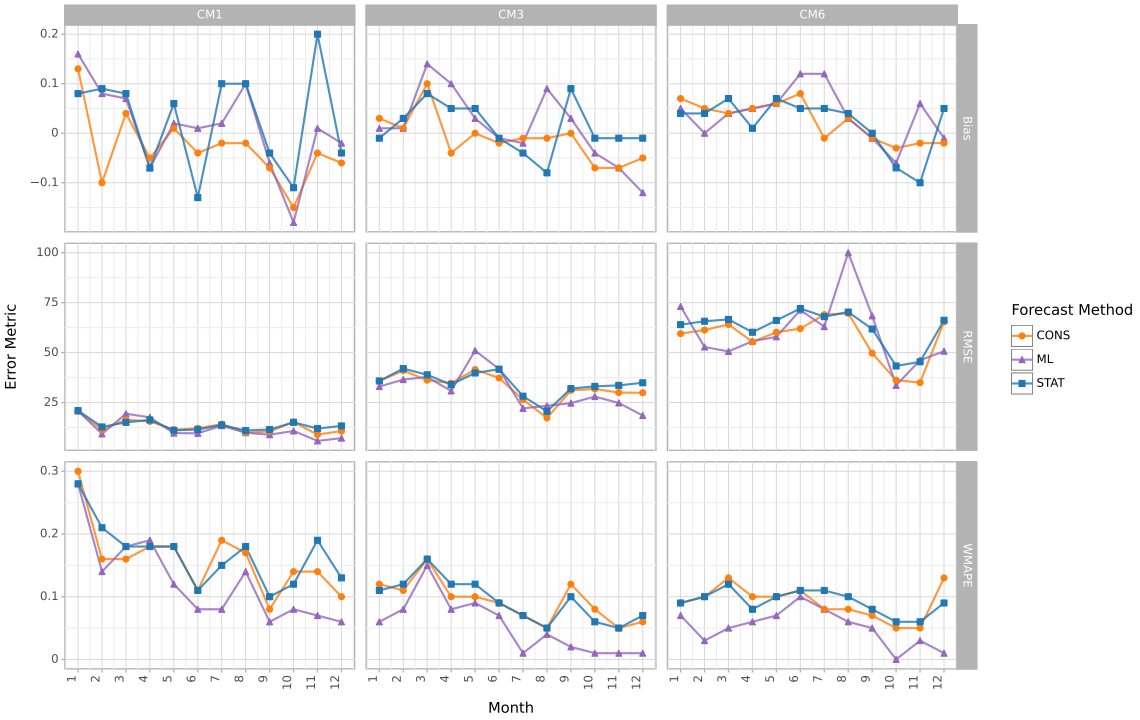


Figure 3.2: Bias, WMAPE, and RMSE metrics over 12 months show that the ML model is consistently among the top performers of the three models. CM1 is point forecast, while CM3 and CM6 are three and six months cumulative forecasts, respectively.

Table 3.3: Forecasting Accuracy Metrics: Bias, wMAPE, RMSE Comparison for CONS, ML, and STAT Methods.

Cumulative	Comparison	Bias	RMSE	WMAPE
CM1	CONS vs ML	-1.295 (0.209)	0.716 (0.482)	1.421 (0.169)
	STAT vs ML	0.385 (0.704)	1.128 (0.272)	1.832 (0.080)
CM3	CONS vs ML	-0.929 (0.363)	0.518 (0.610)	2.528 (0.019)
	STAT vs ML	-0.065 (0.949)	1.089 (0.288)	2.541 (0.019)
CM6	CONS vs ML	-0.701 (0.490)	-0.507 (0.617)	3.526 (0.002)
	STAT vs ML	-0.767 (0.451)	0.399 (0.694)	4.074 (0.001)

Note: The accompanying table presents t-statistics and p-values (in brackets) for an in-depth assessment across various cumulative forecast horizons.

particularly surprising since it was trained with RMSE as the loss function. In the case of RMSE, which is sensitive to large forecast errors, the ML forecast again proves to be more adept than others, though not statistically significant.

However, the ML forecast does not consistently dominate across all metrics and comparisons. When considering Bias, which reflects the systematic error in forecasts (either as overestimation or underestimation), the ML method does not exhibit a statistically significant difference from the statistical or consensus forecasts in any of the cumulative periods (CM1, CM3, and CM6), as evidenced by p-values greater than 0.05. Our model exhibits higher bias compared to the Consensus and Statistical models. We observed a strong tendency for the ML model to underforecast, particularly over longer time horizons. This issue appears to be influenced by the intermittent demand of many products, where the model occasionally learns to forecast zero incorrectly. While this may explain the underforecasting, further investigation is required to definitively identify the root cause.

These results suggest that, in certain scenarios — particularly those involving longer-term predictions — the Consensus forecast may provide more accurate outcomes than our method. This contrast underscores the ML forecast's strengths in specific contexts, guiding the modeling team in targeting improvements and enabling the business team to select the best-performing model for each product and country. By acting as a "human in the loop", the business team plays a crucial role in validating and verifying forecasts generated by the automated model. The data in Table 3.3 and the trends in Figure 3.2 collectively bolster the case for adopting the ML model alongside the Statistical and Consensus models at HP, contributing to an integrated effort aimed at improving overall forecasting performance.

3.6 Conclusion and Future Work

In this chapter, we introduced a comprehensive machine learning (ML) approach to demand forecasting that effectively competes with, and frequently outperforms,

traditional statistical and consensus-based methods. Our LightGBM-based model (Algorithm 1) demonstrates particular strength in weighted Mean Absolute Percentage Error (wMAPE) for medium- and long-term forecasts, primarily due to its ability to incorporate diverse features and capture complex non-linear relationships. This strength makes it especially valuable for practical supply chain applications.

The iterative forecasting algorithm we developed is modular, allowing continuous model updates and ensuring forecasts remain relevant even if the underlying model architecture transitions from LightGBM to another approach. Our diverse feature set—including lagged demands, rolling statistics, seasonal indicators, product lifecycle data, and channel metrics—richly captures the multitude of factors influencing demand. A meticulous feature-selection process ensures the resulting features are stable, informative, and free of redundancy.

The successful performance of our ML-based forecasting approach validates the significant investment required for developing robust project management and MLOps infrastructure to support its production deployment. This extensive infrastructure, described in detail in the next chapter (Chapter 4), facilitates accurate and timely forecasts aligned with critical business KPIs. Ultimately, this system supports enhanced decision-making throughout the supply chain, driving improvements in inventory management, production efficiency, and customer satisfaction.

Despite these strengths, our approach has some limitations. A notable drawback is the tendency to underforecast, especially for products characterized by intermittent demand, which necessitates further investigation and potential model refinements. Additionally, while the ML model excels at reducing wMAPE, it does not consistently outperform other methods across all metrics. This indicates that an ensemble approach may be most effective in practical scenarios. Here, our 'human-in-the-loop' strategy, allowing planners to select the best forecast among analytical methods (statistical or ML-based) or their own consensus forecasts, provides a notable advantage. This concept is further elaborated in Chapter 5.

Several promising directions emerge for future modelling research. One path involves exploring hybrid models based on alternative architectures, such as Long Short-Term Memory (LSTM) networks, which can seamlessly integrate into our iterative forecasting algorithm. Improving the model's handling of intermittent demand patterns to reduce bias and mitigate underforecasting also represents a critical area for enhancement. Incorporating external factors—such as macroeconomic indicators, competitor activities, and market sentiment—could further enrich the input feature set, helping the ML model better capture broad economic trends influencing demand.

Additionally, extending forecast horizons beyond the short-to-medium term (1–12 months) to include long-term horizons (up to 18 months) without sacrificing accuracy would deliver strategic insights valuable to business planning.

Chapter 4

MLOps: Machine Learning Operations

In theory, there is no difference between theory and practice. But in practice, there is. — Yogi Berra

Large-scale machine learning projects with numerous collaborators and users necessitate robust coordination and maintenance tools. In our implementation of ML forecasting, we developed a comprehensive machine learning operations (MLOps) framework that not only addressed immediate operational needs but established a foundation for long-term scalability and knowledge retention. To enhance value creation and streamline the entire ML project life cycle, we repurposed several DevOps concepts as MLOps, a state-of-the-art practice in scalable machine learning (Mboweni et al., 2022). This adaptation was necessary because traditional software development practices fail to address ML-specific challenges such as data drift, model decay, and the inherent experimental nature of model development. Figure 4.1 shows the basic working system of MLOps.

John et al. (2021) emphasize the indispensability of MLOps frameworks for ensuring robustness in machine learning workflows. These frameworks facilitate rigorous tracking of data lineage by meticulously documenting data sources, transformations,

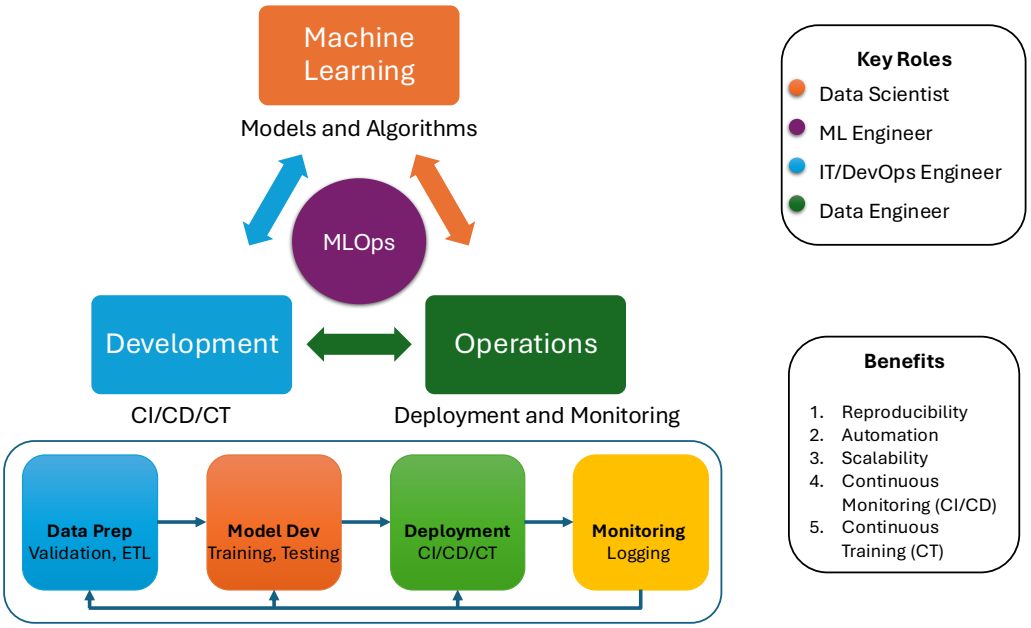


Figure 4.1: Figure of integrated MLOps framework showing the intersection of Machine Learning, Development, and Operations with a continuous workflow pipeline and key roles and benefits highlighted.

and feature engineering steps, thereby enhancing transparency and reproducibility. Additionally, they enable systematic validation of ML models through standardized evaluation protocols and metrics, ensuring consistency in performance assessment. Controlled release of models is another critical aspect, achieved via global backtesting of models before pushing to production, which mitigate risks associated with model updates. Furthermore, MLOps frameworks support the comprehensive storage of serialized models, allowing for seamless replication and future warmstarting of training process, thus fostering long-term sustainability in machine learning operations.

The MLOps infrastructure has been detailed in our accepted publications at the *INFORMS Journal of Applied Analytics* (Harshvardhan et al., 2025b) and *Foresight* (Harshvardhan et al., 2025a). While this chapter adds many details to our prior works, the overall idea remains consistent with those publications.

In Table 4.1, we outline the core components that drive MLOps, akin to the essential wheels of a well-functioning machine.

4.1 Ensuring Reproducible Experimentation

Our ML enhancements were primarily driven by systematic experimentation, unlike traditional software development, which follows a fixed specification. Instead, our process required exploring multiple dimensions simultaneously. We experimented with various dataset variations, including different time windows, feature combinations, and preprocessing techniques. Additionally, we tested diverse variable transformations such as normalization methods, encoding strategies, and feature interaction generation. Model exploration spanned gradient-boosted trees to neural networks with varying layer configurations, comparing performance across different software library implementations using Python's `darts` package. Various hyperparameter combinations, as detailed in the previous chapter, were also evaluated.

