LUISS

Master Degree in Data Science and Management (LM-91)

International Operations and Global Supply Chain

AI-based decision making in supply chains: prototyping a ML-driven APS (Advanced Planning System) using Walmart open-source datasets

Prof.ssa Lorenza Morandini

Relatore

Prof. Paolo Spagnoletti

Correlatore

Marco Coci

Candidato

Academic Year 2024–2025

Abstract

This experimental research addresses a prevalent inefficiency in retail operations: waste generated through imprecise demand forecasting and suboptimal inventory management. Adopting a Design Science Research (DSR) framework, the study develops a system that combines machine learning techniques to enhance prediction accuracy and optimize inventory levels.

An analysis of Walmart data spanning over 58 million daily observations (58,327,370 store-day records), covering three product categories (Foods, Hobbies, and Household) across ten stores in California, Texas, and Wisconsin from 2011 to 2016, revealed marked regional variations in demand patterns. These data represent 3,049 distinct products and generate 30,490 unique time series at the most granular level, with differences of up to 30% in response to events for identical products across locations. This geographic variability in customer behavior, influenced by numerous factors often invisible to the human eye, underscores the critical importance of capturing local nuances before establishing a functional supply chain.

To address the challenge of real-world retail data, it was necessary to manage intermittent demand patterns, where over 80% of products in categories like Hobbies exhibit zero sales on more than 60% of days. An adaptive forecasting framework was implemented using a two-stage approach that separately models the probability of sale and the expected quantity. This methodology distinguishes between whether a product will sell on a given day and how much will sell if a transaction occurs. The forecasting models achieved promising accuracy, with average Mean Absolute Errors between 0.46 and 1.16 units per day across product categories. The evaluation was conducted on a validation dataset that contain 30 days of sales following the training period, so these metrics reflect performance over a one-month horizon.

The resulting model was used in a custom APS system that dynamically recalculates Reorder Points based on predicted demand patterns, automatically adjusting safety stocks using factors specific to each demand profile while maintaining user-selected service levels (90%, 95%, or 99%). Current inventory levels at the specific store are also provided as input to the system.

The core contribution of this research in the DSR context is twofold: the creation of a technological artifact that integrates machine learning forecasts into an Advanced Planning and Scheduling (APS) system, and the development of design knowledge on effective approaches to intermittent demand forecasting in retail contexts. This system aligns safety stock levels with desired service rates while accounting for operational constraints. For new

products lacking historical data, similarity-based forecasting methods were implemented, leveraging patterns from comparable established items.

This research demonstrates how data-driven approaches can transform supply chain management by capturing nuanced demand signals across regions, product categories, and time periods, ultimately reducing stockouts, waste, and excess inventory while improving overall operational efficiency. The DSR framework enables balanced contributions to both practical operations management and theoretical understanding of retail demand forecasting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Supply Chain and Its Transformation in the AI Era

1.1 Supply Chain and Its Challenges

Modern supply chains have evolved into intricate global networks characterized by interdependence, volatility, and data abundance. As products flow from suppliers to retailers, businesses face fragmented demand patterns, shortened product lifecycles, and heightened customer expectations for rapid deliveries (Chopra and Meindl 2019). This complexity, amplified by expanding e-commerce channels, creates vulnerability where even minor shipment delays or forecasting errors can cause stockouts, lost sales, excessive inventory costs, or waste.

One of the core challenges is the sheer amount of data collected along the supply chain from supplier to point-of-sale, and in real-time tracking of logistics. Conventional methods of forecasting will not capture subtle patterns or be able to move fast enough to keep up with changes in demand, especially if products demonstrate patterns of intermittent demand in interspersed zero-sales and rapid peaks. The challenge intensifies further if several thousand SKUs (stock keeping unit) need to be handled in combination in several different types of stores and geographies.

Additionally, another complexity is that different regions and product categories react differently to occasions and promotions. Black Friday, for example, may cause electronics sales in one area to grow, while in another market, household items may see the most significant sales lift. Government aid programs, specifically SNAP (Supplemental Nutrition Assistance Program), may also affect the demand in some states more than in others. Therefore, differing strategies may be required state to state and product to product. These variations expose the limits of a one-size-fits-all approach in modern retail: universal methods simply won't work, so it's essential to adopt differentiated tactics to each market

and product.

Global disruptions exemplify these vulnerabilities on a larger scale. The 2021 Suez Canal blockage demonstrated how a single bottleneck can paralyze interconnected supply networks, with effects reverberating globally (Kelkar, Marya, and Mysore 2024). Research indicates companies risk losing over 40% of their annual profits due to supply chain disruptions over a decade (Figure 1.1). Early identification and mitigation of such risks has become essential, driving organizations toward more sophisticated analytics approaches that can anticipate and respond to potential disruptions before they severely impact operations (Agrawal et al. 2024) .

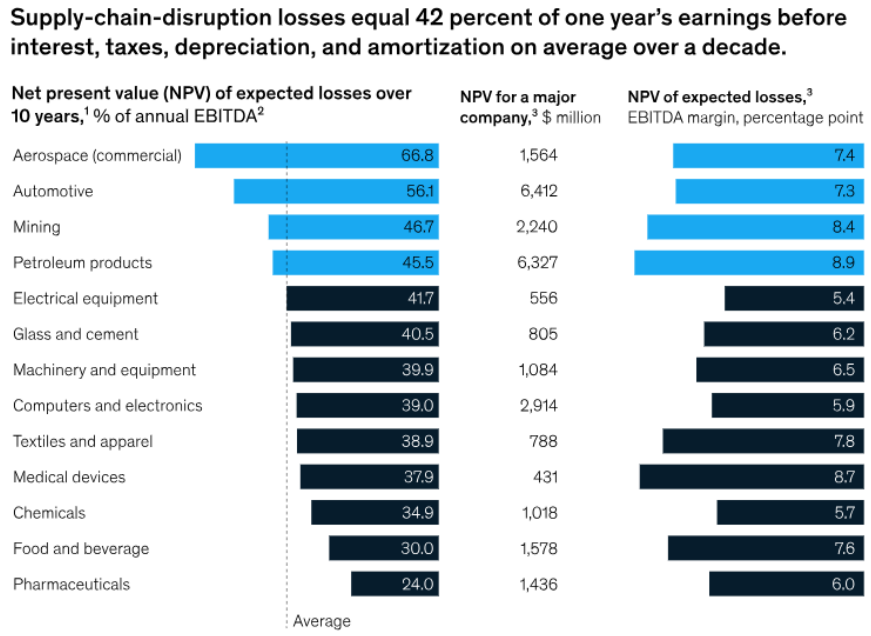

Figure 1.1: Supply chain disruption losses can equal 42% of a company's annual EBITDA (adapted from McKinsey & Company).

Inventory decisions themselves hinge on a few core metrics, most notably **Safety Stock** and the **Reorder Point (ROP)**. Safety stock acts as a buffer to absorb variability in both demand and lead time. It ensures product availability by compensating for deviations from expected consumption and delays in procurement. It is calculated as:

$$SS = z \cdot \sigma \cdot \sqrt{L}$$

where z is the Z-score corresponding to the desired service level, σ is the standard deviation of demand, and L is the lead time in days. This formula enables inventory systems to maintain availability despite fluctuations, while balancing cost efficiency.

The **Reorder Point (ROP)** identifies the stock level at which a replenishment order should be placed. It is given by:

$$ROP = \mu \cdot L + SS$$

where μ is the average demand during lead time (the time between placing an order and receiving it), and SS is the safety stock calculated using the formula above. The ROP represents the inventory threshold that triggers a replenishment order, ensuring that stock levels remain sufficient to meet customer demand during the procurement period while maintaining the desired service level. These concepts are particularly relevant in the Walmart case, where factors such as seasonality, localized promotions, and socioeconomic programs (e.g., SNAP) introduce high variability in daily demand. The demand dynamics for Foods, Hobbies, and Household categories differ significantly, reinforcing the need for region and category specific forecasting strategies.

Lead Time, defined as the time elapsed between placing an order and receiving the goods, varies based on production location, product turnover rate, and seasonality. Products with rapid turnover or high volatility may benefit from shorter lead times, often achieved through nearshoring or domestic manufacturing, to enable agile replenishment cycles. Accurate demand forecasting not only ensures product availability but also informs strategic choices such as production location.

Service Level is the expected (in B2C) or contractually defined (in B2B) performance level of a logistics service. It encompasses delivery frequency, reliability, speed, and adherence to time windows. Higher service levels help reduce lost sales and enhance customer satisfaction but typically require maintaining more safety stock, increasing inventory investment. Inventory management decisions rely on these metrics to guide operational and strategic decisions, especially in complex retail networks like Walmart's.

These fundamental inventory management metrics heavily depend on the quality of demand forecasts. For this reason, advanced forecasting techniques are radically transforming decision-making processes in modern supply chain management.

1.2 How Demand Forecasting is Changing the Supply Chain

Today, **demand forecasting** has become an important element for guiding supply chain decisions, influencing inventory policies, service levels, and production planning. Its primary objective is to predict how many units customers will likely purchase over specific time intervals by considering local variations and external factors such as events, promotions, and macroeconomic indicators.

Historically, many companies relied on basic spreadsheet forecasts or deterministic

statistical models that only considered historical sales. These methods often struggled in highly volatile environments or when demand was intermittent. The transition from these traditional approaches to data-driven methodologies represents a revolution in supply chain management, enabling predictive capabilities that were previously unattainable. Today, a *data-driven* approach leverages:

- **Granular data at store level** (including point-of-sale, inventory records, lead times).
- **Machine Learning models** that capture seasonalities, local promotions, and exceptional events like Valentines Day or important sports finals.
- **Real-time updates** that allow frequent adjustments of inventory targets to match observed demand patterns (McKinsey & Company 2020; McKinsey & Company 2019).

These integrated models make it possible to dynamically recalculate reorder points (ROP) and safety stocks while also considering other constraints, such as warehouse capacity or supplier lead times. For instance, as highlighted by (Agrawal et al. 2024), linking advanced predictions with an *Advanced Planning and Scheduling (APS)* system can synchronize procurement, replenishment, and production activities more efficiently.

Accurate forecasts not only minimize lost sales due to stock-outs but also prevent overstocking and obsolescence. Kelkar studies show that forecasting errors can reduce profits by up to 45% over a decade if the company lacks preventive measures. Additionally, data-driven models allow simulation of "what-if" scenarios enabling managers to evaluate alternative sourcing plans, diversified suppliers, or new transportation routes. This is especially useful when disruptive events occur, such as unexpected surges in demand or major global interruptions (Kelkar, Marya, and Mysore 2024).

The practical application of these principles becomes particularly evident when analyzing large-scale retail sales data, as demonstrated in the present study.

This project with the Walmart data was a key example of these principles. 58 million rows were analyzed, consisting of the sales history for three product lines (Foods, Hobbies, Household) across ten different U.S. locations. The differences in demand patterns across regions was particularly intriguing, with the most surprising being how variable government assistance programs were in driving sales by state. To illustrate, in Wisconsin, SNAP (Supplemental Nutrition Assistance Program) payment days created food sale peaks of as much as 30%, whereas in California, the effect was significantly less, in the range of approximately 10%.

These regional differences were completely invisible in the aggregate data, thus reaffirming the belief that contemporary supply chains needed to support forecasts with the ability to capture and respond to patterns that differ by region, product category, and time.

The experience with Walmart data illustrates how implementing advanced forecasting techniques translates into concrete and measurable operational advantages such as:

1. **Reduced Stock-Outs:** Predictive models can anticipate sudden demand spikes, especially around promotional periods or regional events, given what has happened previously and how the model has been trained.

2. **Optimized Inventory Costs:** By leveraging machine learning forecasting models that accurately predict future sales volumes and patterns, companies can calculate more precise safety stock requirements. This precision prevents capital from being unnecessarily tied up in excess inventory while still maintaining sufficient stock to meet customer demand, effectively balancing the cost of holding inventory against the risk of stockouts.

3. **Scenario Analysis for Risk Management:** Through digital twins or advanced simulation tools, decision-makers can examine how a supply disruption in one region influences company-wide service levels (Agrawal et al. 2024).

Recent analyses by McKinsey indicate that AI-driven forecasting solutions can reduce forecast errors by **up to 50%** and trim inventory expenses by roughly **10%**, allowing organizations to respond faster to market changes. In some cases, planning cycles also become **40% faster**, which helps firms react swiftly to sudden shifts in demand or unforeseen supply chain disruptions (McKinsey & Company 2020; McKinsey & Company 2019).

Shifting from a purely historical forecast to a more intelligent one able to incorporate multiple internal and external signals has become a decisive factor in building a resilient, agile, and profitable supply chain. As suggested by (McKinsey & Company 2020), companies adopting AI forecasting have managed to halve forecasting errors while cutting inventory costs, ultimately improving revenue, and their ability to respond quickly to sudden changes in demand.