Table 4.1: Summary of MLOps Principles and Implementation Components

Principle	Description	Key Benefits	Tools Implemented
Experimentation Tracking	Systematic logging of experiments with unique identifiers, version control, and detailed metadata to ensure reproducibility	Transparency in model development; Historical comparison of approaches; Efficient knowledge transfer	MLflow; Git/Version Control
Model Registry	Centralized repository for tracking model versions, transitions between environments, and performance metrics	Controlled model deployment; Versioning for rollback capability; Performance comparison	MLflow Registry; Custom metadata tags
Data Lineage Tracking	Documentation of data sources, transformations, and feature engineering steps	Enhanced transparency; Reproducibility of results; Simplified debugging	MLflow; Version-controlled datasets
Notebook Orchestration	Converting notebooks into function-like components with standardized inputs/outputs and templated structures	Consistency across experiments; Parameterized execution; Simplified collaboration	Jupyter Notebooks; Papermill library
Code Navigation System	Strategic keyword placement (e.g., `Monkey`) to facilitate easy identification of critical configuration points	Efficient onboarding; Reduced errors in recurring tasks; Streamlined maintenance	Comment keywords; Standardized code organization
Optimized Data Storage	Selection of appropriate file formats based on performance characteristics for ML datasets	Reduced storage costs; Improved processing efficiency; Enhanced reproducibility	Apache Feather; Parquet (alternative)
Rapid Experimentation	Multi-stage testing process using automatic machine learning for efficient hypothesis validation	Resource optimization; Faster iteration cycles; Prioritization of promising approaches	FLAML; Experimenting on data subsets; Cost-aware optimization
Model Serialization	Storage of trained models for future use, including warm-starting, reproduction, and comparison	Reduced computation time; Audit capability; Parallel testing	MLflow model artifacts; Binary serialization
Continuous Deployment	Pipeline for regular model updates and deployment to production	Systematic model refreshes; Consistent deployment process; Reduced operational risk	Project management workflow; MLflow; Automated pipelines

Tracking the effectiveness of these strategies was crucial, as some yielded significant improvements while others had minimal impact. To maintain structured experimentation, we assigned each experiment a unique identifier, tracked it with version control, and documented metadata using MLFlow (Zaharia et al., 2018).

Given the dependency of model performance on input data and training, *reproducibility* was paramount. Each month, a selected model was deployed to generate ML forecasts for operational decisions, while ongoing experimentation continued to refine models for future use. Our project management strategy is illustrated in Figure 4.2.

4.2 MLFlow

We adopted various open-source tools in our reproducibility strategy, each selected after careful evaluation of alternatives. MLflow formed the cornerstone of our experimentation and reproducibility infrastructure. This open-source ML platform (Zaharia et al., 2018) addresses challenges linked to experimentation, reproducibility, and deployment. We selected MLflow for several key reasons: its self-hosting capabilities allowed complete control over our sensitive data without external dependencies; the flexible tracking API supported both automated logging and metric registration; seamless artifact storage integration connected with our existing storage infrastructure; and the accompanying Python package enabled programmatic querying of experimental results which we useful for evaluation and expert feedback.

Our MLflow implementation configured experiment locations with artifact storage paths and employed a nested run structure for model development. We leveraged MLflow's four core components: Tracking for logging metrics, parameters, artifacts, and code versions; Projects for ensuring reproducible runs with consistent environments; Models for standardized packaging and deployment; and Registry for managing model versions through various stages.* We logged comprehensive

*MLFlow Core Concepts: https://mlflow.org/

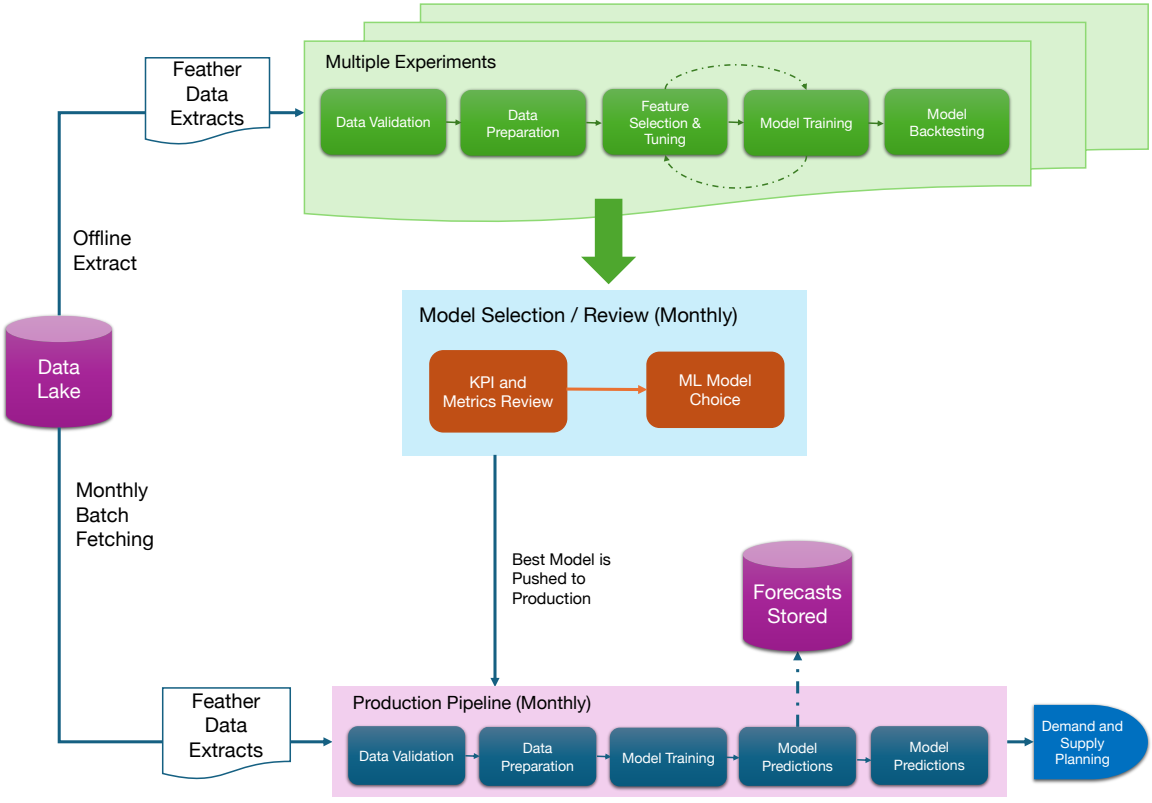


Figure 4.2: Project management for continuous deployment pipeline of our ML forecasting efforts.

metadata including data version, feature sets, and input data characteristics such as row counts and missing value percentages. For hyperparameter variations, we created nested runs that tracked specific parameters, trained models with those configurations, logged performance metrics including RMSE, number of iterations, and training time, saved model artifacts, and generated visualizations such as feature importance plots. This implementation allowed us to track over 200 experiments, with full lineage from data inputs and package versions to prediction outputs and accuracy metrics.

MLflow's model registry provided a centralized repository for tracking model versions, transitions from staging to production, and performance metrics. We extended this with custom metadata tags for business geographies and segments. The serialized models served multiple crucial reproducibility purposes: warm-starting future training by initializing new models with previous weights reduced computational time by several hours in our experiments (recall that each production run took over eight hours); result reproduction enabled exact recreation of historical forecasts for audit purposes; and experimentation allowed simultaneous testing of multiple model versions to compare different modeling techniques.

4.3 Notebook Orchestration and Reproducibility Framework

Jupyter Notebooks served as another pillar in our reproducibility strategy, allowing detailed annotations on processes, inputs, and outputs using markdown cells. Beyond basic documentation, we implemented advanced notebook orchestration techniques. We employed notebook parameterization, converting notebooks into function-like components with standardized inputs/outputs using the `papermill` library. Our template structure established consistent notebook sections for configuration, data loading, preprocessing, modeling, evaluation, and visualization. Our parameterized

notebook execution used the papermill library to run template notebooks with specific configurations for data paths, model paths, forecast horizons, confidence intervals, and execution dates. See Figure 4.3 for an overview of the `papermill` library and its key benefits.

We further enhanced our workflow with a code navigation system that implemented a keyword system using strategically placed comments. Specifically, we incorporated the keyword `Monkey` into our codebase to facilitate quick navigation for necessary adjustments before re-running routine scripts. For example, we added comments like:

```
# MONKEY: Update date ranges for monthly retraining
start_month = "2021-01-01"
end_month = "2024-08-01"

# MONKEY: Update feature list if new features are added
feature_list = ["feature1", "feature2", "feature3"]
```

This approach simplified the identification of key areas for updates, enhancing efficiency in recurring tasks like monthly time-series forecasting. By searching for 'MONKEY' across the codebase, new team members could quickly locate critical configuration points. This approach allowed us to maintain a balance between flexibility for experimentation and standardization for production, ultimately enabling us to automate the transition from experimental notebooks to production workflows.

4.4 Data Storage

Data storage for reproducible ML experiments demanded substantial disk space and careful consideration of performance characteristics. Deciding the right format for storing static data files can have considerable downstream impact on overall storage

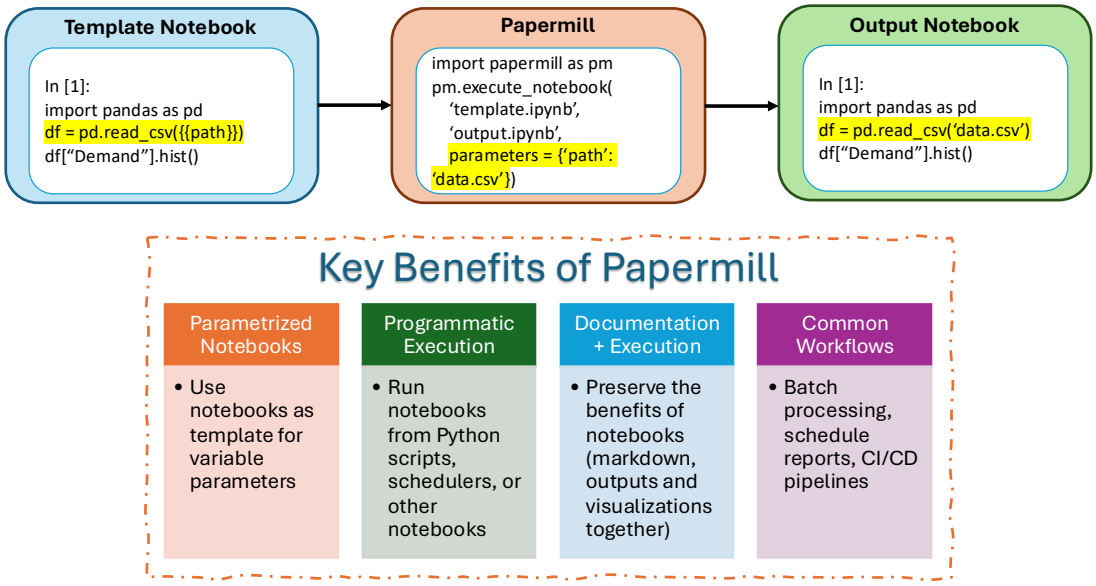


Figure 4.3: Overview of `papermill` library and its key benefits.

costs, processing efficiency, and long-term reproducibility. We evaluated several common file formats with different design philosophies:

1. **Row-based formats:** CSV (plain text, human-readable), CSV_gzip (compressed CSV), JSON (hierarchical text format with schema flexibility)

2. **Binary serialization:** Pickle (Python-specific binary serialization), MessagePack (language-agnostic binary serialization)

3. **Column-oriented formats:** Parquet (compressed columnar format with schema), Feather (fast columnar format optimized for dataframes)

4. **Scientific data formats:** Native HDF5 (hierarchical binary format for scientific data), PyTables (Python implementation built on HDF5)

To systematically evaluate these formats, we simulated a synthetic dataset with 11 columns (5 numeric, 5 categorical, and one timestamp) and one million rows, with each value randomly generated including varying cardinalities for categorical variables. This resulted in a 348 MB dataset in memory. We measured reading time, writing time, and storage size for each format. To ensure reproducibility, we repeated each operation five times and calculated average performance metrics.

Based on this benchmarking, Apache Parquet and Feather formats demonstrates superior performance across all metrics. Parquet achieves the best overall compression (46MB, a 3x reduction compared to CSV) while maintaining fast read/write speeds. Feather is the fastest format for both reading (0.12s) and writing (0.22s), making it 34x faster than CSV for writing operations and 8.9x faster for reading. Traditional formats like CSV perform poorly, with large file sizes (139MB) and slow write speeds (7.6s), though read performance was reasonable. Compressed CSV (CSV_gzip) achieves good compression (58MB) but at the cost of extremely slow write times (21.2s). JSON proves to be the worst performer overall, with the largest file size (249MB) and slowest read times (4.7s). HDF5 and Pickle formats offered balanced

performance, with Pickle excelling at read operations (0.4s) while maintaining moderate file sizes. However, Pickle's key issues emerge from forward and backward compatibility — changing Python or Pandas versions leads to previous versions of Pickle files unusable. See Table 4.2 for detailed results on this quantitative comparison on synthetic data.

While the above results represent a synthetic benchmark, our approach for the actual project evolved through several stages as we encountered limitations with the datasets for our project. We initially used Python's Pickle for data snapshots due to its simplicity and native Python integration. However, this approach revealed significant shortcomings that threatened reproducibility. We experienced version incompatibility when Pickle files created with Python 3.8 failed to load properly in Python 3.9. File size inefficiency became a major problem as our dataset grew to over 10 GB for a single product set in a geography, causing storage and transfer bottlenecks.

After rigorously evaluating alternatives with our proprietary datasets — using methodology similar to the synthetic benchmark presented above — we transitioned to Apache Feather. This format provided several advantages critical for our reproducibility goals. Compression efficiency significantly reduced file sizes by approximately 80% compared to CSV and 30% compared to Pickle, alleviating storage constraints. Version stability ensured forward and backward compatibility across versions, addressing the reproducibility challenges we faced with Pickle. Fast read/write performance—10-30 times faster than CSV for large datasets—dramatically improved our data processing efficiency, especially during iterative model development and evaluation cycles.

4.5 Experimentation with FLAML

Our expansive dataset made it impractical to run full experiments each time a new idea was proposed, yet we needed to maintain reproducibility throughout

Table 4.2: File Format Benchmark Results for 1M Rows and 10 columns Synthetic Dataset with Relative Performance

Format	Raw Performance			Relative to CSV		
	Write (s)	**Read (s)**	**Size (MB)**	**Write (×)**	**Read (×)**	**Size (×)**
Parquet	0.43	0.14	**45.88**	17.54	7.53	**3.03**
Feather	**0.22**	**0.12**	75.78	**34.00**	**8.85**	1.83
Pickle	1.19	0.40	90.21	6.41	2.58	1.54
HDF5	0.93	0.74	98.85	8.16	1.40	1.41
PyTables	2.76	1.62	124.24	2.76	0.64	1.12
CSV	7.61	1.03	138.99	1.00	1.00	1.00
CSV_gzip	21.22	1.75	58.22	0.36	0.59	2.39
MessagePack	6.79	3.20	194.15	1.12	0.32	0.72
JSON	2.33	4.67	249.41	3.26	0.22	0.56

Note: Time is reported in seconds (s) and file sizes in megabytes (MB).

our rapid testing process. To address this challenge, we implemented a multi-stage experimentation process using FLAML (Fast and Lightweight AutoML), an automatic machine learning model in Python developed by Microsoft (Wang and Wu, 2019).[†] FLAML's efficient search and evaluation mechanisms provided rapid feedback on potential approaches while maintaining reproducibility through several key features: cost-aware optimization automatically balanced exploration versus exploitation based on computational budget; multi-fidelity trials started with small samples and progressively increased as promising areas were identified; early stopping terminated underperforming trials to focus resources on promising directions; and transfer learning used knowledge from previous experiments to warm-start new runs.

Our FLAML configuration for rapid hypothesis testing included several key elements. We specified time budgets to control how long model training could run, either in terms of time or iterations. Metric definitions, estimator lists, and ensemble settings were also included. Experiments would compare models like LightGBM, XGBoost, or opt for an ensemble approach. Early stopping flags were set to halt training when accuracy showed no improvement after multiple iterations. Additionally, comprehensive logging ensured thorough tracking of the process. This approach allowed us to test hundreds of configurations at a fraction of the computational cost of exhaustive grid search while maintaining reproducibility. Once promising directions were identified through rapid testing, we proceeded to more comprehensive experimentation, ultimately integrating findings into our primary Iterative Forecasting Algorithm.

4.6 Conclusion

In this chapter, we detailed the implementation of an MLOps framework tailored for large-scale demand forecasting at HP. Our approach emphasized automation, reproducibility, and scalability, ensuring that machine learning workflows remained

[†]Microsoft's FLAML: https://microsoft.github.io/FLAML/

robust across iterative improvements. By integrating DevOps principles into MLOps, we addressed challenges unique to ML systems, such as model decay, data drift, and experiment management.

A key component of our strategy was the adoption of MLflow for experiment tracking, artifact storage, and model versioning. This facilitated seamless experimentation and deployment while maintaining lineage across datasets, feature transformations, and hyperparameter tuning. The incorporation of Jupyter Notebook orchestration, powered by `papermill`, further streamlined structured experimentation, enabling parameterized execution and standardized workflows. Additionally, our keyword-based navigation system provided an efficient way to manage recurring updates, reducing the complexity of routine forecasting tasks.

Our storage strategy evolved through extensive benchmarking, leading to the adoption of Apache Feather for its superior read/write efficiency, compression, and version stability—critical for maintaining reproducibility as datasets scaled. We also demonstrated how FLAML accelerated experimentation, allowing rapid prototyping of models while conserving computational resources.

By leveraging these tools and methodologies, we established a robust, reproducible, and scalable ML pipeline that ensures the integrity and performance of our forecasting models. The next chapter will build on our work in ML-based demand forecasting, and demonstrate its importance in ultimate decision making. While it is easy to judge forecasting algorithms on accuracy, their ultimate test is their impact on the final decision making. We will look into how the ML forecasts affect the decision made by planners, and its effect on inventory through our innovative human-in-the-loop decision making framework.

Chapter 5

Operationalizing ML Forecasting with Human-in-the-loop Framework

Would you accept that intelligence is not the product of thought? If intelligence is the product of thought, then intelligence is mechanical. Thought can never be non-mechanical. — Jiddu Krishnamurti

In the previous chapters, we explored the significance of automating demand forecasting using machine learning algorithms. We also introduced MLOps as the essential framework that enables the practical deployment and management of these models.

In this chapter, we will delve deeper into what it takes to make forecasts truly actionable within a large organization. We will begin by outlining our implementation journey, detailing the key stages of our approach. Next, we will discuss two critical components that bridge the gap between predictions and decision-making: a visual dashboard and a human-in-the-loop strategy. Following that, we will highlight the operational benefits derived from our efforts. Finally, we will conclude with key lessons and best practices for organizations looking to implement similar programs.

Our 'human-in-the-loop' system has been succinctly mentioned in our accepted peer-reviewed work at the *INFORMS Journal of Applied Analytics* (Harshvardhan et al., 2025b) and detailed in *Foresight* (Harshvardhan et al., 2025a). While this chapter adds many details to our prior works, the overall idea remains consistent with those publications.

5.1 Implementation Journey

Before diving into the specifics of our human-in-the-loop implementation, let us walk through its evolution. Our implementation of ML-based forecasting followed a strategic progression that increasingly integrated analytical forecasts into decision-making processes. Each stage built upon the previous one, creating a foundation of trust and demonstrating incremental value before advancing to deeper integration. The steps are summarized in Figure 5.1.

Business KPI Dashboard We began by establishing a single integrated KPI dashboard for the entire Print Business. This dashboard aligned executive KPIs with operational metrics, instituted a monthly review process, and introduced Forecast Value Add (FVA) as a key metric to inform decisions. By creating visibility into forecasting performance across all business units, we established a common language for discussing forecast accuracy and a baseline against which improvements could be measured. This foundation was crucial for gaining executive sponsorship and establishing the organizational discipline necessary for subsequent stages.

ML Forecast Pilot Starting in 2019, the second stage introduced ML visibility in the dashboard, serving as directional guidance primarily for forecast bias reduction. Without forcing adoption, we demonstrated how ML forecasts could complement existing processes by addressing specific weaknesses in the statistical approaches. This pilot phase allowed planners to observe ML performance over multiple cycles

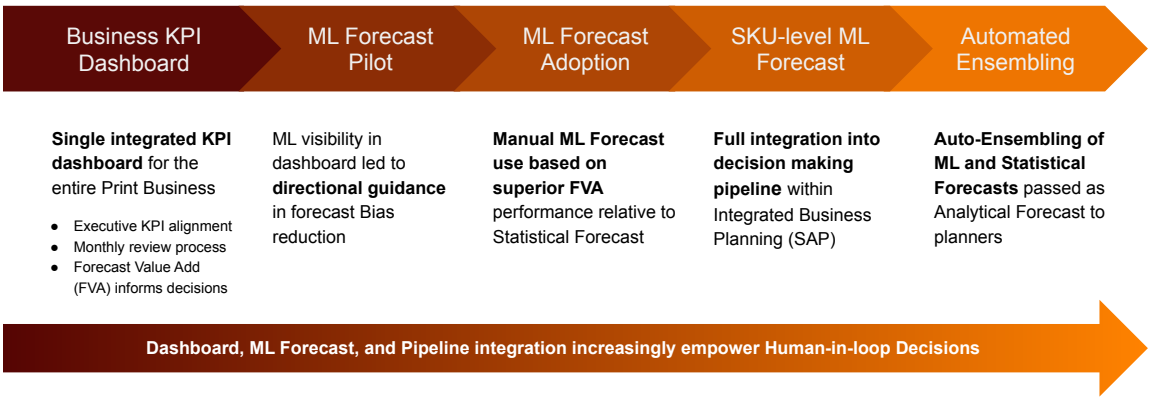


Figure 5.1: Implementation journey of our ML forecasting project at HP Inc.

without disrupting established workflows, gradually building confidence in the new methodology. The tangible improvements in bias reduction provided evidence of ML's potential value while minimizing organizational resistance.

ML Forecast Adoption As confidence grew, we moved to manual ML forecast use based on superior FVA performance relative to Statistical Forecast. Planners began selectively incorporating ML forecasts when their performance consistently outperformed other methods. This selective adoption approach respected planner expertise while encouraging data-driven decision-making. The principle of "analytics as guidance" rather than replacement proved critical for gaining planner buy-in, as they maintained control over final forecast decisions while benefiting from ML insights. We reiterated that the ML (and analytical) forecasts are only to augment the work being done by the planners and not to replace their jobs.

SKU-level ML Forecast With established credibility, we implemented full integration into the decision-making pipeline within Integrated Business Planning (SAP). ML forecasts became directly accessible within the tools planners used daily, eliminating friction in accessing and applying analytical insights. This integration represented a shift from ML as an optional reference to an embedded component of the planning workflow. The convenience of having forecasts available at the point of decision-making significantly increased adoption rates and impact on inventory management.

Automated Ensembling The final stage introduced auto-ensembling of ML and Statistical Forecasts passed as Analytical Forecast to planners. This sophisticated approach automatically combined the strengths of different forecasting methods based on historical performance patterns. By presenting a unified analytical forecast that leveraged the best available methods for each context, we simplified the planner experience while maximizing forecast accuracy. This automation of technical

decisions allowed planners to focus their expertise on incorporating market intelligence and contextual factors that models could not capture.

Throughout this journey, the progressive integration of dashboard analytics, ML forecasting, and business planning systems increasingly empowered human-in-loop decisions, resulting in the significant operational improvements documented in subsequent sections. The deliberate sequencing of these stages was instrumental in overcoming organizational resistance while building capabilities that delivered substantial business value.

5.2 Dashboard

The forecasting analytics dashboard served as a pivotal instrument in communicating model performance to planners and decision-makers, significantly enhancing transparency and facilitating informed decision-making. Prior to the implementation of machine learning (ML) forecasts, the development of this comprehensive dashboard laid the groundwork necessary for gaining crucial stakeholder acceptance.

The dashboard provided extensive insights into historical performance, showcasing both traditional statistical and ML forecasts alongside the Consensus forecasts. Performance metrics such as Bias, weighted Mean Absolute Percentage Error (wMAPE), and Root Mean Square Error (RMSE) were systematically displayed to allow direct comparison across models. This transparency empowered planners to meticulously evaluate and identify the most appropriate forecasting method for each geographic region and individual stock-keeping unit (SKU).