These transformations in demand forecasting systems are part of a broader context of supply chain digitalization. In Italy, for instance, the government launched a strategic plan in 2016 to accelerate technology adoption and boost the Italian production system, particularly focusing on SMEs which represent the majority of Italian enterprises (Kazemargi and Spagnoletti 2020). Analysis of investments made by 1889 Italian SMEs revealed that 74% of Industry 4.0 investments were directed toward advanced manufacturing technologies, while 12.73% were invested in system integration that enables monitoring and control of products through data exchange, both across different departments and along the supply chain (Kazemargi and Spagnoletti 2020). This trend underscores the growing importance of data integration and predictive systems in modern supply chains.

To achieve this transformation from purely historical forecasts to intelligent models capable of incorporating multiple signals, machine learning emerges as a fundamental enabling technology, as we will explore in the following section.

1.3 The Role of Machine Learning

Machine learning (ML) in contemporary supply chains is now used as an enhancement tool by mitigating the basic shortcomings of traditional forecasting methods. In choosing the right forecasting models, there are several basic factors that need to be taken into account: data dimensions (quantity, quality, granularity), product characteristics (number of SKUs, seasonality patterns, frequency of sales), and whether the product is established with known history or newly introduced. The best model should also consider the general complexity of the patterns in demand and any operational restriction that may affect the inventory decisions.

Traditionally, forecasting has relied on statistical models such as ARIMA (AutoRegressive Integrated Moving Average). ARIMA models are among the most widely used approaches for modeling and predicting time series data due to their intuitive structure and interpretability. The ARIMA model integrates three distinct components, each designed to capture specific characteristics of the data (GeeksforGeeks 2024):

- **Autoregressive (AR) terms:** These predict future values by looking at patterns in past observations. An AR(p) model essentially says, "Today's value depends on what happened in the previous p days." For example, an AR(3) model would use data from the last three days to predict today.
- **Integrated (I) components:** This handles data that shows trends or changes in statistical properties over time. The I(d) component transforms unstable patterns into stable ones through d rounds of differencing, making the data easier to predict. For instance, rather than forecasting absolute sales, it possible to forecast day-to-day changes in sales.
- **Moving Average (MA) terms:** These focus on how unexpected events affect future values. An MA(q) component considers the previous q prediction errors to refine current forecasts. This helps the model adjust for recent surprises, like an unexpected sales spike due to unplanned promotions.

When seasonal patterns are evident, the ARIMA model can be extended into Seasonal ARIMA (SARIMA), explicitly incorporating parameters to capture recurring periodic fluctuations. Furthermore, ARIMAX models extend ARIMA by integrating exogenous

variables such as promotional activities, price changes, or external events, allowing for more accurate forecasts that explicitly account for external influences (GeeksforGeeks 2024).

Despite their simplicity, ease of handling, and efficiency in linear and behaved situations, ARIMA-based models are not capable in general of detecting sophisticated, non-linear patterns, long-run dependencies, and sudden structural variations in the data. These limitations have, therefore, inspired and led to the exploration and use of more versatile machine learning models that are capable of handling such complexities.

Modern data-driven approaches have evolved to overcome these limitations by effectively modeling complex patterns, non-linear interactions, and abrupt changes that traditional methods struggle to capture. Among the key models employed in practice are:

- **Tree-Based Machine Learning Models:** Algorithms such as Random Forests and Gradient Boosting Machines (GBM) provide reliable and interpretable frameworks for forecasting. In particular, boosting frameworks such as XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017) have gained prominence due to their excellent performance and computational efficiency. XGBoost employs regularized boosting techniques that introduce penalty terms in the model to prevent overfitting and improve predictions on new data. It leverages parallel tree-building and gradient-based optimization techniques, allowing efficient processing of large datasets while maintaining prediction accuracy.

  LightGBM further improve computational efficiency through two key innovations:

  - **Gradient-based One-Side Sampling (GOSS)**: This technique focuses the training process by prioritizing the most informative data points (those with larger gradient magnitudes) while still maintaining some randomness. This approach allows models to train faster without sacrificing accuracy, which proved essential when working with such a large dataset.
  - **Exclusive Feature Bundling (EFB)**: In datasets with many sparse features (mostly zeros), EFB intelligently identifies features that rarely appear simultaneously and groups them into bundles. This bundling dramatically reduces the complexity and dimensionality of the data, enabling faster model training without negatively impacting prediction quality.

- **Neural Networks:** Deep learning models have and are revolution forecasting thanks to their flexibility in capturing complex, temporal patterns. Among these models, Long Short-Term Memory (LSTM) networks are particularly powerful when analyzing time series data that includes long-term dependencies. Traditional

Recurrent Neural Networks (RNNs) often face difficulties known as vanishing or exploding gradients, meaning they struggle to maintain useful information from earlier time steps. LSTMs address these issues using specialized "gates" (input, forget, and output gates) to selectively preserve or discard historical information. This capability allows LSTMs to effectively capture recurring patterns, seasonalities, and rapid shifts in consumer demand.

- **Clustering and Similarity-Based Methods:** These methods are highly beneficial when dealing with intermittent demand or forecasting products with limited historical data. By grouping similar products or stores based on their historical sales patterns, clustering techniques allow models to learn from collective behaviors rather than relying solely on individual, potentially sparse, data points. Similarly, similarity-based forecasting leverages data from products or markets with comparable characteristics, enabling more accurate predictions even for new or rarely sold items.

For instance, in Walmart dataset that comprises extensive data and many products with zero sales and others that have large peaks. I started with preliminary experiment on a sub-sample of 15 SKUs. Picked products in each category: with extremely low sales with high peaks, with normal, consistent sales, and with highly event-stimulated variable sales. This research aimed to ascertain what model is best suited to the particular shape of the dataset, recognizing that no one model is always best as noted above; the optimum model varies with the specific inside of the data, this research will feature in greater detail in Chapter 3.1.

These planning models are integrated with an Advanced Planning and Scheduling (APS) system in order to dynamically recalculate parameters like the Reorder Point (ROP) and safety stock. This synchronization of procurement, replenishment, and production decisions in real time optimizes supply chain resilience overall.

1.4 Advanced Planning and Scheduling (APS)

Advanced Planning and Scheduling (APS) serves as the interface where forecasts translate into actionable inventory decisions. Unlike traditional Enterprise Resource Planning (ERP) systems that manage basic business processes, modern APS systems dynamically recalculate inventory policies based on continuously updated forecasts and incorporate complex optimization techniques especially valuable for multi-echelon inventory networks.

This research integrates these phases into a single decision-making process, where the machine learning model is directly optimized for operational metrics that matter to the business (inventory costs and service levels). This method, which Agrawal et al. (Agrawal

et al. 2024) call "optimal machine learning" (OML), simultaneously considers forecast uncertainty and operational constraints.

Implementation incorporates a 'digital twin' of the supply chain, a comprehensive virtual representation of the actual logistics network that dynamically simulates all processes, material flows, and operational decisions. This digital replica enables testing alternative scenarios in real-time, visualizing the impact of inventory parameter changes, and anticipating the effects of potential disruptions before they occur in the real world. In this implementation, the digital twin enables simulation of inventory policies and a dynamic safety stock calculation that adapts to changing demand patterns:

$$ROP = \mu_{ML} \cdot L + SS \tag{1.1}$$

$$SS = z \cdot \sigma_{ML} \cdot \sqrt{L} \tag{1.2}$$

where ROP is the reorder point, μ_{ML} is the *mean* (expected value) of the predicted demand (forward-looking, accounting for seasonality, events and price changes), L is the procurement lead time in days, SS is the safety stock, z is the service factor corresponding to the desired service level (e.g. $z = 1.65$ for 95 % and $z = 2.33$ for 99 %), and σ_{ML} is the *standard deviation* of the predicted demand, capturing expected variability under current conditions.

(Kelkar, Marya, and Mysore 2024) highlight that effective early warning systems must 'prompt action' rather than simply classify risk levels. Following this principle, APS implementation automatically triggers procurement actions when inventory falls below dynamically calculated thresholds. Their research shows this approach reduced parts shortages by 70% in manufacturing environments, a significant improvement that I want to replicate in retail context.

The semiconductor equipment case study from (Agrawal et al. 2024) provides a valuable benchmark. Their implementation achieved a fill rate increase from 77% to 85% without additional inventory investment by incorporating product installation data into inventory policies. Similarly, this APS algorithm integrates SNAP calendar data, regional sales patterns, events and change price which preliminary testing suggests can improve forecast accuracy.

Have designed APS system architecture following what (Agrawal et al. 2024) describe as 'end-to-end data architecture,' where 'the storage system should be able to pool data across teams, locations, and products and make it possible to update and access that information in near real time.' Using ML models allows solution to maintain hierarchical relationships between products, stores, and regions while capturing temporal patterns that

are very important for accurate forecasting.

The innovation in the approach is the transition from static, periodically updated planning parameters to a continuous, ML-driven recalculation strategy. This enables inventory policies that adapt to both gradual shifts in consumer behavior and sudden changes during promotional events or supply disruptions.

This Advanced Planning and Scheduling system, integrating machine learning forecasts with inventory optimization, aligns with the Design Science Research (DSR) framework proposed by (Baskerville et al. 2018). DSR emphasizes creating technological artifacts that solve practical problems while simultaneously developing theoretical knowledge. This research follows this approach by developing a practical system that optimizes inventory decisions while also contributing design principles for handling intermittent demand patterns in retail environments. The technological artifact (the ML-driven APS system) addresses real-world inventory challenges, while the design knowledge generated contributes to our understanding of effective approaches to intermittent demand forecasting.

Subsequent chapters will detail the methodology and implementation, focusing on the technical challenges of integrating machine learning forecasts with optimization algorithms for inventory management, and evaluating the system against real-world retail metrics using the Walmart dataset. Both the practical utility of the artifact and the theoretical contributions to design knowledge will be assessed in the evaluation chapters.

Chapter 2

Data, Exploratory Analysis, and Preprocessing

2.1 Dataset Description: The Walmart M5 Case

For purposes of research, I chose the Walmart M5 Forecasting dataset, providing a rich depiction of retail operations with fine-grained sales data. This dataset was originally published in association with the fifth Competition, hosted by the Makridakis Open Forecasting Center (MOFC) of the University of Nicosia. The competition involved predicting Walmart stores' sales on a daily basis, and thus constituted an excellent proving ground with which to assess traditional statistical techniques along with contemporary machine learning techniques in the context of demand prediction. (Kaggle 2020)

The M5 dataset is comprised of Walmart stores in three U.S. states with about 5.5 years of daily sales data. What is most valuable to this research is how dataset reflects real-world retail complexity, and that the data preserves the actual difficulties experienced by retailers in making forecasts in differing product categories, regions, and time periods.

Working with the dataset, I found that it has 3,049 distinct products spread across three major product categories: Foods, Hobbies, and Household. These products are present in 10 Walmart stores in California (4 stores), Texas (3 stores), and Wisconsin (3 stores). For the purposes of analysis, I converted the original data in the wide format to long format so that there are 58,327,370 store-day records. (27,461,070 in the case of Foods, 20,008,170 in the case of Household, and 10,858,130 in the case of Hobbies).

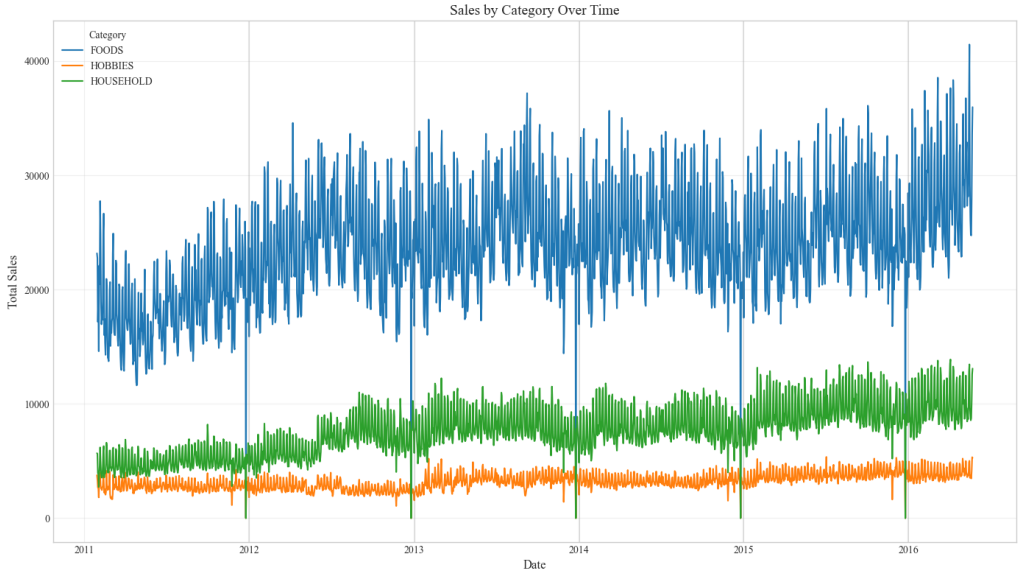

Figure 2.1: Trend of sales from 2011 to 2016

NB: The sharp drop in sales on December 25 is due to store closures during Christmas Day.