A critical feature of the dashboard was its *interactive heatmap*, which clearly illustrated the best-performing model—determined by the lowest wMAPE—across different time periods and product categories within HP's print product line. This visualization enabled planners to quickly discern patterns and make precise selections tailored to their forecasting needs.

Additionally, the dashboard quantified Forecast Value Add (FVA), highlighting the incremental accuracy gains achieved through model selection compared to baseline forecasts. This metric underscored the practical benefits of leveraging advanced analytics in forecasting decisions.

By encompassing all products, geographical locations, and forecasting methodologies in a single, user-friendly interface, the dashboard's universality was instrumental in securing buy-in from planners and stakeholders. Its ability to present historical accuracies, comparative model performance, and visual analytics created a strong foundation of trust, ultimately driving widespread adoption and integration of advanced forecasting methods within the planning processes.

5.3 Human-in-the-loop Ensembling

Our forecasting system leverages a sophisticated human-in-the-loop framework that seamlessly integrates ML and statistical models with human expertise to generate accurate and reliable forecasts. Initially, ML and statistical forecasts are combined through analytical ensembling, guided by performance-based heuristics, producing an *Analytical forecast*. This analytical forecast serves as a critical decision support tool for human planners.

Depending on specific circumstances, planners may directly utilize the analytical forecast or adjust their Consensus forecasts based on it. When additional human judgment is crucial—such as during ongoing price promotions or when humans possess contextual knowledge beyond the scope of the models—planners apply their expert judgments and rely on Consensus forecasts. Conversely, in scenarios where products or SKUs exhibit consistent and clear demand trends, the analytical forecast or even the ML forecast alone may be directly adopted as the final forecast. This decision making flow in illustrated in Figure 5.2.

This dynamic interplay ensures that human planners strategically focus their efforts on high-value analytical tasks, optimizing their use of expertise where

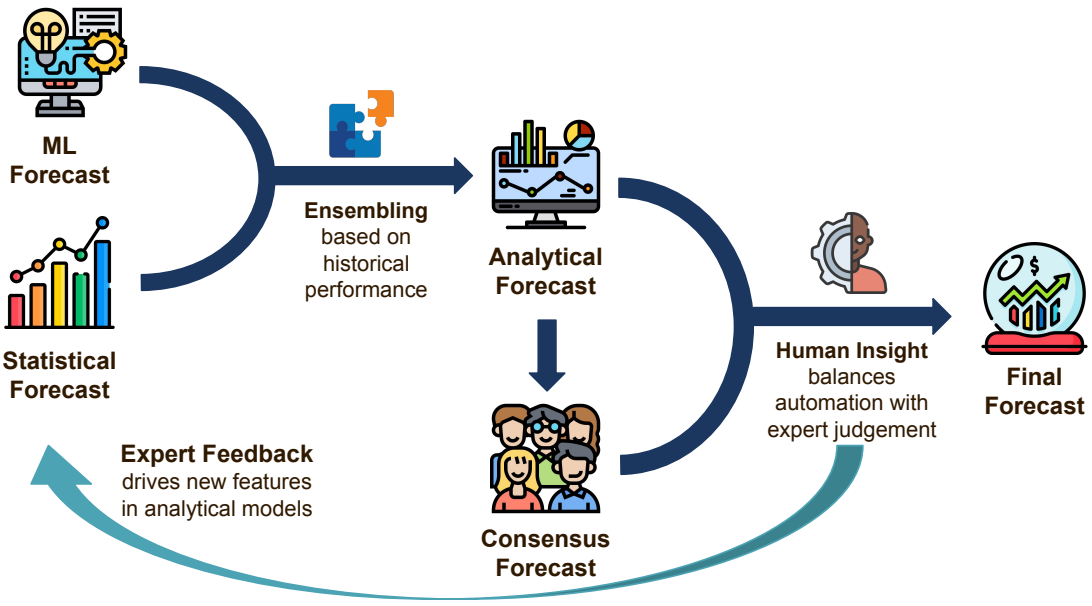


Figure 5.2: Illustration of Human-in-the-Loop Forecasting that combines Machine Precision with Human Insight.

it most significantly improves forecast accuracy. Furthermore, expert feedback continually informs the iterative improvement of both the ML and statistical models by highlighting areas where additional model features could be beneficial. This *closed-loop feedback* mechanism not only refines forecasting accuracy but also enhances model explainability and causal understanding.

Ultimately, this collaborative process achieves an optimal balance, capitalizing on the precision and efficiency of automated forecasts while leveraging human insight to account for market nuances and unforeseen anomalies. As the system evolves, future enhancements, such as AI-driven drift and anomaly detection integrated with advanced analytical dashboards, will further empower human planners and improve decision-making efficacy.

5.4 Operational Benefits

5.4.1 Inventory Reduction

The implementation of our human-in-the-loop forecasting architecture delivered substantial and measurable business value across multiple dimensions. By successfully balancing human judgment with machine learning capabilities, we achieved significant operational improvements over a three-year period.

The most notable impact was a dramatic 28.5% reduction in in-hand inventory levels over three years, without compromising customer service levels. This sustained inventory decline followed a clear downward trend ($R^2 = 0.781$), demonstrating the consistent effectiveness of our integrated approach. Figure 5.3 illustrates this reduction trajectory from 2022 to 2025 for a select group of HP Print product portfolio, encompassing 1,484 distinct products on an anonymized scale.

This inventory optimization directly translates into substantial working capital improvements. By maintaining lower inventory levels while still meeting customer demand, HP effectively freed up significant capital that was previously tied up in

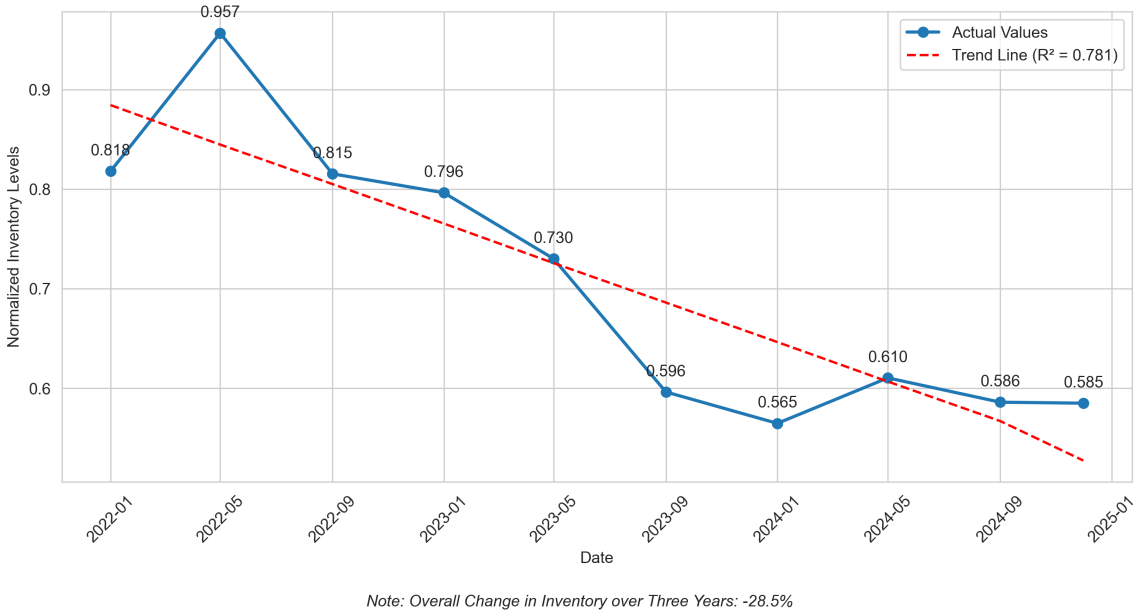


Figure 5.3: Inventory trajectory from 2022 to 2025 showing reduction by 28.5%.

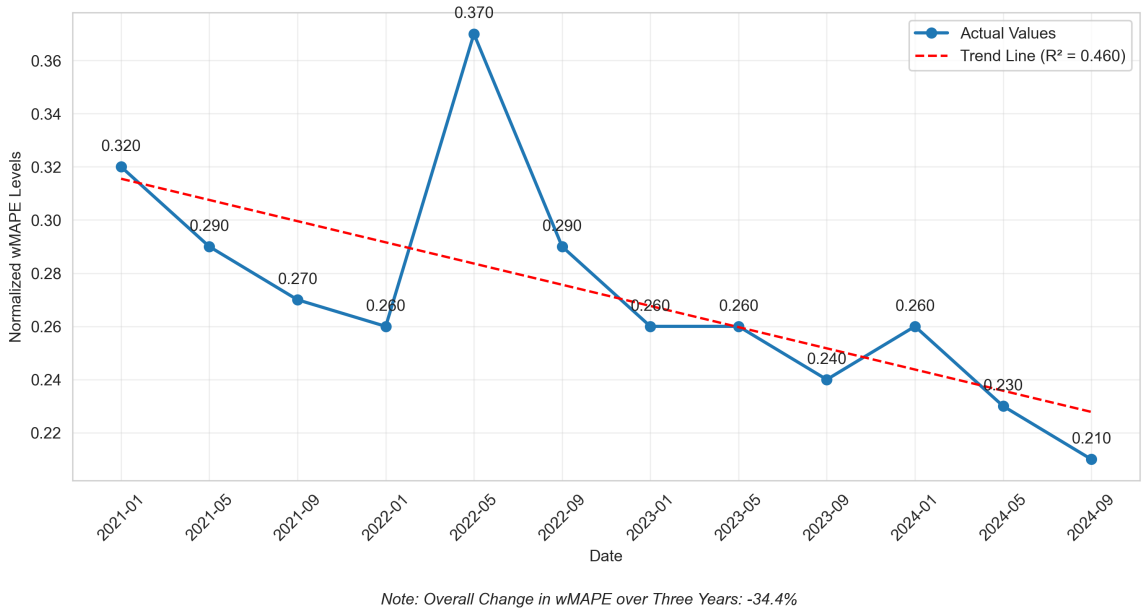
excess stock. The normalized inventory values shown in the figure (relative to peak levels) reveal how the implementation steadily improved inventory efficiency quarter over quarter, with minor seasonal fluctuations that did not disrupt the overall downward trend.

5.4.2 Forecast Accuracy Improvements

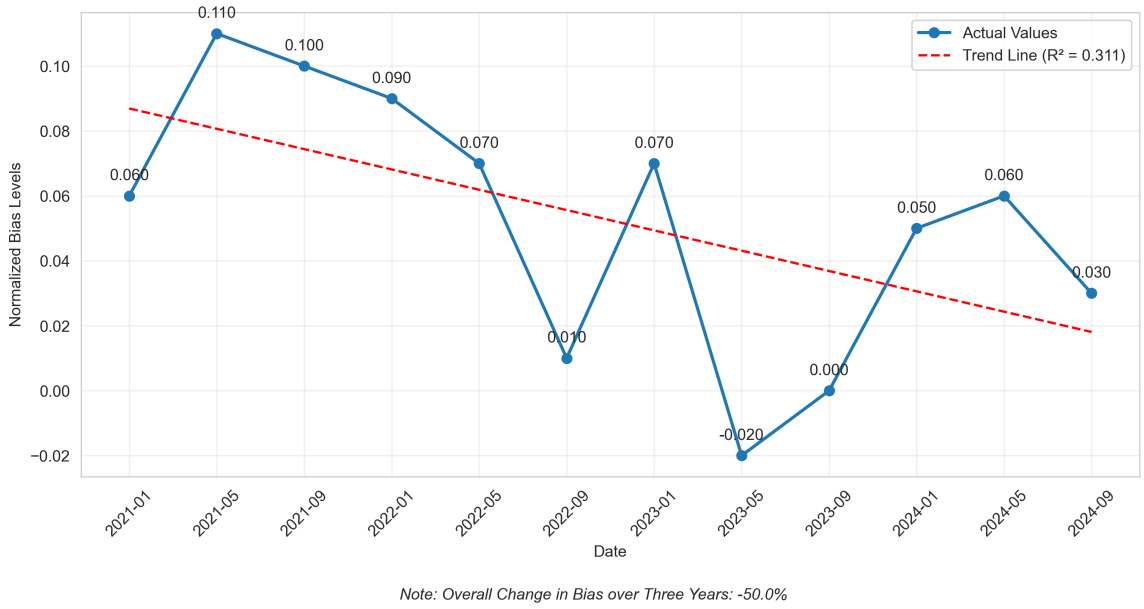
Our key performance indicators for demand forecasting demonstrated substantial improvement across the board; see Figure 5.4. The weighted Mean Absolute Percentage Error (wMAPE) decreased by 34.4% over three years for the same selected group of 1,484 distinct products. This reduction signifies a dramatic improvement in the precision of our demand predictions, enabling more effective inventory management and production planning.

Simultaneously, we observed an even more pronounced reduction in Bias, with a 50% decrease over the same period for the same group of 1,484 products. This remarkable improvement in Bias represents a fundamental shift from systematically over- or under-forecasting toward more balanced and realistic demand projections. The reduction in forecasting bias directly contributed to the inventory optimization discussed earlier, as planners could make decisions based on more accurate expected demand patterns rather than compensating for known systematic errors.

These accuracy improvements were not merely statistical achievements but translated directly into operational benefits. The enhanced forecast quality enabled more precise procurement, production scheduling, and distribution planning. With more reliable demand signals, the organization could confidently maintain lower inventory levels while still meeting customer service requirements.



(a) wMAPE KPI from 2022 to 2025 showing reduction by 34.4%.



(b) Bias KPI from 2022 to 2025 showing reduction by 50%.

Figure 5.4: Performance metrics illustrating forecasting accuracy improvements: (a) wMAPE and (b) Bias from 2022 to 2025.

5.5 Principles and Lessons: Making Analytical Forecasts Actionable

Despite the importance of demand forecasting, transforming forecasts into actionable insights faces technical and organizational hurdles. Research shows many respondents "are satisfied with the way we now make projections," indicating resistance to change (Goodwin et al., 2023, p. 5). Human planners often distrust 'black box' forecasts they cannot easily understand. Sometimes, organizational politics also create incentives to manipulate forecasts—either inflating them ("enforcing") to please investors or account for upcoming promotions, or deflating them ("sandbagging") to exceed targets or align with product phase-outs.

Based on our implementation experience at HP, we propose a three-pronged approach to implementation of ML-based demand forecasting approach at a large scale: having interactive dashboard that gives planners visibility and control over entire forecasting pipeline, authority to use analytical forecasts as a starting point rather than forcing it, and ensuing analytical forecasts are available when and where they need it.

Visual Dashboards Modern systems require interactive dashboards where planners can visualize, adjust, and channel forecasts into downstream decisions across production, R&D, pricing, and operations. Well-designed dashboards display statistical forecasts alongside planner overrides and actual outcomes, enabling continuous learning between humans and systems. Dashboards enhance transparency, building trust as planners learn when models excel and when human intuition adds value.

Analytics as Guidance Goodwin et al. (2023, p. 9) note that clear baseline forecasts provide stakeholders a common starting point and make adjustments explicit. Organizations embracing analytical forecasting report greater objectivity as baselines "allowed discussion of how the future will be different or why judgment

calls differ from the baseline." That is, humans adopt analytical forecasts only when they maintain final control and trust they can override predictions (Dietvorst et al., 2018).

Integration into Planning Tools Only when the forecasting system becomes an integral component of Integrated Business Planning rather than existing as an isolated tool, the forecasters feel obliged to use the analytical forecasts directly. Unless the analytical forecasts are available when they need it, it can only act as "directional guidance" rather than have complete adoption.

5.6 Conclusion

The availability of high-quality analytical forecasts as baseline guidance substantially reduced the time required for forecast preparation. The semi-automated approach optimized workforce allocation, allowing skilled planners to concentrate on strategic decision-making rather than tactical forecasting. This resulted in more efficient use of human resources across the planning organization. The operational success of this implementation demonstrates that the challenges of making analytical forecasts actionable can be effectively overcome through thoughtful system design that enhances transparency, preserves human agency, and integrates seamlessly into existing planning workflows.

To conclude, three key elements make forecasts truly actionable: interactive dashboards showing both analytical and judgmental forecast performance; algorithms that augment rather than replace human work; and forecasts that are directly accessible without additional effort. In essence, analytical forecasting should represent a "nudge" rather than "sludge" (Luo et al., 2023). The resulting 28.5% inventory reduction stands as compelling evidence that ML-based forecasting, when properly implemented with humans in the loop, can deliver transformative business value at enterprise scale.

Chapter 6

Bridging Demand Forecasting and Decision Optimization

अब उसे देख ललचाना क्या?
पीछे को पाँव हटाना क्या?
जय को कर लक्ष्य चलेंगे हम,
अरि-दल का गर्व दलेंगे हम।
— रामधारी सिंह 'दिनकर', रश्मि-रथी (पंचम सर्ग)*

The final use case for most forecasting methods is decision-making, which typically combines both machine learning and optimization. The machine learning model estimates unknown parameters for the optimization problem, while the optimization component guides the actual decision-making process. However, since both prediction and optimization are inherently complex, practitioners often default to a *Predict-Then-Optimize* paradigm that treats these as separate sequential steps. In contrast, a *Predictive Optimization* framework integrates the decision-making optimization problem directly into the machine learning or deep learning prediction task. This approach makes the constraints and objectives of decision-making readily available to the forecasting model, improving outcomes without increasing computation time.

*Translation: Why now crave for what is lost? Why retreat from the battlefield? We march forward with victory as the sole aim, vanquishing the pride of enemy host. — Ramdhari Singh 'Dinkar', *Rashmi-Rathi (Fifth Canto)*

In this chapter, we first use a simulation study to demonstrate that good prediction does not always lead to good decisions. While there appears to be a general trend where lower prediction error corresponds to lower decision error, this relationship is violated in a sufficient number of cases to warrant reexamination of the underlying approach. Based on these insights and motivated by Mao et al. (2023), we propose extending the concept of end-to-end predictive optimization to supply chain management. Our approach applies the principles of integrating forecasting and decision optimization demonstrated by Mao et al. (2023) in advertising, adapting them to the unique challenges of demand forecasting in an enterprise supply chain context.

6.1 Good Demand Forecasts $\neq$ Good Production Planning Decision

To empirically demonstrate the importance of forecast accuracy in production planning, we conducted an extensive simulation study based on proprietary data from HP examining how forecast errors affect decision quality and overall costs. Our simulation focuses on a multi-product manufacturing environment where production decisions must be made under demand uncertainty. The study specifically investigates how different types of forecast errors—purely random versus correlated with actual demand—influence the optimality gap in production decisions. This analysis provides insights into the value of predictive accuracy and supports our argument for an integrated end-to-end predictive optimization approach.

We formulate the production planning problem as follows:

$$\min_{x_t, I_t, e_t} \quad \sum_{t=1}^{T} \sum_{p=1}^{P} (V_p x_{t,p} + Z_p I_{t,p}^{+} + OC_p e_{t,p}) \tag{6.1}$$

$$\text{s.t.} \quad I_{t,p} = I_{t-1,p} + x_{t,p} - D_{t,p} \quad \forall t \in \{1, \ldots, T\}, p \in \{1, \ldots, P\} \tag{6.2}$$

$$I_{t,p} \geq -e_{t,p} \quad \forall t, p \tag{6.3}$$

$$I_{t,p}^{+} \geq I_{t,p} \quad \forall t, p \tag{6.4}$$

$$I_{t,p}^{+} \geq 0 \quad \forall t, p \tag{6.5}$$

$$x_{t,p} \leq \text{Capacity}_p \quad \forall t, p \tag{6.6}$$

$$x_{t,p}, e_{t,p} \geq 0 \quad \forall t, p \tag{6.7}$$

Where:

- $x_{t,p}$ represents the production quantity for product p in period t
- $I_{t,p}$ is the inventory level (positive or negative) for product p at the end of period t
- $I_{t,p}^{+}$ captures positive inventory for product p at the end of period t
- $e_{t,p}$ represents the shortage (unmet demand) for product p in period t
- V_p is the unit production cost for product p
- Z_p is the unit inventory holding cost for product p
- OC_p is the opportunity cost (penalty) for each unit of unmet demand for product p
- $D_{t,p}$ is the demand for product p in period t

Our simulation approach compares two scenarios:

1. An ideal scenario where perfect demand information is available

2. A realistic scenario where production decisions are based on forecasts with errors

The key metric of interest is the *cost difference* between these scenarios, which represents the economic impact of forecast errors on decision quality. For our simulations, we utilized a subset of HP product demand data, applying different error structures to generate realistic forecasts.

6.1.1 Uncorrelated Random Errors: Theoretical Best Case

In our first simulation scenario, we examine the impact of purely random (uncorrelated) forecast errors on production planning decisions. This represents a theoretical best-case scenario in forecasting, where errors are not systematically related to the actual demand values. For each simulation run, forecasts were generated by adding normally distributed random noise to the actual demand:

$$\hat{Y} = Y + \varepsilon, \quad \text{where} \quad \varepsilon \sim \mathcal{N}(0, \sigma_Y) \tag{6.8}$$

With correlation coefficient $\rho = 0$, errors are purely stochastic and independent of the demand magnitude. We conducted 50 simulation runs with different random error seeds to capture the range of possible outcomes.

Figure 6.1 displays the relationship between forecast error (measured by RMSE) and the resulting cost difference between ideal and forecast-based production plans. Our simulation reveals a positive correlation between forecast error magnitude and cost penalty. The regression analysis shows a positive relationship ($r = 0.28$, $p = 0.05$) between forecast error and cost difference. The R^2 value of 0.08 indicates that approximately 8% of the variance in cost difference can be explained by the magnitude of forecast errors, which isn't much. The average cost difference across simulations was substantial, highlighting that even random forecast errors translate into significant economic losses. The relatively high variance in the cost difference at similar RMSE levels indicates that the specific pattern of errors, not just their magnitude, affects decision quality.

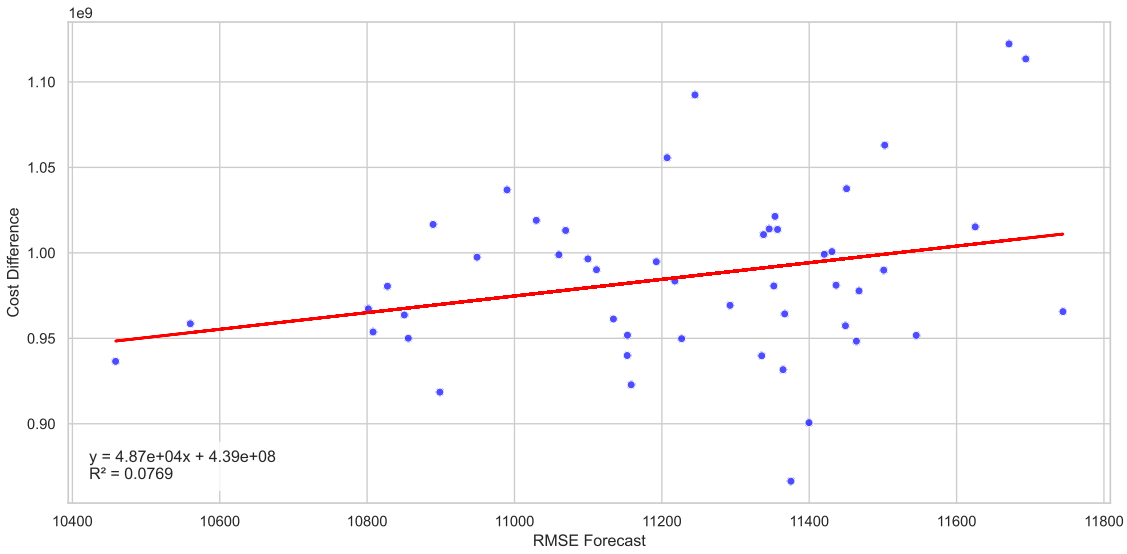


Figure 6.1: Decision errors measured as objective cost difference against forecast errors measured with RMSE for purely uncorrelated forecasts with random errors.

The regression coefficient suggests that for each unit increase in RMSE, the cost difference increases by approximately 4.87×10^4 monetary units. This confirms that larger forecast errors consistently lead to greater cost penalties, reinforcing the intuitive understanding that better forecasts lead to better decisions in a quantifiable manner when errors are random.

6.1.2 Correlated Forecast Errors: The Realistic Scenario

In practice, forecast errors are rarely purely random but often exhibit correlation with the actual values. Our second simulation scenario introduces correlation between forecast errors and actual demand values, with $\rho = 0.7$, providing a more realistic representation of forecasting challenges. The correlated forecasts were generated using:[†]

$$\hat{Y} = Y + \rho Y + \sqrt{1 - \rho^2} \cdot \varepsilon, \quad \text{where} \quad \varepsilon \sim \mathcal{N}(0, \sigma_Y) \tag{6.9}$$

This formulation creates a systematic relationship between the magnitude of demand and the size of forecast errors. Similar to the uncorrelated case, we conducted 50 simulation runs with different random seeds.