This transformation from wide to long format was essential for machine learning approach. In the original structure, each row represented a product-store combination with 1,913+ separate columns for daily sales. The restructured format created a single observation per product-store-date combination, significantly facilitating feature engineering and model development. While this increased the total row count substantially, presenting computational challenges. It created an ideal structure for time series modeling by enabling:

- Creation of temporal features (lags, moving averages, day-of-week effects)
- Integration of external variables and event indicators
- Application of machine learning algorithms requiring row-based observations
- Detection of complex relationships between sales and influencing factors across time

This transformation proved essential for capturing the nuanced patterns that drive retail demand across different products, locations, and time periods.

The temporal dimension of the dataset spans from January 29, 2011, to June 19, 2016, with daily granularity. This extensive time period enabled me to analyze long-term trends, seasonal patterns at various scales (yearly, monthly, weekly), and the impact of specific events on demand. The combination of products and stores generates 30,490 distinct time series at the most granular level, providing a reliable foundation for developing and testing machine learning forecasting models.

During the analysis, I worked with four datesets. The main ones, sales_train_validation.csv, contains daily unit sales for each product-store combination from day 1 to day 1,913. The sales_train_evaluation.csv extends this historical data to include days 1 to 1,941, which I used for validation purposes. Complementing the sales data, the calendar.csv file provides essential contextual information about each date, including weekends, holidays, special events (such as the Super Bowl and Black Friday), and SNAP (Supplemental Nutrition Assistance Program) disbursement days. Finally, the sell_prices.csv file contains daily pricing information for each product at each store location.

The validation framework of the dataset was of especial utility to this research approach. The competition officials deliberately established a validation window of 28 days (days 1,914 to 1,941) during which ground truth values were ultimately made available. This enabled me to use a validation approach in which I could train models against the early historical data, compare performance on actual values during the validation window, and value the model.

Perhaps most intriguing in the dataset is the existence of explanatory factors impacting demand patterns other than the one that I will define for the model. Aside from simple calendar characteristics such as day of the week and month, the data account for price fluctuations, promotional efforts, and holiday and holiday weekend impacts. While conducting exploratory analysis, I did observe significant variation in the impact these factors have by product category and by region. As one example, SNAP payment days were found to have more of an impact on the sale of food items in Wisconsin stores than in California stores.

The data also pose the realistic challenge of sparse demand, with the majority of products having sparse sales patterns defined by periods of zero demand separated by occasional buying or product that did not sell. This intermittency is highly variable across product categories and stores and necessitates sophisticated modeling that is capable of handling regular and irregular patterns of demand.

For purposes of evaluation, the competition utilized the Weighted Root Mean Squared Scaled Error (RMSSE) measure. RMSSE is most appropriate in retail forecasting environments because it mitigates several of the challenges associated with comparing forecasts between varied products. First, it scales the errors relative to the historical volatility of each series, allowing errors to be comparable between products of varying volumes. For example, an error in prediction of 10 units has very different consequences in the context of a high-volume product that has thousands of units sold every day compared with a specialty product that sells several units most every day. Second, the weighting part of RMSE captures differences in relative importance in the business across different products, with more importance assigned to products that have greater impacts on overall

revenue or units sold. This maps the evaluation metric onto genuine business concerns, where accuracy for high-volume or high-dollar products most often is more important than accuracy in low-volume products.

The richness and complexity of the Walmart M5 dataset offer the perfect platform on which to conduct research combining machine learning-based demand prediction with stock optimization using Advanced Planning and Scheduling systems. The dataset's direct application to actual stock management decisions allows me to show how AI-based methods will enhance supply chain efficiency.

Methodological Note: No specific outlier detection or treatment was performed on the dataset. The original data were used exactly as provided, was just merged with spark reflecting realistic conditions and challenges typically encountered in real-world retail forecasting scenarios.

2.2 Exploratory Data Analysis (EDA)

After preparing and understanding the structure of the dataset, was conducted a comprehensive exploratory analysis to discover patterns and relationships that would help modeling strategy. This exploration was important to develop effective prediction models, identifying relevant features, and determining the appropriate modeling approach for inventory optimization.

The goal of the EDA is to identify temporal patterns, category behaviors, and demand variability. I focused particularly on understanding how different product categories behave across time and locations and how each product behaves in a different way to the event, as this would directly impact inventory management strategies.

The data reveals significant long-term trends and annual seasonality patterns that vary by product category. Figure 2.1 displays the total sales by category over the full 2011–2016 period.

The time series analysis in Figure 2.1 reveals several insights:

- All categories demonstrate an overall positive growth trend from 2011 to 2016, with the steepest growth occurring in 2015-2016
- Foods category maintains the highest sales volume throughout the period, approximately 3-4 times larger than Household and 6-8 times larger than Hobbies
- Annual seasonality is evident across all categories, with regular peaks occurring during holiday seasons (November-December)

- Several significant drops in sales occur at consistent times each year, corresponding to major holidays when stores operate with reduced hours or close entirely
- The characteristic sawtooth pattern visible in the time series suggests a strong weekly cyclical pattern that warrants closer examination

Analysis of category-specific patterns revealed distinct characteristics with important implications for inventory management. Foods showed the highest volume with pronounced seasonality, suggesting the need for responsive inventory strategies. Household items demonstrated moderate but increasing demand with less volatility than Foods, potentially allowing for more efficient safety stock planning. Meanwhile, Hobbies, though smallest in volume, exhibited the most consistent year-over-year patterns, making historical data potentially more reliable for forecasting in this category.

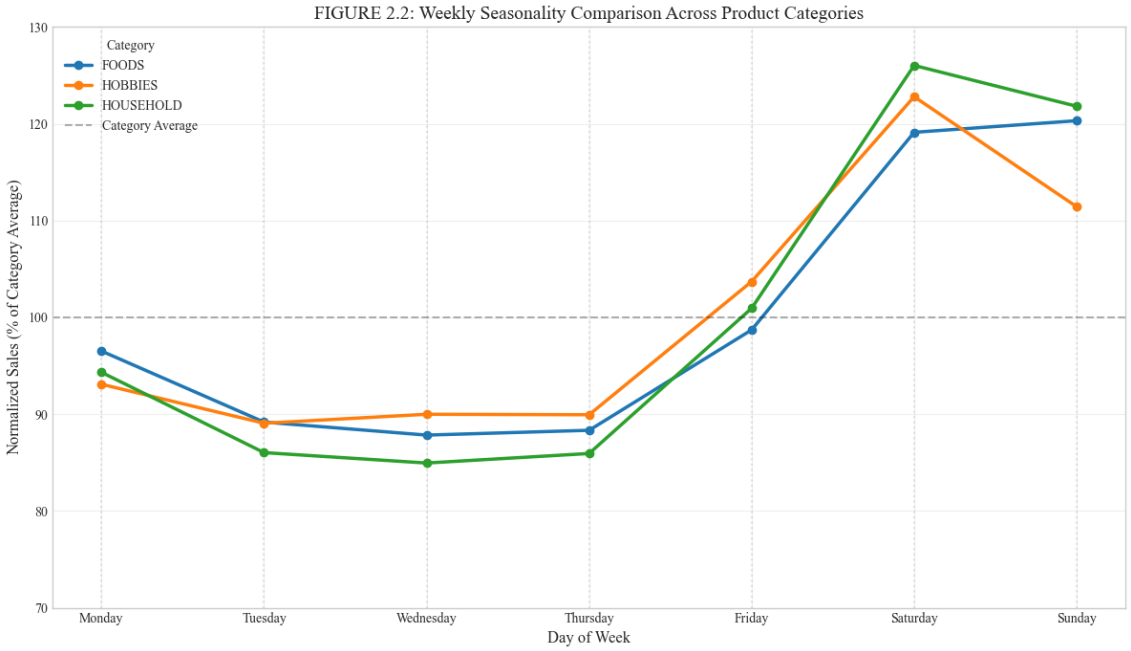

Figure 2.2: Weekly Seasonality Comparison Across Product Categories

Examining the data at a finer granularity reveals consistent weekly seasonality across all product categories. Figure 2.2 presents the normalized sales by day of week for each product category, demonstrating clear cyclical patterns so that the weekend the number of all product sales increase. Beyond weekly patterns, special events and holidays demonstrate significant impacts on sales dynamics that vary substantially between categories. To provide a clear visualization of these effects, I analyzed the top five events with the greatest impact on sales variability across product categories, as shown in Figure 2.3.

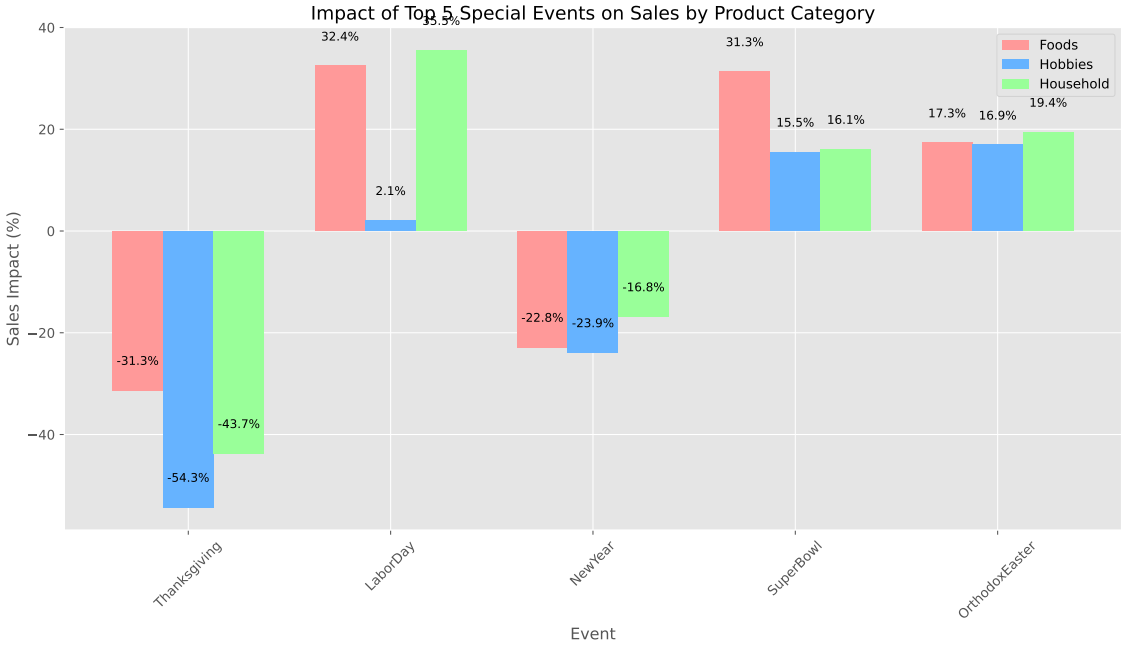

Figure 2.3: Impact of Top 5 Special Events on Sales by Product Category

A significant observation that emerged from the analysis is how different product categories respond to the same events. What initially seemed like a straightforward "event effect" turned out to be a complex web of category-specific and often counterintuitive responses.

Take Thanksgiving, for example. It was surprising to find substantial negative impacts across all categories, with Hobbies suffering the most dramatic decrease (-54.3%). The initial hypothesis suggested food sales would increase before the holiday, but the data revealed that the shopping pattern was more complex, consumers likely make purchases earlier, resulting in decreased store visits during the holiday itself.

Labor Day showed the opposite pattern. Foods (32.4%) and Household (35.5%) categories experienced significant boosts, while Hobbies showed minimal change (2.1%). This pattern suggests end-of-summer purchasing focused on home essentials rather than leisure items.

The Super Bowl impact was particularly interesting because it affected all categories positively, with Foods showing the strongest response (31.3%). This cross-category effect suggests that Super Bowl preparations involve not just food but a broader shopping behavior for entertainment needs.

These diverse responses to events challenged the initial modeling approach. It became clear that simply including an "event flag" variable would be insufficient, the models needed to capture the specific interaction between each event type and product category.