Figure 6.2 illustrates the relationship between RMSE and cost difference for the correlated error scenario. The simulation demonstrates that correlated errors result in a different pattern of economic impact compared to random errors. We can observe that for similar RMSE values, the cost differences are generally lower than in the uncorrelated case. This suggests that the optimization model can better accommodate systematic errors than purely random ones. Our regression analysis for correlated errors shows a stronger relationship ($r = 0.40$, $p < 0.01$) between forecast error and cost difference compared to the uncorrelated case. The R^2 value of 0.16 indicates that

[†]We direct the interested readers to https://stats.stackexchange.com/a/71303/185332 for an intuitive explanation of this formula.

approximately 16% of the variance in cost difference can be explained by forecast error magnitude—about twice as high as in the uncorrelated scenario.

The regression coefficient reveals that for each unit increase in RMSE, the cost difference increases by approximately 4.66×10^4 monetary units. Notably, this coefficient is slightly lower (about 4% less) than in the uncorrelated case, suggesting that while there is a stronger relationship between error magnitude and cost penalty, the economic impact per unit of RMSE is similar when errors are correlated with demand values. This difference in economic impact suggests that the structure of errors—not just their magnitude—affects how forecast inaccuracies translate to economic penalties. In practical terms, this means that standard forecast error metrics like RMSE may not proportionally reflect the economic impact of errors when correlation exists.

6.1.3 Implications for Predictive Optimization

Our simulation findings offer several key insights regarding the relationship between forecast quality and decision optimality:

First, forecast errors invariably lead to suboptimal decisions and increased costs, confirming the critical role of accurate forecasting in the two-stage approach. The average cost difference was 9.86×10^8 for uncorrelated errors and 1.32×10^9 for correlated errors, representing significant economic penalties from imperfect forecasting. Second, the nature of forecast errors—not just their magnitude—significantly affects decision quality, with correlated errors showing different patterns of economic impact than uncorrelated ones. While the slopes of the regression lines are similar (4.87×10^4 vs 4.66×10^4), the correlated case shows a stronger statistical relationship ($R^2 = 0.16$ vs $R^2 = 0.08$). Third, standard forecast error metrics like RMSE do not fully capture the decision-relevant quality of predictions, as evidenced by the relatively low R^2 values in both scenarios.

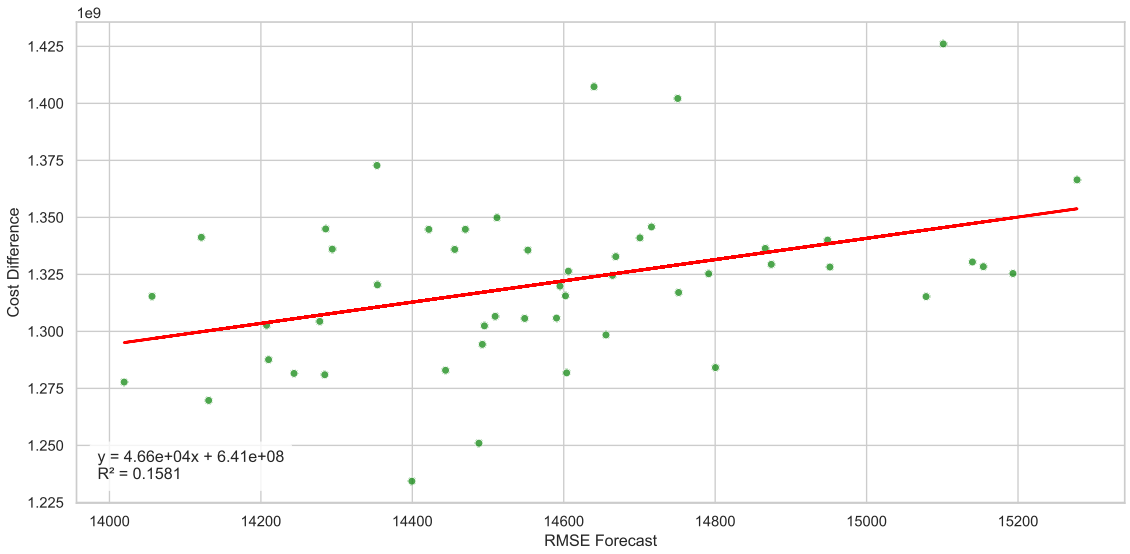


Figure 6.2: Decision errors measured as objective cost difference against forecast errors measured with RMSE when forecasts and true values have correlation $\rho = 0.7$.

These findings strongly support the case for an end-to-end predictive optimization approach. By merging the forecasting and optimization stages, we can directly optimize for decision quality rather than forecast accuracy. This integrated approach accounts for how errors propagate through the decision process and naturally accommodates the complex relationship between prediction errors and economic outcomes.

The simulation results also suggest that practitioners should evaluate forecasting methods not merely on statistical accuracy but on their ultimate impact on decision quality and economic outcomes. This decision-focused perspective aligns with the growing literature on end-to-end learning for optimization problems and offers a promising direction for improving supply chain and production planning systems.

6.2 Predictive Optimization for Supply Chain Management

Building on our simulation insights, we now propose a framework that directly connects demand prediction to production decisions. Our approach extends the end-to-end predictive optimization principles demonstrated by Mao et al. (2023) in advertising to the unique challenges of production planning under demand uncertainty. The proposed framework integrates the decision-making optimization problem directly into the machine learning prediction task, enabling the forecasting model to learn patterns that minimize economic impact rather than just statistical error.

6.2.1 Integrated Demand Forecasting and Production Optimization

Traditional supply chain decision-making typically follows a two-stage approach: first forecasting demand, then optimizing production and inventory decisions based on

those forecasts. Our proposed framework integrates these components into a single end-to-end system that directly optimizes for decision quality rather than forecast accuracy. The framework consists of three essential components:

1. **Demand Forecasting Model**: A deep learning model that predicts future demand based on historical patterns and contextual information
2. **Production Planning Optimization**: A formulation that determines optimal production quantities, inventory levels, and shortages
3. **Differentiable Lagrangian Layer**: A novel component that allows optimization decisions to inform the training of the prediction model

In the demand forecasting stage, we train a model to predict demand $\hat{y} = f(z; \omega)$, where z represents input features and ω represents model parameters. However, unlike traditional approaches that minimize prediction error, our framework minimizes the impact of prediction errors on production decisions.

6.2.2 Production Planning Formulation

We now present a vectorized version of the production planning formulation from Section 6.1. Let $\boldsymbol{y} \in \mathbb{R}^T$ be the vector of actual demand for T time periods, $\hat{\boldsymbol{y}} \in \mathbb{R}^T$ be the vector of predicted demand, $\boldsymbol{x} \in \mathbb{R}^T$ be the vector of production decisions, and $\boldsymbol{E} \in \mathbb{R}^T$ be vector of unmet demand for T time periods.

Inventory $\boldsymbol{I}_t$ at any time period t is calculated as the sum of last period's inventory $\boldsymbol{I}_{t-1}$, production $\boldsymbol{x}_t$, minus demand $\boldsymbol{y}_t$, plus unmet demand $\boldsymbol{E}_t$:

$$\boldsymbol{I}_t = \boldsymbol{I}_{t-1} + \boldsymbol{x}_t - \boldsymbol{y}_t + \boldsymbol{E}_t \tag{6.10}$$

In vectorized notation, this can be written as:

$$\boldsymbol{I} = \boldsymbol{I}^{-1} + \boldsymbol{x} - \boldsymbol{y} + \boldsymbol{E} \tag{6.11}$$

where $\boldsymbol{I} \in \mathbb{R}^T$ is the inventory vector and $\boldsymbol{I}^{-1} \in \mathbb{R}^T$ is the shifted inventory vector with first element zero. Through algebraic manipulation, we can express the inventory as:

$$\boldsymbol{I} = L(\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{E}) \tag{6.12}$$

where L is a lower triangular matrix of ones that captures the cumulative nature of inventory over time:

$$L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{6.13}$$

The objective function minimizes the total cost, which includes production costs, inventory holding costs, and opportunity costs for unmet demand:

$$\min_{\boldsymbol{x},\boldsymbol{E}} \boldsymbol{v}'\boldsymbol{x} + \boldsymbol{z}'L(\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{E}) + \boldsymbol{q}'\boldsymbol{E} = (\boldsymbol{v}' + \boldsymbol{z}'L)\boldsymbol{x} + (\boldsymbol{q}' + \boldsymbol{z}'L)\boldsymbol{E} - \boldsymbol{z}'L\boldsymbol{y} \tag{6.14}$$

where $\boldsymbol{v}$ is the variable cost of production per unit, $\boldsymbol{z}$ is storage cost per unit, and $\boldsymbol{q}$ is the opportunity cost of unmet demand.

Subject to the following constraints:

$$\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{C} \quad \text{(Production capacity constraints)} \tag{6.15}$$

$$\boldsymbol{E} \geq \boldsymbol{0} \quad \text{(Non-negative unmet demand)} \tag{6.16}$$

$$L(\boldsymbol{x} - \boldsymbol{y} + \boldsymbol{E}) \geq \boldsymbol{0} \quad \text{(Non-negative inventory)} \tag{6.17}$$

To formulate this in standard form, we define: $\boldsymbol{m} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{E} \end{bmatrix}$, $\boldsymbol{a'} = \Big[\boldsymbol{v'} + \boldsymbol{z'}L, \boldsymbol{q'} + \boldsymbol{z'}L\Big]$,

$B = \begin{bmatrix} \mathrm{I} & 0 \\ -\mathrm{I} & 0 \\ 0 & -\mathrm{I} \\ -L & -L \end{bmatrix}$, and $\boldsymbol{p} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ -L\boldsymbol{y} \end{bmatrix}$.

This gives us the standard form:

$$\min_{\boldsymbol{m}} \boldsymbol{a'}\boldsymbol{m} - \boldsymbol{z'}L\boldsymbol{y}, \tag{6.18}$$

such that $B\boldsymbol{m} \leq \boldsymbol{p}$. Note that $\boldsymbol{z'}L\boldsymbol{y}$ is constant and can be ignored during optimization.

6.2.3 Extension to Multiple Products

For multiple products (denoted by P), we extend the formulation to account for production and inventory decisions across the product portfolio. We define combined variable matrix:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{x}^1 & \boldsymbol{x}^2 & \boldsymbol{x}^3 & \dots & \boldsymbol{x}^P \\ \boldsymbol{E}^1 & \boldsymbol{E}^2 & \boldsymbol{E}^3 & \dots & \boldsymbol{E}^P \end{bmatrix} \in \mathbb{R}^{2T \times P}, \tag{6.19}$$

with corresponding parameter matrices for variable production costs $\boldsymbol{V}$, storage costs $\boldsymbol{Z}$, opportunity costs $\boldsymbol{Q}$, demand vectors $\boldsymbol{Y}$, and capacity constraints $\boldsymbol{C}$.

The objective function becomes:

$$\min_{\boldsymbol{M}} \sum_{i=1}^{P} \left[\boldsymbol{v}^{i\top}\boldsymbol{x}^i + \boldsymbol{z}^{i\top}L(\boldsymbol{x}^i - \boldsymbol{y}^i + \boldsymbol{E}^i) + \boldsymbol{q}^{i\top}\boldsymbol{E}^i\right] \tag{6.20}$$

$$= \min_{\boldsymbol{M}} \sum_{i=1}^{P} \left[(\boldsymbol{v}^i + L^\top\boldsymbol{z}^i)^\top\boldsymbol{x}^i + (\boldsymbol{q}^i + L^\top\boldsymbol{z}^i)^\top\boldsymbol{E}^i - \boldsymbol{z}^{i\top}L\boldsymbol{y}^i\right], \tag{6.21}$$

which can be expressed in matrix form using the Frobenius inner product:

$$\min_{\boldsymbol{M}} \langle \boldsymbol{A}, \boldsymbol{M} \rangle_{\mathcal{F}} - \sum_{i=1}^{P} \boldsymbol{z}^{i\top} L \boldsymbol{y}^{i} = \min_{\boldsymbol{M}} \operatorname{trace}(\boldsymbol{A}^{\top} \boldsymbol{M}) - \sum_{i=1}^{P} \boldsymbol{z}^{i\top} L \boldsymbol{y}^{i}. \tag{6.22}$$

6.2.4 Differentiable Lagrangian Optimization

The key innovation in our approach is the introduction of a differentiable Lagrangian layer that enables end-to-end training. This layer reformulates the production planning problem as a differentiable optimization problem whose gradients can be backpropagated through the neural network during training. The Lagrangian layer acts as a forward pass in the neural network architecture, connecting the demand prediction outputs to the optimization outcomes.