These diverse responses to special events highlight the need for sophisticated forecasting

approaches that consider not just the presence of an event, but the specific type of event and its unique interaction with each product category. Simply flagging days as "event days" in a forecasting model would be insufficient, as the data clearly show that different events drive dramatically different purchasing behaviors. For example, while Labor Day boosts Household sales by 35.5%, Thanksgiving reduces them by 43.7%, treating these events identically would lead to substantial inventory errors. Effective demand forecasting must therefore incorporate event-specific features that capture these nuanced relationships, allowing retailers to adjust inventory levels appropriately for each product category during different types of special event.

The SNAP (Supplemental Nutrition Assistance Program) effect analysis yielded one of the most surprising discoveries in this research. While anticipated some impact on food sales, the magnitude of regional differences was unexpected. As shown in Figure 2.4, Wisconsin stores demonstrated an exceptional response to SNAP disbursement days, with Foods sales increasing by 30.0% compared to non-SNAP days. This effect progressively decreased in Texas (+15. 6%) and California (+10. 4%).

These geographical variations necessitated a more advanced modeling approach than originally planned. Rather than implementing a single national SNAP effect variable, I developed state-specific interaction features to capture these regional differences accurately. This implementation required careful calibration to balance model complexity against potential overfitting.

The category-specific nature of the SNAP effect was equally revealing. While Foods sales showed substantial increases on disbursement days, Hobbies and Household categories remained largely unaffected (with modest effects between 1.9% and 3.8%). This pattern aligns with the program's intended purpose to support essential food purchases rather than discretionary items and provided valuable guidance for feature engineering in the forecasting models.

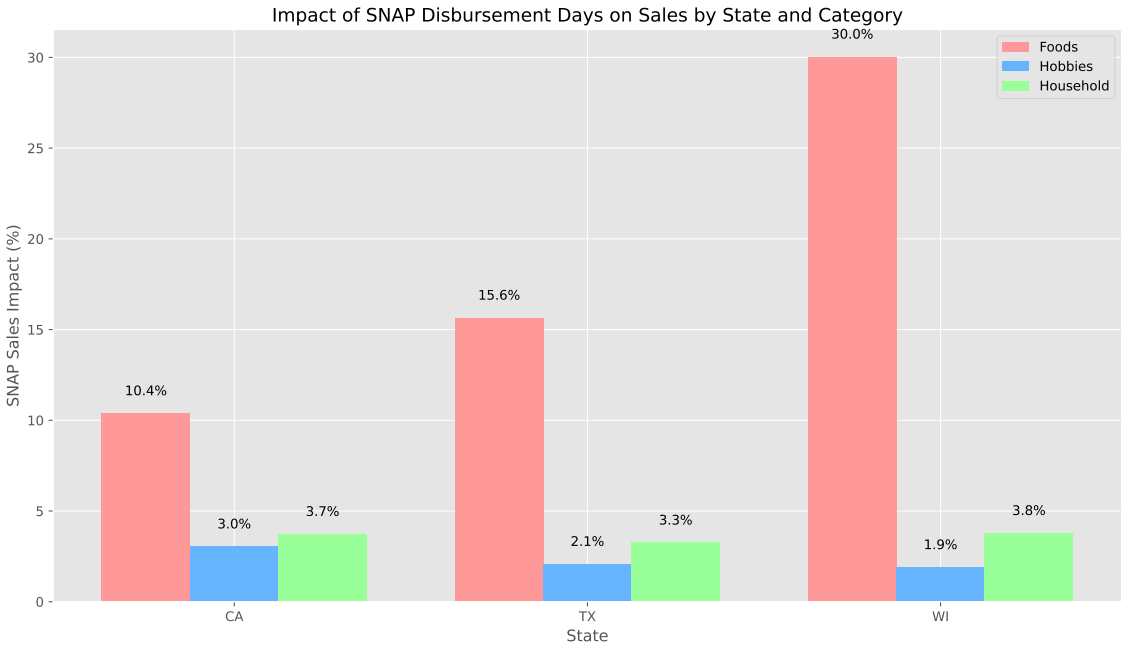

Figure 2.4: Impact of SNAP Disbursement Days on Sales by State and Product Category

Price sensitivity analysis revealed additional insights relevant to inventory optimization. Foods demonstrated the highest price elasticity (3.5% sales decrease per 1% price increase), followed by Household items (2.1%) and Hobbies products (1.7%). This variation extends to geographical differences as well, with California customers generally exhibiting higher price sensitivity than Wisconsin shoppers. These findings directly informed feature engineering strategy, with price-related variables receiving greater weight in Foods category models.

Examining the relationship between price changes and sales reveals notable demand spikes during promotional periods. When prices drop by 10% or more, Foods products experience an average sales increase of 127%, significantly higher than the increase 85% for household products and 73% for hobbies under similar discount conditions. This suggests that promotion features (decrease of price) would be especially valuable for forecasting food category demand.

The analysis also uncovers substantial variation in demand consistency across product categories. Foods products demonstrate the most consistent demand patterns, with only 26% of product-store combinations exhibiting extended periods without sales. In contrast, Hobbies and Household categories show much higher intermittency, with 42% and 37% of product-store combinations experiencing prolonged zero-sales periods, respectively. This intermittency characteristic has strong implications for model selection, as standard time-series approaches often perform poorly with sparse or irregular demand patterns, a challenge that will be addressed more comprehensively in section 2.3.

Another element that was introduced to the APS system was lead time, which was

not included in the given data set. Drawing on industry averages and retail supply chain behavior, I created differentiated lead times based on product category attributes: Food products were assigned short lead times (average between 3 and 5 days) that reflect their perishable nature and high turnover speeds, while the Household and Hobby segments were assigned longer and variable lead times (between 5 and 12 days) due to the complexity of their supply chains and the frequency of replenishment. This variability in lead time, alongside the patterns evidenced in demand, has direct impacts on safety stock planning and stock optimization strategies, as will be discussed in Chapter 4.

Variability in patterns both within product categories, between regions, and over time emphasizes the necessity of accommodating forecasting methods capable of describing intricate interactions between several variables. Traditional statistical techniques might not capture such advanced relationships that were discovered during the EDA, especially the differential SNAP impacts, event response variations in particular categories, and heterogeneous price sensitivities which our analysis exhibits. As shown in Section 3.2, machine learning using gradient boosting architectures and two-stage methods, are capable of capturing these non-linear patterns while being able to accommodate the heterogeneous patterns of intermittency in the exploratory analysis. The modeling structure will focus on state-specific SNAP interactions, event-type indicators, and category-specific features in order to capture these complex patterns of demand in Foods, Hobbies, as well as Household goods. Before constructing such forecasting models, however, the right methods of handling the intermittency in the data are necessary, which will be discussed in the next section.

2.3 Preprocessing and Analysis of Intermittent Demand

The data introduced several preprocessing problems that first required to be solved in order to build prediction models, with the pattern of intermittent demand most notable among these. Intermittent demand with frequent zero-sales periods and irregular non-zero demand periods is problematic to traditional methods that expect continuous and regular patterns.

To quantify this problem, I developed an analysis function that systematically examines zero-sales patterns in the dataset. This challenge of intermittent demand is not unique to this specific dataset but represents a widespread issue faced by supply chain companies across different sectors in real-world scenarios. The function performs two key calculations for each product-state combination:

- **Percentage of days with zero sales:** This metric quantifies how frequently a product remains unsold. For example, if a product shows no sales on 70 out of 100 days, it has a 70% zero-sales rate, indicating low sales frequency overall.

- **Average length of consecutive zero-sales runs:** This measures the typical duration of sales 'droughts' for a product. For instance, if a product experiences three periods without sales lasting 5, 8, and 12 days respectively, the average run length would be 8.33 days. This indicates how long a product typically remains on shelves without being purchased.

This dual approach enables differentiation between distinct intermittency patterns. For example, a product with a high percentage of zero-sales days (70%) but short average sequences (2-3 days) suggests rare but somewhat regular sales (perhaps weekend-only purchases). Conversely, the same percentage with long sequences (15-20 days) indicates clustered sales events separated by extended inactivity periods.

The analysis was applied separately to each product category to identify category-specific intermittency patterns. Analysis revealed that there were no products with zero sales across the entire observation period, indicating that every product was sold at least once. However, as shown in the results, many products still exhibit highly intermittent demand patterns, with sales occurring very infrequently. Figure 2.5 illustrates the distribution of zero-sales frequencies across the three product categories, highlighting what percentage of products in each category fall into different intermittency ranges.

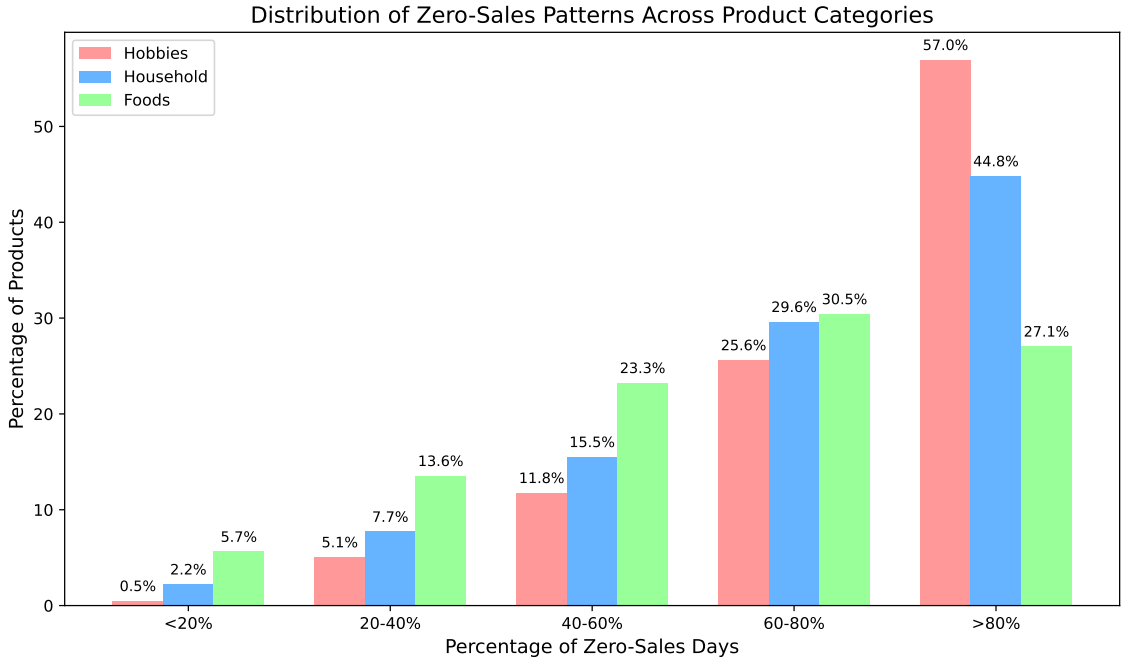

Figure 2.5: Zero sales distribution across sectors

The analysis revealed striking differences in demand consistency across product categories:

- **Hobbies:** The most intermittent category, with 82.6% of products showing zero sales on more than 60% of days (57.0% had zero sales on over 80% of days)
- **Household:** Slightly less extreme but still highly intermittent, with 74.4% of products having zero sales on more than 60% of days
- **Foods:** The most stable category, yet still challenging 57.6% of products showed zero sales on more than 60% of days

These patterns align with the weekly trends observed in the exploratory analysis, where Foods products demonstrated a more consistent demand. However, the prevalence of intermittent demand across all categories confirmed that specialized modeling approaches would be necessary.

This analysis confirms the limitations of traditional forecasting methods when applied to real-world data characterized by intermittent demand patterns. To effectively address these challenges, a comprehensive preprocessing pipeline was implemented to address both general data quality issues and the specific challenges of intermittent demand.

The three datasets were merged with spark due to the huge amount of data (product information, calendar data, and sales records) and then split the resulting dataset by product category, I did this for validation (the file with 30 days more than the training set) and training dataset. This separation was motivated by the distinct characteristics and challenges observed in each category during exploratory analysis. After splitting, I addressed missing values in the price data (approximately 3.2% of records) using forward-fill imputation, which maintains the most recently known price until a new one is observed. This approach reflects the real-world scenario where prices typically remain stable until explicitly changed. While the sales data itself had no missing values, there were many legitimate zero values representing days without sales, which required special treatment as part of the intermittent demand modeling strategy.

Feature engineering proved to be essential to capture the complex patterns identified in the exploratory analysis. I created temporal features (day of week, month, holidays), lag features (prior 1, 7, 14, and 28 days' sales), rolling statistics (7, 14, and 28-day moving averages and standard deviations), and price-related features (current price, price changes, promotional indicators), the feature engineering is going to be explained better in Chapter 4. I also added dedicated event indicator features for major events like Easter and Super Bowl final, allowing the model to learn the specific impact of each event on different product categories. For the SNAP program effects that were particularly prominent in Wisconsin, specific indicator variables for SNAP days and their interaction with the state were added, allowing the model to capture the varying intensity of SNAP effects across different regions.

The high prevalence of intermittent demand patterns across all product categories presents significant challenges for traditional forecasting approaches. As demonstrated by the zero-sales analysis, more than half of the product-store combinations across all categories show zero sales on at least 60% of days. Such sparse and irregular demand patterns require specialized modeling techniques beyond standard time series methods.

While routine preprocessing methods such as feature engineering and normalization are required, these are not enough to deal with the particular problems created by intermittency. I will outline the particular methods used to address this issue in the next section, to deal with these intermittency issues, I used specialized methods instead of data cleaning. For highly sparse patterns of sales, employed a dual-model approach that models separately how likely non-zero demand is and how much is likely to be demanded should there be demand. This was more effective than usual forecasting methods for products with highly irregular sales patterns. The technicalities and performance assessments of this dual-model strategy will be shown in the subsequent chapter in which the full forecasting approach is presented.

Chapter 3

Forecasting Modeling and Implementation

3.1 Preliminary Model Selection and Evaluation

Before developing a complete forecasting solution for the entire dataset, I conducted a preliminary analysis to identify the best modeling approaches. Given the computational challenges presented by the full dataset (exceeding 58 million observations), this targeted evaluation allowed me to determine which algorithms would be most effective for the subsequent full-scale implementation.

For this preliminary assessment, fifteen SKUs were selected from the Hobbies category. This category was specifically chosen as it presented the most challenging forecast scenario, with 82.6% of products showing zero sales on more than 60% of days (as detailed in Section 2.3). These characteristics make Hobbies an ideal test case: If models perform well in these challenging patterns, they should also handle the comparatively more regular patterns in food and household categories effectively.

The SKU selection process employed a systematic classification approach that calculated several key metrics for each product:

- Zero percentage (proportion of days with zero sales)
- Coefficient of variation (standard deviation divided by mean)
- Weekly autocorrelation (correlation between sales with 7-day lag)
- Sales frequency (proportion of days with non-zero sales)

Based on these metrics, each SKU was classified into one of several types of demand pattern. The final selection included:

- 10 items with highly intermittent 'lumpy' demand (high zero percentage and high coefficient of variation)
- 4 items with more moderate 'regular' sales patterns (moderate zero percentage, more consistent patterns)
- 1 item with strong 'seasonal' demand influences (high weekly autocorrelation)

This selection ensured the evaluation would reflect the full forecasting challenges present in the dataset, while emphasizing the intermittent demand patterns that are most prevalent in retail environments.

For each selected SKU, I implemented and compared five distinct forecasting approaches: ARIMA, Prophet, LSTM, LightGBM Direct, and LightGBM Two-Stage (all models discussed in Section 1.3). During the preliminary tests I realized that a standard application of LightGBM (the Direct approach) might not be optimal for the challenging intermittent patterns prevalent in retail data. Inspired by techniques in demand planning literature, I decided to test this Two-Stage implementation that better reflects the practical retail reality. Instead of forcing a single model to predict both zeros and quantities simultaneously, Two-Stage approach first determines whether a product will sell at all on a given day, then, only if a sale is expected, estimates how much will be purchased. This dual-model strategy mirrors the actual retail purchasing decision process: first a customer decides to buy a product (or not), and only then decides how many units to purchase. This intuitive decomposition proved particularly effective for the highly intermittent patterns in the Hobbies category, where standard regression models often struggle to accurately represent the sparse sales punctuated by occasional large purchases.

Each model was assessed using complementary performance metrics that capture different aspects of forecast accuracy, and as target variable the sales contained in the validation dataset:

- **RMSE (Root Mean Square Error)**: The square root of the average of squared differences between predicted and actual values. RMSE gives higher weight to large errors and is particularly sensitive to outliers, making it appropriate for penalizing significant forecasting mistakes that could lead to stockouts.
- **MAE (Mean Absolute Error)**: The average of the absolute differences between predicted and actual values. MAE represents the average error in units of sales, providing a directly interpretable measure of prediction accuracy.
- **MAPE (Mean Absolute Percentage Error)**: The average of absolute percentage errors, expressed as a percentage relative to the actual values. MAPE facilitates

comparison across different SKUs with varying sales volumes, although it can be inflated when actual values are close to zero, a common situation with intermittent demand.

These complementary metrics provide a comprehensive assessment of the performance of the forecast, with lower values indicating a better precision on all three measures. Table 3.1 presents the average performance across all tested SKUs:

Table 3.1: Average Performance Comparison of Forecasting Models on Hobbies SKUs

Model	**RMSE**	**MAE**	**MAPE**
LightGBM_Two_Stage	0.93	0.52	61.6%
LightGBM_Direct	0.93	0.55	61.9%
LSTM	1.22	0.88	67.0%
Prophet	1.24	0.86	72.1%
ARIMA	1.39	0.91	82.2%

The results demonstrate that both LightGBM implementations significantly outperformed traditional time series models, with approximately 33% lower error rates compared to ARIMA. The neural network approach (LSTM) performed better than traditional statistical methods but still lagged behind the gradient boosting implementations.

This performance differential aligns with the theoretical strengths discussed in Section 1.3, but reveals additional insights specific to this specific context:

- **Zero-inflation handling**: The superior performance of LightGBM for Hobbies products confirms that its architecture is particularly well-suited for the highly intermittent demand patterns identified in Section 2.3, where it's possible to observed that 82.6% of Hobbies products showed zero sales on more than 60% of days.

- **Data efficiency**: Despite the limited historical data available for individual SKU-store combinations, gradient boosting models achieved reliable performance without overfitting, an important advantage over deep learning approaches like LSTM that typically require larger training datasets to generalize effectively, potentially requiring more SKUs and stronger hardware components.

- **Feature utilization**: The importance of temporal features (wm_yr_wk, rmean_14) and intermittency indicators (zero_pct) shown in Table 3.3 demonstrates how effectively LightGBM leverages the retail-specific features engineered based on the exploratory analysis in Chapter 2.

The nearly identical overall RMSE between the Two-Stage and Direct LightGBM approaches (both 0.93) warrants deeper investigation, as the aggregate metrics alone don't

reveal the pattern-specific advantages of each approach. Table 3.2 provides this breakdown by demand pattern type.

Table 3.2: RMSE by Demand Pattern Type and Forecasting Model

				LightGBM	
SKU Type	**ARIMA**	**Prophet**	**LSTM**	**Two-Stage**	**Direct**
Lumpy	1.5	1.3	1.3	**1.0**	1.0
Regular	1.0	0.9	0.9	**0.6**	0.6
Seasonal	1.8	1.9	1.8	1.7	**1.7**

This pattern-specific analysis revealed that:

- For highly intermittent 'lumpy' demand (which constitutes the majority of Hobbies SKUs), the Two-Stage approach showed a 2.5% improvement over the Direct approach and approximately 30% improvement over the next best non-LightGBM model
- For regular demand patterns, both LightGBM approaches performed similarly, with minimal difference (0.56%)
- For seasonal demand, the Direct approach they have a very similar performe as well.

To understand which variables have the greatest impact on the model, the concept of feature importance was used. Feature importance quantifies each variable's contribution to the model's predictive power. In simple terms, a feature is considered important if changing its values causes a significant change in the model's predictions. In tree-based models like LightGBM, feature importance is calculated by observing how frequently a variable is selected for splitting the data and how much these splits improve the overall accuracy of the model. Features with high importance are those that effectively distinguish between different demand levels, thus providing greater predictive power.

The features used in the models can be grouped into six intuitive categories, each capturing a different aspect of retail sales patterns:

- **Calendar-based features**: These capture how sales vary based on time. For example, `wm_yr_wk` identifies specific weeks of the year (like the week of Black Friday), `dayofweek` distinguishes between weekdays and weekends, and `month` captures seasonal patterns like holiday shopping in December.
- **Event indicators**: These flag special days that influence shopping behavior. The `event_name_1` feature identifies specific events (like Super Bowl, Valentine's Day, or Thanksgiving), while `event_type_1` categorizes events by type (sporting, cultural, or national holiday). These features help the model capture the unique impact of each event, which can vary dramatically as shown in Section 2.2.

- **SNAP disbursement features**: The original SNAP indicators from the dataset are preserved, allowing the model to learn the effects of benefit disbursement days on purchasing patterns.

- **Recent history features**: These look at what happened in the immediate past. For instance, `lag_1` answers "how many units did we sell yesterday?", `lag_7` shows "what were sales on this same day last week?", and `lag_14` reveals "what happened two weeks ago on this day?". These help the model recognize patterns like "sales typically spike the day after a promotion."

- **Trend indicators**: These smooth out daily fluctuations to reveal underlying patterns. The `rmean_7` feature shows the average sales over the last week, helping identify if a product is trending upward or downward. Similarly, `rmax_7` captures recent peaks, answering "what was the highest daily sales volume this past week?"

- **Purchasing frequency metrics**: These specifically address how often a product sells. The `zero_pct` feature answers "what percentage of days does this product not sell at all?", while `days_since_last_sale` tracks how long it's been since someone bought the item. These are important to distinguish between popular daily items and specialty products that sell occasionally.

- **Price factors**: These capture how pricing affects sales. The `sell_price` shows the current price, `price_lag_1` reveals if there was a recent price change, and `price_change` flags when prices have just been adjusted, helping identify price sensitivity and promotion effects.

These feature categories work together to model the complex dynamics of retail demand. For example, the model can learn that a product with high `zero_pct` suddenly sells well on `snap_WI` days with a temporary price reduction (`price_change`), particularly in the `FOODS` category, reflecting the real-world pattern observed in the exploratory analysis.

Table 3.3 presents the top 15 features ranked by importance:

Diverse insights emerge from this analysis:

- Temporal patterns (particularly week of year) exhibit the strongest predictive power, reflecting the strong weekly and annual cycles observed in Section 2.2

- Recent sales history metrics (rolling means and lags) provide essential context for predictions, outperforming simple calendar variables

- Intermittency indicators (zero_pct, days_since_last_sale) rank among the top predictors, confirming the importance of explicitly modeling zero-inflation patterns

Table 3.3: Top-15 Features by Importance in LightGBM for the 15 sample sku

Rank	Feature	Description
1	wm_yr_wk	Week-of-year calendar encoding
2	rmean_14	14-day rolling mean of sales
3	rmean_7	7-day rolling mean of sales
4	zero_pct	Percentage of zero-sales days in historical window
5	dayofweek	Day of week (0–6)
6	lag_1	Previous day's sales
7	month	Month of year (1–12)
8	lag_7	Sales from same day previous week
9	lag_14	Sales from two weeks prior
10	store_id	Store identifier (location effects)
11	rmax_7	Maximum sales in previous 7 days
12	days_since_last_sale	Count of days since non-zero demand
13	month_sin	Sinusoidal month encoding (seasonality)
14	rmax_14	Maximum sales in previous 14 days
15	day_cos	Cosine encoding of day-of-week cycles

These feature importance rankings also explain why the LightGBM models outperformed traditional time series approaches. Tree-based methods like LightGBM can automatically capture nonlinear relationships between features and learn complex interaction patterns, particularly important for features like zero_pct and days_since_last_sale that have threshold-like effects on demand prediction.

The preliminary analysis yielded two very important insights that guided subsequent implementation: first, the superior performance of gradient boosting methods over traditional approaches for intermittent demand; and second, the pattern-specific advantages of each LightGBM variant. While both Direct and Two-Stage approaches demonstrated similar overall performance, their effectiveness varied meaningfully across demand patterns, suggesting an opportunity for an adaptive framework that could leverage the strengths of each approach.

With these findings in hand, I proceeded to develop a full-scale implementation that could process the entire 58-million-observation dataset while intelligently selecting the best modeling approach based on each product-store combination's characteristics.

3.2 Complete Forecasting Framework Development

Based on the results of the preliminary analysis conducted in Section 3.1, I choose LightGBM as model and develop a complete adaptive forecasting framework capable of scaling to the entire Walmart dataset divided by category. This implementation leveraged

both LightGBM variants within a unified system that automatically selected the appropriate approach based on demand pattern characteristics.