Using the Lagrangian formulation:

$$\mathcal{L}(\boldsymbol{m}, \lambda) = \boldsymbol{a}'\boldsymbol{m} - \boldsymbol{z}' L \boldsymbol{y} + \lambda^{T}(B\boldsymbol{m} - \boldsymbol{p}), \tag{6.23}$$

where λ represents Lagrangian multipliers corresponding to constraints. The Karush-Kuhn-Tucker (KKT) conditions characterize the optimal solution:

$$\nabla_{\boldsymbol{m}} \mathcal{L}(\boldsymbol{m}, \lambda) = \boldsymbol{a} + B^{T}\lambda = 0 \tag{6.24}$$

$$\lambda \odot (B\boldsymbol{m} - \boldsymbol{p}) = 0 \tag{6.25}$$

$$\lambda \geq 0 \tag{6.26}$$

$$B\boldsymbol{m} - \boldsymbol{p} \leq 0, \tag{6.27}$$

where $\odot$ denotes element-wise multiplication.

These KKT conditions enable differentiability by providing an analytical characterization of how the optimal decision changes with respect to the inputs (including demand predictions). During the forward pass, the Lagrangian layer solves the

optimization problem using the predicted demand as input. This allows the gradient information to flow from the optimization outcome back to the prediction model parameters, enabling the prediction model to learn patterns that lead to better decisions rather than just better statistical accuracy.

6.2.5 Optimization Regret and End-to-End Training

By computing the gradient of the Lagrangian with respect to model parameters, we can train the forecasting model to minimize the optimization regret—the difference between decisions made with predicted demand versus perfect information:

$$L_{opt} = \frac{1}{T}[\boldsymbol{a}_2'\boldsymbol{E} + (\boldsymbol{v}' + \boldsymbol{z}'L)'\hat{\boldsymbol{x}} - \boldsymbol{z}'L\boldsymbol{y}] - [\boldsymbol{a}'\boldsymbol{m} - \boldsymbol{z}'L\boldsymbol{y}]. \tag{6.28}$$

This regret must be positive as knowing true demand $\boldsymbol{y}$ should give the lowest cost of operations. The total loss function combines both prediction accuracy and optimization regret:

$$L_{total} = \alpha L_{pred} + (1 - \alpha)L_{opt}, \tag{6.29}$$

where $\alpha \in [0, 1]$ is a hyperparameter that balances the importance of prediction accuracy versus decision quality. This end-to-end approach allows the forecasting model to learn patterns that minimize the economic impact of prediction errors rather than just their statistical magnitude.

6.3 Conclusion and Future Directions

This integrated predictive optimization framework offers several potential advantages for supply chain management. First, it naturally handles asymmetric costs where underforecasting has different implications than overforecasting. Second, the forecasting model becomes constraint-aware, generating predictions that respect operational limitations. Third, the system directly optimizes for relevant business metrics rather

than statistical accuracy measures. Fourth, as demonstrated in our simulation study, decision quality can be substantially improved with an end-to-end approach.

We believe this predictive optimization framework is particularly promising for settings characterized by high demand uncertainty, complex constraints, and asymmetric costs of errors, such as perishable goods, fashion retail, or spare parts inventory management. Future work will focus on empirical validation using real-world supply chain data, with particular attention to scenarios where traditional methods struggle, such as new product introductions, seasonal transitions, and high-volatility product categories.

Chapter 7

Concluding Remarks

I'm a scientist; because I invent, transform, create, and destroy for a living, and when I don't like something about the world, I change it. — Rick Sanchez from Rick and Morty

This dissertation has presented a comprehensive framework for implementing machine learning-based demand forecasting at enterprise scale, demonstrating both the technical and organizational dimensions of this complex challenge. We began by exploring the evolution of demand forecasting methodologies, contextualizing our work within both historical approaches and contemporary advancements in machine learning. Our implementation of LightGBM-based forecasting models demonstrated significant improvements over traditional statistical approaches, with particular strength in reducing forecast bias and weighted Mean Absolute Percentage Error (wMAPE). The iterative forecasting algorithm developed in this work effectively captured complex demand patterns across diverse product categories and geographical regions, while maintaining computational efficiency necessary for enterprise-scale deployment.

The MLOps infrastructure established for this project represented a significant advancement in reproducible machine learning experimentation, combining MLflow for experiment tracking, parameterized Jupyter notebooks for workflow orchestration,

and optimized data storage formats for scalability. Through careful feature engineering and selection, we incorporated diverse data inputs including lag demands, rolling statistics, product lifecycle information, and channel metrics, providing rich context for the forecasting models. Our extensive benchmarking of file formats led to the adoption of Apache Feather, which delivered superior performance in terms of compression efficiency, version stability, and read/write performance compared to traditional alternatives.

Perhaps most significantly, this work demonstrated the critical importance of human-in-the-loop ensembling in realizing the full potential of machine learning forecasting. By integrating ML forecasts with human expertise through interactive dashboards and transparent performance metrics, we established a framework that preserved human agency while leveraging algorithmic advantages. This balanced approach led to substantial operational benefits, including a 28.5% reduction in inventory levels and a 34.4% improvement in forecast accuracy over three years. The three-pronged implementation approach—visual dashboards, analytics as guidance, and seamless integration into planning tools—proved essential for successful adoption at scale.

Our extensions into predictive optimization further highlighted an important insight: good forecasts do not automatically translate to good decisions. Through simulation studies, we demonstrated that the nature of forecast errors—not merely their magnitude—significantly impacts decision quality, suggesting limitations in the traditional predict-then-optimize paradigm. Finally, we propose a predictive optimization framework that integrates forecasting with decision-making in the supply chain domain.

Collectively, this research makes several key contributions to both theory and practice. It advances our understanding of how machine learning can enhance demand forecasting while preserving valuable human expertise. It establishes a blueprint for enterprise-scale MLOps implementation that addresses both technical and organizational challenges. Most importantly, it demonstrates that through

thoughtful integration of analytics and human judgment, significant operational improvements can be achieved, delivering substantial business value in inventory management, forecast accuracy, and resource allocation.

Several promising directions exist for extending this research in the future. Hybrid forecasting models combining tree-based methods with deep learning architectures could further enhance performance, particularly for capturing long-range dependencies and complex seasonal patterns. Various hierarchal forecasting algorithms and their ensembles could also be considered. Improvements in model interpretability would strengthen the human-in-the-loop framework, enabling planners to better understand and trust ML forecasts in ambiguous scenarios. Automatic feature discovery could expand the range of inputs considered by the model, potentially identifying previously overlooked demand signals.

The most compelling direction for future work lies in end-to-end predictive optimization for supply chain management. Building on our simulation results and Mao et al. (2023), an integrated framework directly connecting demand forecasting to production and inventory decisions could yield significant improvements over traditional sequential approaches. Such a system would be particularly valuable in contexts characterized by high demand uncertainty, complex constraints, and asymmetric costs of errors.

The framework developed in this dissertation provides a solid foundation for these future enhancements, establishing both the technical infrastructure and organizational practices necessary to support continued innovation in machine learning-based demand forecasting and supply chain optimization. As computational capabilities and algorithmic approaches evolve, the human-in-the-loop approach presented here offers a balanced pathway for enterprise adoption that maintains the critical role of human expertise while leveraging the analytical power of machine learning.

Bibliography

Agrawal, A., Gans, J., and Goldfarb, A. (2020). How to win with machine learning. *Harvard Business Review*. 24

Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5–6):594–621. 26

Alla, S., Adari, S. K., Alla, S., and Adari, S. K. (2021). *What is mlops?* Springer. 30

Alvarado-Valencia, J., Barrero, L. H., Önkal, D., and Dennerlein, J. T. (2017). Expertise, credibility of system forecasts and integration methods in judgmental demand forecasting. *International Journal of Forecasting*, 33(1):298–313. 23

Amou Najafabadi, F., Bogner, J., Gerostathopoulos, I., and Lago, P. (2024). An analysis of mlops architectures: A systematic mapping study. In *European Conference on Software Architecture*, pages 69–85. Springer. 29, 30, 31

Archer, B. H. (1980). Forecasting demand: quantitative and intuitive techniques. *International Journal of Tourism Management*, 1(1):5–12. 17

Arvan, M., Fahimnia, B., Reisi, M., and Siemsen, E. (2019). Integrating human judgement into quantitative forecasting methods: A review. *Omega*, 86:237–252. 22, 23

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. 20

Baier, L., Jöhren, F., and Seebacher, S. (2019). Challenges in the deployment and operation of machine learning in practice. In *ECIS*, volume 1. 29, 31, 32

Bandara, K., Bergmeir, C., and Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert systems with applications*, 140:112896. 26

Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., et al. (2017). Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1387–1395. 28, 30

Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR. 38

Billington, C., Callioni, G., Crane, B., Ruark, J. D., Rapp, J. U., White, T., and Willems, S. P. (2004). Accelerating the profitability of hewlett-packard's supply chains. *Interfaces*, 34(1):59–72. 9

Blattberg, R. C. and Hoch, S. J. (1990). Database models and managerial intuition: 50% model + 50% manager. *Management Science*, 36(8):887–899. 22, 23

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons. 2, 19

Box, G. E. P. and Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA. 19

Brau, R., Aloysius, J., and Siemsen, E. (2023). Demand planning for the digital supply chain: How to integrate human judgment and predictive analytics. *Journal of operations management*, 69(6):965–982. 22

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32. 3

Brown, R. G. (1956). *Exponential smoothing for predicting demand*. Little. 2

Carbonneau, R., Laframboise, K., and Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European journal of operational research*, 184(3):1140–1154. 35

Cargille, B. and Branvold, D. (2000). Diffusing supply chain innovations at hewlett-packard company: Applications of performance technology. *Performance Improvement Quarterly*, 13(4):6–15. 9

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. 3

Curtland, C., Neto, P., and Ghozeil, A. (2022). Hp inc.. advanced analytics powers technology in the service of humanity. 34

De Gooijer, J. G. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473. 20

Deng, T., Zhao, Y., Wang, S., and Yu, H. (2021). Sales forecasting based on lightgbm. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 383–386. IEEE. 26

Deng, Y., Zhang, X., Wang, T., Wang, L., Zhang, Y., Wang, X., Zhao, S., Qi, Y., Yang, G., and Peng, X. (2023). Alibaba realizes millions in cost savings through integrated demand forecasting, inventory management, price optimization, and

product recommendations. *INFORMS Journal on Applied Analytics*, 53(1):32–46. 24, 25

Dietvorst, B. J., Simmons, J. P., and Massey, C. (2018). Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management science*, 64(3):1155–1170. 80

Dodin, P., Xiao, J., Adulyasak, Y., Alamdari, N. E., Gauthier, L., Grangier, P., Lemaitre, P., and Hamilton, W. L. (2023). Bombardier aftermarket demand forecast with machine learning. *INFORMS Journal on Applied Analytics*. 24, 25, 29, 32

Ferreira, K. J., Lee, B. H. A., and Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & service operations management*, 18(1):69–88. 24, 29

Fildes, R. (1991). Efficient use of information in the formation of subjective industry forecasts. *Journal of Forecasting*, 10(6):597–617. 23

Fildes, R., Goodwin, P., Lawrence, M., and Nikolopoulos, K. (2009). Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. *International journal of forecasting*, 25(1):3–23. 7

Fildes, R., Ma, S., and Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318. 6

Fildes, R., Nikolopoulos, K., Crone, S. F., and Syntetos, A. A. (2008). Forecasting and operational research: a review. *Journal of the Operational Research Society*, 59(9):1150–1172. 35

Fildes, R. and Petropoulos, F. (2015). Improving forecast quality in practice. *Foresight: The International Journal of Applied Forecasting*, 36:5–12. 22

Fildes, R. and Stekler, H. (2002). The state of macroeconomic forecasting. *Journal of macroeconomics*, 24(4):435–468. 21