The core of the implementation is an adaptive model selection system that analyzes each time series using three key metrics derived from the exploratory analysis: zero percentage (proportion of days without sales), coefficient of variation (standard deviation divided by mean), and weekly autocorrelation (correlation between values separated by 7 days). Based on these metrics, each product-store combination is classified into one of four patterns:

- 'Lumpy' pattern (zero_percentage > 0.7 & CV > 1.0): Two-Stage approach
- 'Seasonal' pattern (weekly_autocorr > 0.5): Direct approach
- 'Regular' pattern (zero_percentage < 0.4): Direct approach
- 'Intermittent' pattern (other cases): Two-Stage approach

Here a graphic representation of the framework's pipeline:

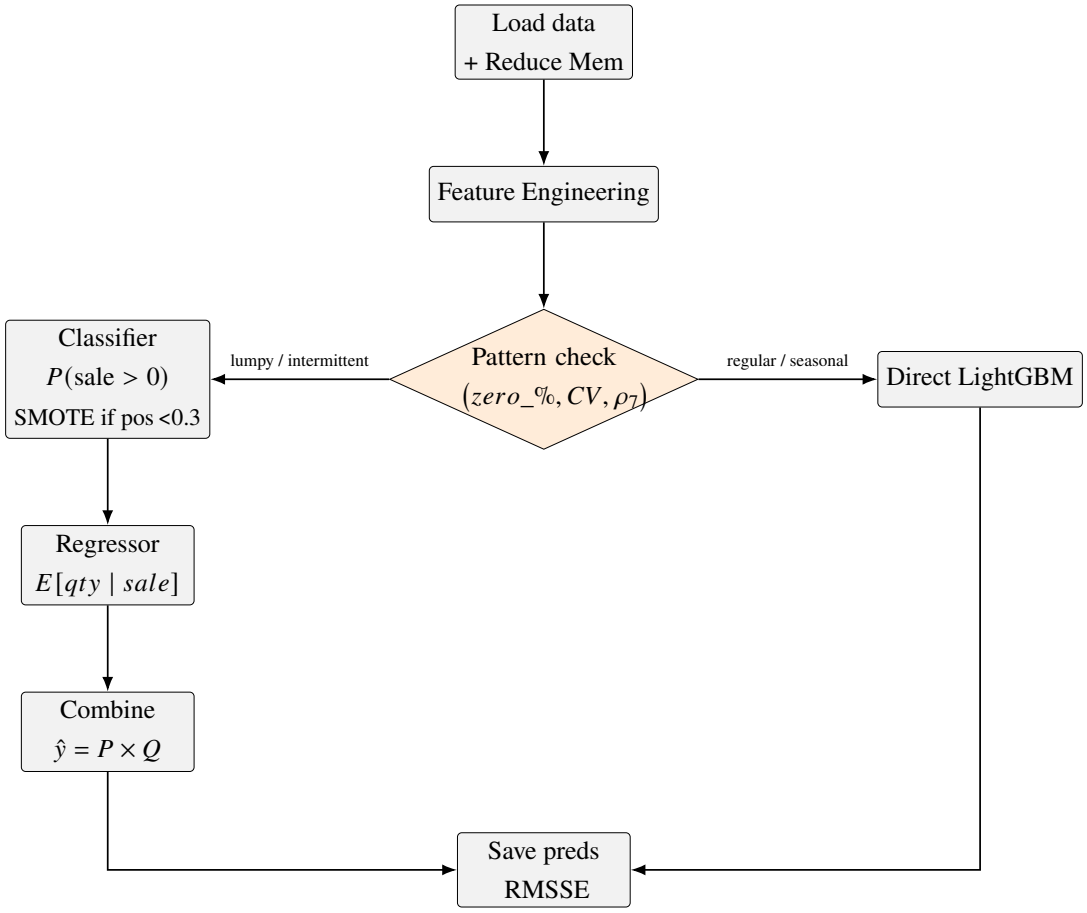

Figure 3.1: Two-Stage adaptive forecasting pipeline

To evaluate the forecasting accuracy of the framework in retail contexts, Root Mean Squared Scaled Error (RMSSE) metric was selected , as it effectively normalizes the

forecast errors relative to the inherent historical variability of each individual time series. The RMSSE formula is defined as follows:

$$\text{RMSSE} = \frac{\sqrt{\frac{1}{h}\sum_{t=n+1}^{n+h}(y_t - \hat{y}_t)^2}}{\sqrt{\frac{1}{n-1}\sum_{t=2}^{n}(y_t - y_{t-1})^2}}$$

where y_t denotes the actual sales at time t, $\hat{y}_t$ is the forecasted sales at time t, n is the number of historical observations, and h represents the forecast horizon.

The denominator of this equation represents the root mean squared error of a naïve forecasting model, which simply predicts that future sales will be identical to sales from the preceding period. Consequently, the RMSSE provides a relative measure of forecast accuracy that accounts for each series' historical volatility. This normalization is important, as it allows for meaningful performance comparisons across Stock Keeping Units (SKUs) that differ greatly in sales volume and volatility. Metrics like the RMSE fail to capture these contextual differences, as they measure absolute prediction errors without reference to historical variability.

Implementing this framework on such a large dataset required specific technical optimizations. The memory management strategy was essential, involving a systematic approach to data type optimization. For numeric columns, integers were downcasted to the minimum required bit depth (8, 16, or 32-bit) based on their range of values, while floating-point values were standardized to 32-bit precision instead of the default 64-bit. This methodical optimization reduced the memory footprint by approximately 21.7% across all three categories, allowing the entire dataset to be processed without specialized high-memory hardware.

The feature engineering process was optimized for both computational efficiency and predictive power. Rather than generating all possible time-based features, the implementation focused on the most predictive features identified in Section 3.1. Temporal features included week-of-year encoding (wm_yr_wk), day of week (dayofweek), and month, complemented by cyclic encodings (month_sin, day_cos) to capture seasonal patterns. Lag features were selectively created for 1, 7, and 14-day offsets, while rolling statistics (rmean_7, rmean_14, rmax_7) captured recent trends. Intermittency indicators (zero_pct, days_since_last_sale) were specifically designed to address the challenges of sparse sales patterns. Price-related features (price_lag_1, price_change) completed the feature set, capturing price sensitivity effects. This selective approach significantly reduced computation time while maintaining predictive power.

Developed a framework with a store-level checkpoint system that saved partial results, allowing for recovery and resumption in case of processing interruptions. This proved valuable when handling the multi-hour processing required for the complete dataset across

all stores.

As thoroughly discussed in Section 2.3, the Walmart dataset presents significant challenges due to highly intermittent demand patterns, characterized by extensive periods with zero sales interspersed by occasional spikes in demand. To reliably address this specific issue, I implemented a specialized forecasting approach known as the Two-Stage model. This methodology explicitly accounts for the intermittency by decomposing the forecasting task into two sequential phases:

1. **Classification Stage**: Predicting the probability of whether a product will sell on a given day.

2. **Regression Stage**: Predicting the expected quantity sold, conditioned on a sale occurring.

The classifier learns from all historical data to identify when sales occur, while the regressor is trained only on positive-sales instances to estimate purchase quantities.

Synthetic Minority Over-sampling Technique (SMOTE) is a technique for addressing highly unbalanced datasets by synthesizing examples of the minority class. implemented using the Python library imbalanced-learn (Imbalanced-learn developers 2023), this algorithm came in handy while handling the extreme class imbalance problem that is native to intermittent demand forecasting. For retail settings, this imbalance is evidenced by datasets in which the overwhelming majority of the days record zero sales (majority class), thereby posing a major hurdle for machine learning models. While analyzing Walmart data, I noted that standard classification models were highly "majority class biased," predicting "zero sales" for almost every day in cases where those models were trained on highly imbalanced data, essentially neglecting the sparse but commercially valuable sales occurrences.

Given the severe class imbalance in the classification phase (as zero-sales days vastly outnumber days with positive sales), I employed SMOTE to balance the training data. SMOTE in the classification part generates synthetic minority class instances by interpolating between existing minority examples, effectively balancing the dataset and enhancing the model's predictive capabilities for rare but commercially significant sales events.

Specifically, SMOTE creates synthetic data points according to the following formulation:

$$x_{\text{new}} = x_i + \alpha \cdot (x_{\text{nn}} - x_i)$$

where x_i is an existing minority class instance, x_{nn} represents one of its nearest neighbors

in feature space, and α is a random value uniformly drawn between 0 and 1, determining the interpolation point between these instances.

Through this targeted approach, the Two-Stage model with SMOTE effectively captures the intermittent nature of product demand, significantly outperforming standard methods that fail to explicitly model this aspect.

The implementation includes error handling that detected missing values and automatically fell back to the original dataset rather than failing, ensuring uninterrupted processing even with imperfect data. Array dimension mismatches a common issue in time series processing due to lagged features were resolved by implementing comprehensive date tracking that maintained alignment between features, predictions, and actual values throughout the pipeline.

Categorical variable handling required special consideration due to LightGBM's requirement for numeric input. A two-step approach was employed: first, categorical variables were detected automatically based on their data type and cardinality; second, they were either encoded using label encoding (for low-cardinality variables) or removed (for high-cardinality variables with limited predictive value). This approach prevented information leakage while maximizing the usable signal from categorical predictors.

To address class imbalance (typical of products with many zero-sales days), the implementation utilizes the balancing technique described earlier, with error handling that gracefully falls back to the original dataset when needed. The final forecast is calculated by multiplying the probability of a sale by the expected conditional quantity:

$$\hat{y}_{\text{final}} \;=\; P\big(\text{sales} > 0\big) \;\times\; E\Big[\text{sales} \,\Big|\, \text{sales} > 0\Big] \tag{3.1}$$

This decomposition mirrors the actual decision process of a customer: *first* the binary choice to purchase, *then* the quantity decision. By taking the product of these two components, the model outputs the *unconditional* expected demand for the day. If either the purchase probability is low or the conditional quantity spikes, their combined effect is naturally tempered, yielding a smooth and interpretable forecast that scales well across items with vastly different sparsity levels.

After initial implementation, I discovered that each product category required specific optimizations to address their unique characteristics:

For FOODS, with higher price sensitivity (3.5%) and strong responses to events like SNAP disbursements, I implemented additional price-related features and adjusted the SMOTE threshold to 0.6 to accommodate its higher baseline of non-zero sales days (ranging from 35% to 47%). Some stores with particularly balanced classes (>42% sales days) encountered minority class sampling errors, requiring fallback to unbalanced training.

For HOBBIES, which exhibited extreme intermittency with all stores classified as "lumpy" (highly sparse sales patterns), I initially attempted standard SMOTE implementation with SMOTE threshold of 0.3. The imputation strategy proved successful for HOBBIES stores, enabling SMOTE application for all "lumpy" pattern stores.

The HOUSEHOLD category required an intermediate SMOTE threshold of 0.3, reflecting its position between the sparse HOBBIES patterns and the more regular FOODS patterns. The imputation strategy proved successful for HOUSEHOLD stores, enabling SMOTE application for all "lumpy" pattern stores.

The SMOTE implementation required category-level modifications. For HOBBIES, the algorithm often struggled with missing values, whereas with some FOODS stores with better balanced classes (>42% sales days), it encountered minority class sampling errors. A robust error-measure handling system, with imputation in advance and graceful resorting to the original dataset if required, solved the problem. This approach proved particularly successful for HOUSEHOLD stores, where the imputation strategy enabled successful SMOTE application for all "lumpy" pattern stores. Processing time also varied dramatically between categories, with FOODS stores requiring substantially longer computation (up to 6,000 seconds per store) compared to HOBBIES (approximately 90-120 seconds per store), reflecting the computational complexity of handling larger sales volumes and different class distributions. Array dimension mismatches between predictions and actual values were resolved by implementing comprehensive date tracking throughout the pipeline, ensuring consistent alignment regardless of the category being processed.

The adaptive framework was applied to all three product categories (FOODS, HOBBIES, and HOUSEHOLD) across all 10 stores. Analysis of demand patterns revealed interesting category-specific characteristics, as detailed in Table 3.4.

Table 3.4: Distribution of Demand Patterns Across Product Categories

Pattern Type	**FOODS**		**HOBBIES**		**HOUSEHOLD**	
	Count	**%**	**Count**	**%**	**Count**	**%**
Lumpy	0	0%	10	100%	6	60%
Regular	0	0%	0	0%	0	0%
Seasonal	0	0%	0	0%	0	0%
Intermittent	10	100%	0	0%	4	40%

The pattern distribution reveals a clear distinction between product categories. For the HOBBIES category, all 10 stores were classified with 'lumpy' demand patterns, confirming the high intermittency observed in the exploratory analysis. The HOUSEHOLD category

showed a more diverse distribution, with 60% of stores exhibiting 'lumpy' patterns and 40% classified as 'intermittent.' In contrast, the FOODS category displayed a consistent 'intermittent' pattern across all stores, indicating a less extreme but still challenging forecasting scenario compared to the other categories.

The performance of the models for each category showed different characteristics, as detailed in Tables 3.5, 3.6, and 3.7.