Findley, D. F. (1983). On the use of multiple models for multi-period forecasting. In *Proceedings of Business and Economic Statistics, American Statistical Association*, pages 528–531. 26

Francis, T. (2012). Plain cigarette packaging begins in australia. *The Lancet*, 380(9857):1896. 20

Franses, P. H. and Legerstee, R. (2011). Experts' adjustment to model-based sku-level forecasts: does the forecast horizon matter? *Journal of the Operational Research Society*, 62(3):537–543. 21, 23

Franses, P. H. and Legerstee, R. (2013). Do statistical forecasting models for sku-level data benefit from including past expert knowledge? *International Journal of Forecasting*, 29(1):80–87. 23

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232. 40

Gardner, E. S. (1990). Evaluating forecast performance in an inventory control system. *Management science*, 36(4):490–499. 5, 18, 25

George, J. and Saha, A. (2022). End-to-end machine learning using kubeflow. In *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 336–338. 30

Goodwin, P. (2002). Integrating management judgment and statistical methods to improve short-term forecasts. *Omega*, 30(2):127–135. 22

Goodwin, P., Hoover, J., Makridakis, S., Petropoulos, F., and Tashman, L. (2023). Business forecasting methods: Impressive advances, lagging implementation. *Plos one*, 18(12):e0295693. 79

Green, K. C. and Armstrong, J. S. (2012). Demand forecasting: Evidence-based methods. *Available at SSRN 3063308*. 23

Gupta, S. (1994). Managerial judgment and forecast combination: An experimental study. *Marketing Letters*, 5:5–17. 21

Haines, S. (2022). Workflow orchestration with apache airflow. In *Modern Data Engineering with Apache Spark: A Hands-On Guide for Building Mission-Critical Streaming Applications*, pages 255–295. Springer. 30

Hamzaçebi, C., Akay, D., and Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications*, 36(2):3839–3844. 25, 26

Harshvardhan, Liu, C., Curtland, C., and Ghozeil, A. (2025a). Forecasting print demand with machine learning at hp inc. *Foresight: The International Journal of Applied Forecasting*, Forthcoming(Forthcoming). Accepted for publication. 15, 56, 69

Harshvardhan, M., Curtland, C., Hwang, J., VanDam, C., Ghozeil, A., Neto, P. A., Marie, F., and Liu, C. (2025b). Print demand forecasting with machine learning at hp inc. *INFORMS Journal of Applied Analytics*, Forthcoming(Forthcoming). Accepted for publication. 15, 36, 56, 69

Hartzel, K. S. and Wood, C. A. (2017). Factors that affect the improvement of demand forecast accuracy through point-of-sale reporting. *European Journal of Operational Research*, 260(1):171–182. 27

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press. 19

Hazelwood, K., Bird, S., Brooks, D., Chintala, S., Diril, U., Dzhulgakov, D., Fawzy, M., Jia, B., Jia, Y., Kalro, A., et al. (2018). Applied machine learning at facebook: A datacenter infrastructure perspective. In *2018 IEEE international symposium on high performance computer architecture (HPCA)*, pages 620–629. IEEE. 31

Heikkilä, J. (2002). From supply to demand chain management: efficiency and customer satisfaction. *Journal of operations management*, 20(6):747–767. 18

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., and Grosch, G. (2022). Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6. 37

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. 20

Hogarth, R. M. and Makridakis, S. (1981). Forecasting and planning: An evaluation. *Management science*, 27(2):115–138. 20

Howard, J. (2019). Practical deep learning. Online Course. 44

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts. 6, 19, 20

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688. 47

Ibrahim, R., Kim, S.-H., and Tong, J. (2021). Eliciting human judgment for prediction algorithms. *Management Science*, 67(4):2314–2325. 22

John, M. M., Olsson, H. H., and Bosch, J. (2021). Towards mlops: A framework and maturity model. In *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 1–8. IEEE. 54

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154. 3, 26, 37, 40

Khosrowabadi, N., Hoberg, K., and Imdahl, C. (2022). Evaluating human behaviour in response to ai recommendations for judgemental forecasting. *European Journal of Operational Research*, 303(3):1151–1167. 7

Kreuzberger, D., Kühl, N., and Hirschl, S. (2023). Machine learning operations (mlops): Overview, definition, and architecture. *IEEE access*, 11:31866–31879. 3

Kurtuluş, M., Ülkü, S., and Toktay, B. L. (2012). The value of collaborative forecasting in supply chains. *Manufacturing & Service Operations Management*, 14(1):82–98. 27

Laval, C., Feyhl, M., and Kakouros, S. (2005). Hewlett-packard combined or and expert knowledge to design its supply chains. *Interfaces*, 35(3):238–247. 9

Lawrence, M., Goodwin, P., O'Connor, M., and Önkal, D. (2006). Judgmental forecasting: A review of progress over the last 25 years. *International Journal of forecasting*, 22(3):493–518. 20, 21, 22

Lawrence, M. J., Edmundson, R. H., and O'Connor, M. J. (1985). An examination of the accuracy of judgmental extrapolation of time series. *International Journal of Forecasting*, 1(1):25–35. 22

Lawrence, M. J., Edmundson, R. H., and O'Connor, M. J. (1986). The accuracy of combining judgemental and statistical forecasts. *Management Science*, 32(12):1521–1532. 22

Lee, H. L. (2002). Aligning supply chain strategies with product uncertainties. *California management review*, 44(3):105–119. 5

Liu, J., Lin, S., Xin, L., and Zhang, Y. (2023). Ai vs. human buyers: A study of alibaba's inventory replenishment system. *INFORMS Journal on Applied Analytics*, 53(5):372–387. 33

Luo, Y., Li, A., Soman, D., and Zhao, J. (2023). A meta-analytic cognitive framework of nudge and sludge. *Royal Society Open Science*, 10(11):230053. 80

Mäkinen, S., Skogström, H., Laaksonen, E., and Mikkonen, T. (2021). Who needs mlops: What data scientists seek to accomplish and how can mlops help? In *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pages 109–112. IEEE. 32

Makridakis, S. and Hibon, M. (2000). The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476. 26

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13. 19, 25, 26

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2021). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*. 26, 37

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364. 26, 37

Manary, M. P., Wieland, B., Willems, S. P., and Kempf, K. G. (2019). Analytics makes inventory planning a lights-out activity at intel corporation. *INFORMS Journal on Applied Analytics*, 49(1):52–63. 6

Mao, W., Liu, C., Huang, Y., Zu, Z., Harshvardhan, M., Wang, L., and Zheng, B. (2023). End-to-end inventory prediction and contract allocation for guaranteed

delivery advertising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1677–1686. 15, 16, 82, 89, 98

Marcellino, M., Stock, J. H., and Watson, M. W. (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of Econometrics*, 135(1-2):499–526. 25, 26

Mboweni, T., Masombuka, T., and Dongmo, C. (2022). A systematic review of machine learning devops. In *2022 international conference on electrical, computer and energy technologies (ICECET)*, pages 1–6. IEEE. 54

McElroy, T. (2015). When are direct multi-step and iterative forecasts identical? *Journal of Forecasting*, 34(4):315–336. 26

O'Brien, A. P. (1989). How to succeed in business: Lessons from the struggle between ford and general motors during the 1920s and 1930s. *Business and Economic History*, pages 79–87. 2

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. 24

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. (2019). N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*. 24

Paleyes, A., Urma, R.-G., and Lawrence, N. D. (2022). Challenges in deploying machine learning: a survey of case studies. *ACM computing surveys*, 55(6):1–29. 28

Petropoulos, F., Kourentzes, N., Nikolopoulos, K., and Siemsen, E. (2018). Judgmental selection of forecasting models. *Journal of Operations Management*, 60:34–46. 7

Petropoulos, F. and Siemsen, E. (2023). Forecast selection and representativeness. *Management Science*, 69(5):2672–2690. 7

Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., and Shen, Z.-J. (2023). A practical end-to-end inventory management model with deep learning. *Management Science*, 69(2):759–773. 25

Ritzman, L. P. and King, B. E. (1993). The relative significance of forecast errors in multistage manufacturing. *Journal of Operations Management*, 11(1):51–65. 5

Rodriguez-Lujan, I., Elkan, C., Santa Cruz Fernández, C., Huerta, R., et al. (2010). Quadratic programming feature selection. *Journal of Machine Learning Research*. 44

Sagaert, Y. R., Aghezzaf, E.-H., Kourentzes, N., and Desmet, B. (2018). Temporal big data for tactical sales forecasting in the tire industry. *Interfaces*, 48(2):121–129. 25, 27

Salinas, D., Flunkert, V., and Gasthaus, J. (2017). Deepar: Probabilistic forecasting with autoregressive recurrent networks. arxiv 2017. *arXiv preprint arXiv:1704.04110*. 24

Sanders, N. R. (1992). Accuracy of judgmental forecasts: A comparison. *Omega*, 20(3):353–364. 22

Sanders, N. R. and Ritzman, L. P. (1995). Bringing judgment into combination forecasts. *Journal of Operations Management*, 13(4):311–321. 21

Sanders, N. R. and Ritzman, L. P. (2004). Integrating judgmental and quantitative forecasts: methodologies for pooling marketing and operations information. *International Journal of Operations & Production Management*, 24(5):514–529. 23

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28. 28, 29, 30

Seifert, M., Siemsen, E., Hadida, A. L., and Eisingerich, A. B. (2015). Effective judgmental forecasting in the context of fashion products. *Journal of Operations Management*, 36:33–45. 5, 6

Simatupang, T. M. and Sridharan, R. (2002). The collaborative supply chain. *The international journal of logistics management*, 13(1):15–30. 27

Simatupang, T. M. and Sridharan, R. (2005). An integrative framework for supply chain collaboration. *The international Journal of Logistics management*, 16(2):257–274. 6

Sinha, S. and Lee, Y. M. (2024). Challenges with developing and deploying ai models and applications in industrial systems. *Discover Artificial Intelligence*, 4(1):55. 28

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. 21, 23, 35

Tabassam, A. (2023). Mlops: a step forward to enterprise machine learning. *arXiv preprint arXiv:2305.19298*. 30

Taylor, S. J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45. 19, 24

Taylor, T. A. and Xiao, W. (2010). Does a manufacturer benefit from selling to a better-forecasting retailer? *Management Science*, 56(9):1584–1598. 27

Uber Engineering (2017). Meet michelangelo: Uber's machine learning platform. Accessed: 2025-02-26. 31

Wang, C. and Wu, Q. (2019). FLAML: fast and lightweight hyperparameter optimization for automl. *CoRR*, abs/1911.04706. 66

Ward, J., Zhang, B., Jain, S., Fry, C., Olavson, T., Mishal, H., Amaral, J., Beyer, D., Brecht, A., Cargille, B., et al. (2010). Hp transforms product portfolio management with operations research. *Interfaces*, 40(1):17–32. 9

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244. 44

Worten, B. (2003). Beware the promises of demand forecasting systems. https://www.cio.com/article/272812/enterprise-software-beware-the-promises-of-demand-forecasting-systems.html. 3, 21

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., et al. (2018). Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4):39–45. 30, 31, 34, 58

Zellner, M., Abbas, A. E., Budescu, D. V., and Galstyan, A. (2021). A survey of human judgement and quantitative forecasting methods. *Royal Society Open Science*, 8(2):201187. 7

Zhang, Y., Zhu, C., and Wang, Q. (2020). Lightgbm-based model for metro passenger volume forecasting. *IET Intelligent Transport Systems*, 14(13):1815–1823. 26

Vita

Harshvardhan was born on August 17, 1998, in Jhumri Tilaiya, Jharkhand, India. He completed his high schooling from Sainik School Tilaiya in 2016. In 2019, he began his Bachelor of Arts (B.A.) in Foundations of Management at the Indian Institute of Management (IIM) Indore. During his undergraduate studies, he participated in the Erasmus+ scholarship program, studying at the University of Latvia in Riga. After completing his B.A., he continued at IIM Indore to pursue a Master of Business Administration (M.B.A.), which he completed in 2021. Following his graduate education, Harshvardhan worked at Aspect Ratio in Pune, India, where he contributed to projects for Merck Pharmaceuticals. Later in 2021, he began his doctoral studies in Business Analytics at the Haslam College of Business. From 2022 to 2023, he served as a data science intern with the Strategic Planning and Modeling team at HP Inc., working from the Vancouver, Washington office and remotely. His research focuses on applications of machine learning and artificial intelligence, business analytics, and demand forecasting.