Table 3.5: Performance Metrics for FOODS Stores

Store	**RMSSE**	**RMSE**	**MAE**	**MAPE**	**Pattern**
CA_1	0.7994	2.6906	1.3220	99.10%	Intermittent
CA_2	0.9104	2.3966	1.2348	92.90%	Intermittent
CA_3	0.6735	2.9160	1.3157	80.09%	Intermittent
CA_4	0.7602	1.5393	0.7414	78.56%	Intermittent
TX_1	0.7136	2.0010	0.8160	80.68%	Intermittent
TX_2	0.7246	2.3997	1.0173	88.99%	Intermittent
TX_3	0.9660	2.8388	1.1998	98.75%	Intermittent
WI_1	0.9304	2.1761	1.1886	96.19%	Intermittent
WI_2	0.9647	3.6343	1.5547	102.85%	Intermittent
WI_3	0.8650	2.9070	1.2525	102.87%	Intermittent
Average	**0.8308**	**2.5499**	**1.1643**	**92.10%**	

Table 3.6: Performance Metrics for HOBBIES Stores

Store	**RMSSE**	**RMSE**	**MAE**	**MAPE**	**Pattern**
CA_1	0.6530	2.0260	0.6664	82.28%	Lumpy
CA_2	0.6144	1.4871	0.4785	64.40%	Lumpy
CA_3	0.6592	2.1110	0.6819	74.91%	Lumpy
CA_4	0.6631	1.4118	0.4158	62.12%	Lumpy
TX_1	0.6845	1.2308	0.3484	65.63%	Lumpy
TX_2	0.6262	1.3051	0.4342	60.07%	Lumpy
TX_3	0.7908	1.4331	0.4470	65.13%	Lumpy
WI_1	0.6341	1.3496	0.4537	66.17%	Lumpy
WI_2	0.7075	1.1192	0.3318	56.59%	Lumpy
WI_3	0.6595	1.1601	0.3284	63.08%	Lumpy
Average	**0.6692**	**1.4634**	**0.4586**	**66.04%**	

Table 3.7: Performance Metrics for HOUSEHOLD Stores

Store	RMSSE	RMSE	MAE	MAPE	Pattern
CA_1	0.7819	1.1596	0.5125	65.22%	Intermittent
CA_2	0.7805	1.2961	0.6259	71.23%	Intermittent
CA_3	0.7934	1.9170	0.8427	70.80%	Intermittent
CA_4	0.7091	0.7111	0.2845	50.20%	Lumpy
TX_1	0.7531	1.2148	0.4945	67.12%	Lumpy
TX_2	1.0691	1.8078	0.5150	62.78%	Intermittent
TX_3	0.7819	1.2004	0.4933	63.48%	Lumpy
WI_1	0.7206	0.9023	0.3745	54.23%	Lumpy
WI_2	0.9875	1.6862	0.5530	61.99%	Lumpy
WI_3	0.7344	1.0464	0.4015	59.68%	Lumpy
Average	**0.8112**	**1.2942**	**0.5093**	**62.67%**	

Analyzing forecasting metrics across the three product categories reveals important insights into the relationship between demand patterns and prediction accuracy. At first glance, RMSE values suggest significant performance differences: FOODS (2.55) shows notably higher errors compared to HOBBIES (1.46) and HOUSEHOLD (1.29). However, this interpretation is misleading as it fails to consider intrinsic category-specific differences in sales volumes and volatility.

The normalized RMSSE metric provides a more accurate picture by accounting for historical variability. FOODS achieves an RMSSE of 0.83, closely aligned with HOUSEHOLD at 0.81 and competitively positioned against HOBBIES at 0.67. This shows that despite higher absolute errors, FOODS forecasts are comparable when scaled against inherent sales volatility.

Several key factors explain these results:

- **Volume and Variability Differences:** FOODS products have higher sales volumes and more pronounced daily fluctuations, naturally producing larger absolute errors. Normalizing these errors with RMSSE clearly demonstrates comparable forecasting performance across categories. For example, a forecasting error of two units in a high-volume FOODS product (typical daily sales ranging 0–10 units) is proportionally less significant than a 0.5-unit error in a lower-volume HOBBIES product (typical daily sales of 0–2 units).
- **Distinct Demand Patterns:** Each category exhibited unique demand characteristics: HOBBIES were exclusively categorized as “lumpy” (100%), FOODS as consistently “intermittent” (100%), while HOUSEHOLD displayed mixed patterns (60% lumpy,

40% intermittent). This confirms the effectiveness of both the classification system and the tailored forecasting strategies applied.

- **Adaptive Sampling Strategies with SMOTE:** The correct use of SMOTE required fine-tuning across categories. HOUSEHOLD required a moderate threshold (0.3), whereas FOODS needed a higher threshold (0.6) due to less extreme class imbalances (35–47% non-zero days). Furthermore, for particularly balanced FOODS subsets (>42%), the model intelligently omitted SMOTE, showcasing adaptive flexibility.

- **Category-Specific Price Sensitivity:** FOODS exhibited greater price sensitivity (3.5% sales reduction per 1% price increase) compared to HOUSEHOLD (2.1%) and HOBBIES (1.7%). This heightened responsiveness demanded careful inclusion of price features in the forecasting model.

- **Differential External Factor Responsiveness:** FOODS demonstrated significant sensitivity to external events, notably SNAP disbursement days (up to 30% sales uplift in Wisconsin stores) and seasonal promotions. This external responsiveness necessitated nuanced forecasting models, successfully addressed by the adopted Two-Stage approach.

Despite pronounced differences in demand characteristics, the adaptive forecasting framework achieved consistent scaled accuracy (RMSSE between 0.67–0.83) across all categories. This uniformity highlights the system's ability to tailor forecasting methodologies according to distinct product characteristics, removing the necessity for entirely separate category-specific approaches. The operational implications of these performance results will be explored further in Section 4.2.

The feature importance analysis presented in Table 3.8 underscores distinctions between the classification and regression components of the Two-Stage forecasting model. In retail demand forecasting, classification and regression mirror sequential consumer decision processes.

The classification stage predicts the occurrence of a transaction, modeling the probability $P(sales > 0)$. This binary task addresses the significant challenge of zero-inflation observed in the exploratory data analysis (Section 2.3).

The regression stage, conversely, estimates the expected quantity sold, conditional on a sale occurring, i.e., $E[sales|sales > 0]$. By focusing solely on positive-sales events, the regression model effectively isolates and models the quantity-purchase behavior, avoiding biases from zero-inflated data.

Distinctive feature importance patterns validate this Two-Stage decomposition. Classification models heavily relied on the intermittency metric (`zero_pct`), consistently ranking

first with importance significantly higher than other features. Conversely, regression models emphasized recent sales features (`lag_1`, `rmean_7`), particularly pronounced in FOODS, illustrating the importance of recent purchasing behavior.

Price-related features consistently ranked high in classification (top-three across categories) but were lower in regression rankings, aligning with practical retail intuition that price changes primarily influence purchase decisions rather than quantity purchased.

Additionally, SNAP-related variables exhibited markedly higher importance for FOODS, consistent with observed SNAP-dependency in the category. Event-related features also mirrored exploratory findings (Section 2.2), affirming the distinct category-specific reactions to external factors.

Table 3.8: Top 20 Feature Importance Comparison Across Categories and Model Types

	FOODS		HOUSEHOLD		HOBBIES	
Rank	**Feature (Clf)**	**Imp.**	**Feature (Clf)**	**Imp.**	**Feature (Clf)**	**Imp.**
1	zero_pct	6219.8	zero_pct	4355.8	zero_pct	3847.3
2	price_norm.	4336.7	price_norm.	3477.6	sell_price	2225.2
3	sell_price	4014.7	sell_price	2967.2	rmean_14	1015.7
4	rmean_14	3492.3	rmean_14	2002.0	price_lag_1	417.1
5	rmean_7	1965.6	days_since_last	461.1	days_since_last	348.4
6	lag_1	1575.8	wm_yr_wk	447.5	wm_yr_wk	324.4
7	rmax_7	770.2	price_lag_1	301.9	month	186.6
8	days_since_last	550.7	lag_1	262.5	rmean_7	164.7
9	wm_yr_wk	531.5	rmean_7	255.4	lag_1	145.2
10	price_lag_1	321.3	month	202.5	dayofweek	86.9
11	is_weekend	313.2	dayofweek	173.4	price_change	41.8
12	month	157.6	rmax_7	156.0	day_cos	27.2
13	dayofweek	131.2	month_sin	88.8	snap_CA	13.6
14	snap_WI	116.1	price_change	54.9	month_sin	9.6
15	price_change	78.0	snap_TX	22.8	lag_14	9.5
16	month_sin	27.9	is_weekend	15.3	rmax_7	9.0
17	lag_14	24.0	lag_14	15.0	lag_7	5.8
18	lag_7	16.8	snap_CA	12.4	snap_WI	5.6
19	snap_CA	10.1	lag_7	11.7	snap_TX	4.0
20	snap_TX	8.1	day_cos	6.2	has_event	2.2
	Feature (Regr.)		**Feature (Regr.)**		**Feature (Regr.)**	
1	lag_1	2473.7	rmean_14	1376.2	rmean_14	826.9
2	rmax_7	1842.9	rmean_7	1182.3	wm_yr_wk	679.5
3	rmean_7	1617.7	wm_yr_wk	1119.0	zero_pct	644.0
4	lag_7	1547.8	lag_1	1056.5	rmean_7	594.3
5	lag_14	1483.9	zero_pct	906.2	lag_14	501.1
6	rmean_14	1420.5	rmax_7	814.0	rmax_7	444.1
7	wm_yr_wk	1358.7	dayofweek	782.8	lag_7	427.1
8	zero_pct	1095.0	lag_14	699.4	lag_1	371.4
9	sell_price	1046.7	lag_7	659.6	price_lag_1	364.0
10	price_norm.	827.3	price_norm.	537.9	dayofweek	254.1
11	dayofweek	805.0	price_lag_1	399.3	sell_price	222.0
12	month	583.3	month_sin	311.5	month	187.8
13	month_sin	432.1	month	284.8	day_cos	179.4
14	is_weekend	414.7	day_cos	203.1	month_sin	109.2
15	day_cos	367.4	sell_price	181.6	snap_TX	57.7
16	price_lag_1	306.3	snap_CA	140.2	snap_CA	54.6
17	snap_CA	293.1	snap_TX	114.1	snap_WI	48.8
18	snap_TX	273.9	has_event	101.0	has_event	29.6
19	snap_WI	267.6	snap_WI	98.4	price_change	4.4
20	has_event	122.6	is_weekend	28.1	days_since_last	0.0

Overall, these insights validate the Two-Stage model's capability to effectively separate and model distinct decision-making processes inherent in retail purchasing behavior. Note that the model was executed store-by-store, naturally incorporating store-specific effects, which would have emerged prominently had "store" been modeled explicitly as a feature.

3.3 Forecasting Demand for New Products via Similarity-Based Approaches

During the development of the forecasting model, was encountered a question: how would the system handle the introduction of a new item with a unique SKU, absent from the historical dataset? This scenario, commonly known as the "cold start" problem, is prevalent in dynamic retail environments like Walmart's, where new products are frequently introduced without prior sales data.

To address this challenge, similarity-based approaches were explored. The core idea is to identify existing products with attributes similar to the new item and to use their historical sales data to predict the demand for the newcomer. This method relies on the assumption that products that share similar characteristics will exhibit comparable demand patterns.

The similarity between products can be formalized mathematically through several metrics. For feature vectors $\vec{A}$ and $\vec{B}$ representing an existing product and a new product respectively, common metrics include:

$$\text{sim}_{\cos}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}|| \cdot ||\vec{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.2}$$

$$\text{dist}_{\text{eucl}}(\vec{A}, \vec{B}) = ||\vec{A} - \vec{B}|| = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{3.3}$$

Calculating similarity between products involves analyzing various features:

- **Categorical Attributes:** Product category, subcategory, brand, and store type.
- **Numerical Attributes:** Price, dimensions, and weight.
- **Textual Descriptions:** Product titles and descriptions.
- **Visual Features:** Product images.

To quantify similarity:

- **Categorical and Numerical Data:** Techniques like one-hot encoding for categorical variables and normalization for numerical variables are applied, followed by distance metrics such as Euclidean or cosine similarity.

- **Textual Data:** Natural Language Processing (NLP) methods, including word embeddings or fuzzy matching, are used to convert text into numerical vectors, enabling similarity computation.

- **Visual Data:** Convolutional Neural Networks (CNNs) can extract feature embeddings from images, which are then compared using similarity measures.

Advanced models may combine these features into a unified embedding space, allowing for a comprehensive similarity assessment.

Once similar products are identified, their historical sales data can inform the demand forecast for the new product. Techniques such as weighted averaging, where weights are based on similarity scores, can be employed. For a new product p_{new}, if we identify k most similar products $\{p_1, p_2, ..., p_k\}$ with similarity weights w_i, the forecast can be calculated as:

$$\text{Forecast}(p_{new}) = \sum_{i=1}^{k} w_i \cdot \text{Historical\_Sales}(p_i) \tag{3.4}$$

In this way machine learning models trained on similar products' data can be adapted to predict the new item's demand.

Several studies and industry applications support this approach:

- Amazon Forecast uses item metadata to identify similar products, enhancing forecast accuracy for new product introductions. Their approach has led to forecasts that are up to 45% more accurate for products with no historical data (Amazon Web Services 2023).

- Impact Analytics developed AI-driven frameworks combining machine learning with domain expertise to tackle cold start challenges in retail demand forecasting. Their client research indicates demand forecasts for new products are between 25-30% more accurate than former judgment-based forecasts (Impact Analytics 2023).

Although this approach shows significant promise for addressing the cold start problem in retail forecasting, its implementation in the context of this study faces several practical limitations. The Walmart M5 dataset lacks most of the essential data types identified earlier, its have only basic category information without detailed subcategories like brands, information on product dimensions or weight, no access to textual descriptions or visual

product images. These data limitations significantly constrain the ability to calculate similarity metrics between products. Computational efficiency would pose a substantial challenge when dealing with large catalogs like Walmart's, requiring specialized hardware resources not available for this research. Despite these implementation barriers, the theoretical framework remains valuable for understanding how retailers could address the cold start problem in production environments where richer product metadata is available.

Incorporating these similarity-based methodologies into demand forecasting models not only addresses the cold start problem but also contributes to more efficient inventory management and supply chain optimization. For large enterprises like Walmart, which regularly introduce new products, this approach enables the prediction of sales performance for new items based on similarities with existing products, ultimately supporting more effective inventory decisions. While implementation was not feasible within the current research constraints, this conceptual framework provides a foundation for future extensions discussed in Section 5.2. After developing forecasting models in this chapters, Chapter 4 focuses on translating these forecasts into concrete operational decisions through the APS system.

Chapter 4

Operational Integration and Results

4.1 Translating Forecasts into Inventory Decisions

The translation of machine learning forecasts into operational decisions marks the point where theoretical research generates tangible business impact. This section presents the development of a technological artifact, aligned with the Design Science Research methodology described by (Baskerville et al. 2018), which encapsulates the design knowledge built in the previous chapters.

Inventory management requires converting statistical forecasts into actionable purchasing and stocking parameters. This research uses the improved accuracy of machine learning models to dynamically optimize inventory decisions.

To operationalize this, I developed an Advanced Planning and Scheduling (APS) system that serves as a **digital twin** for inventory management. In this context, a digital twin represents a real-time virtual replica of the inventory system that continuously simulates demand patterns, stock levels, and replenishment decisions. This digital twin enables decision-makers to:

- Test multiple scenarios instantly without affecting physical inventory
- Visualize the immediate impact of different parameter changes
- Understand financial implications before implementing decisions
- Integrate machine learning predictions with operational constraints

Figure 4.1 illustrates the high-level workflow of the APS digital twin, where machine learning forecasts and demand pattern metrics are combined with user-selected parameters to generate actionable inventory decisions and visual insights.

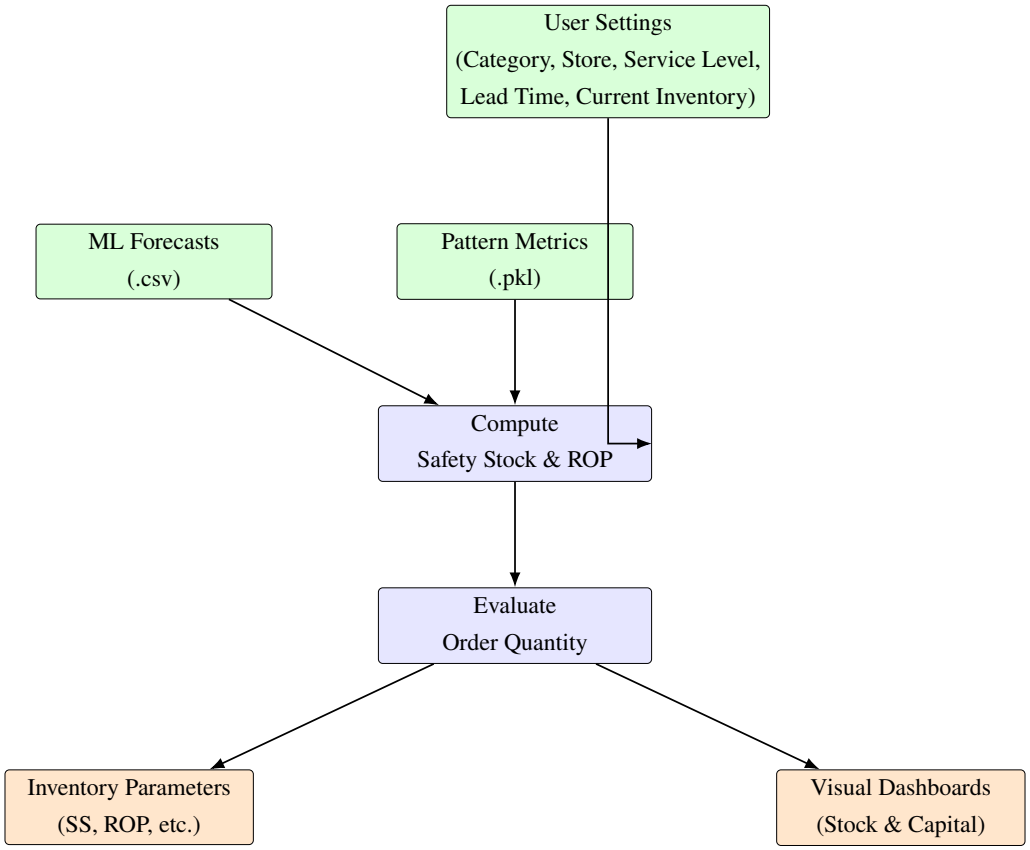

Figure 4.1: APS flow: inputs, core calculations, and resulting insights.

This digital twin provides dynamic, real-time simulation capabilities that translate complex ML forecasts into concrete operational decisions. The APS system automatically loads the relevant ML forecasts and pattern data for each store-category combination, ensuring inventory parameters are based on the most accurate and context-specific predictions available.

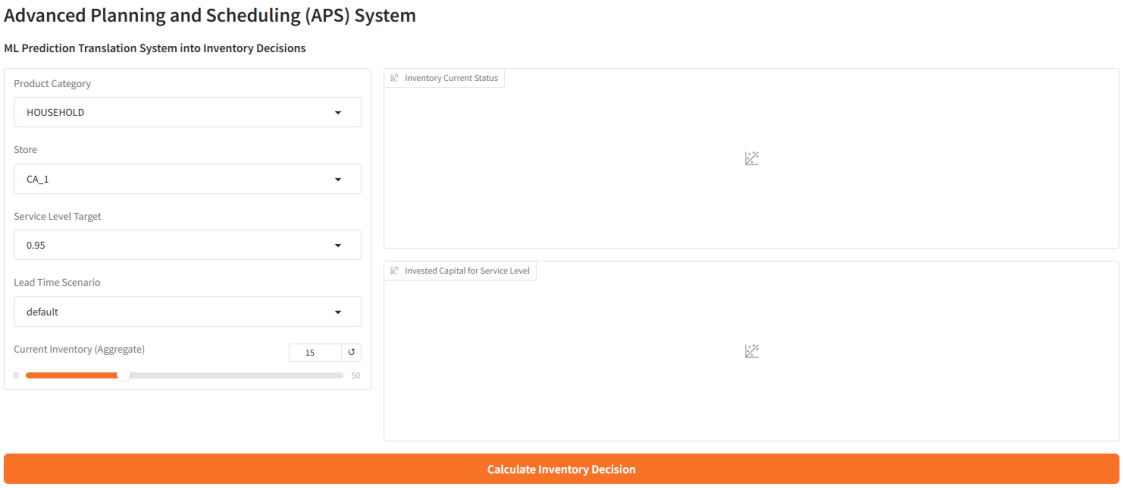

Figure 4.2: Advanced Planning and Scheduling (APS) System Interface showing the input parameters panel

The system operates with predefined operational parameters that reflect typical retail industry values:[1]

Lead Time Scenarios by Category:

- **FOODS:** 3 days (minimum), 4 days (default), 5 days (maximum) - reflecting perishable nature and high turnover
- **HOBBIES:** 5 days (minimum), 8 days (default), 12 days (maximum) - accounting for more complex supply chains
- **HOUSEHOLD:** 5 days (minimum), 8 days (default), 12 days (maximum) - standard replenishment cycles

Average Unit Costs by Category:

- **FOODS:** €5.50 per unit (acquisition and holding cost)
- **HOBBIES:** €12.75 per unit (higher-value specialty items)
- **HOUSEHOLD:** €8.25 per unit (standard consumer products)

The APS interface, developed using Gradio, provides an intuitive environment where retail managers can:

- Select the product category (FOODS, HOBBIES, or HOUSEHOLD) from a dropdown menu
- Choose a specific store location (CA_1, TX_2, etc.) to view location-specific data
- Set the desired service level target (90%, 95%, or 99%) to balance availability against inventory cost
- Specify the lead time scenario (minimum, default, or maximum) to account for supply chain variability
- Input the current inventory level using a slider for immediate visual feedback

The system implements inventory optimization formulas based on traditional safety stock and reorder point concepts, with several key innovations. The $pattern\_factor$ is a central innovation that adjusts safety stock calculations based on the demand pattern identified by the ML models:

[1]These parameters represent typical industry estimates used for testing and demonstrating the framework's capabilities. In production environments, these would be replaced with actual supplier lead times and negotiated costs. A detailed discussion of these assumptions and their limitations is provided in Section 5.1.

- **LUMPY patterns (factor = 1.2):** Demand shows extreme intermittency with long periods of zero sales followed by sudden, unpredictable spikes. The 20% safety stock increase accounts for this high volatility.
- **INTERMITTENT patterns (factor = 1.0):** Demand shows moderate irregularity with occasional zero-sales periods but more consistent purchasing behavior. This represents the baseline pattern requiring no adjustment.
- **REGULAR patterns (factor = 0.9):** Demand is relatively predictable with minimal zero-sales days and consistent purchasing behavior. The 10% safety stock reduction reflects the lower uncertainty inherent in these patterns.

With this adaptive pattern factor in place, the complete inventory optimization formulas are:

$$SS = z \cdot \sigma_{ML} \cdot \sqrt{L} \cdot pattern\_factor \tag{4.1}$$

$$ROP = \mu_{ML} \cdot L + SS \tag{4.2}$$

The ordering logic implemented in the system works as follows:

1. The system continuously monitors current inventory levels
2. When current inventory falls to or below the ROP, it triggers an ordering decision
3. Upon triggering, the system calculates an "order-up-to" level using the formula:

$$\text{Order Up To Level} = ROP + \mu_{ML} \cdot L + \mu_{ML} \cdot L \cdot 0.5 \tag{4.3}$$

$$\text{Order Quantity} = \max(0, \text{Order Up To Level} - \text{Current Inventory}) \tag{4.4}$$

This enhanced ordering approach ensures that when inventory drops below the reorder point, the system orders sufficient quantity to bring the inventory to a level that includes:

- The Reorder Point (ROP) threshold
- Expected demand during the lead time ($\mu_{ML} \cdot L$)
- An additional 50% buffer to cover demand uncertainty beyond ROP ($\mu_{ML} \cdot L \cdot 0.5$)(buffer of 0.5 approximately 4 days of additional coverage for an 8-day lead time)

This approach guarantees that when the order arrives, inventory will be well above the ROP, preventing immediate reordering and providing adequate coverage for sustained operations. Let's see this in practice with a realistic example.

Illustrative example of what happen in the code – HOUSEHOLD category, store `CA_1`

- Current inventory $I_0 = 15$ units (below the ROP, so replenishment is triggered)
- Lead time $L = 8$ days (default for HOUSEHOLD)
- ML prediction: $\mu_{ML} \approx 1.11$ units/day
- Safety stock $SS \approx 7$ units
- Re-order point $ROP = \mu_{ML}\, L + SS = 1.11 \times 8 + 8 \approx 17$ units
- Order-up-to level
  $OUTL = ROP + \mu_{ML}\, L + 0.5\, \mu_{ML}\, L \;=\; 17 + (1.11 \times 8) + (0.5 \times 1.11 \times 8) \approx 30.3$ units
- Quantity to order $Q = OUTL - I_0 = 30.3 - 15 = 15.3$ units

When the shipment is received, on-hand stock will rise to ≈ 30.3 units, providing about $30.3/1.11 \approx 27$ days of coverage and keeping inventory comfortably above the reorder point.

2

[2] *All numerical values in Fig. 4.3 are generated automatically by the APS prototype using the actual forecast file for store* `CA_1`.

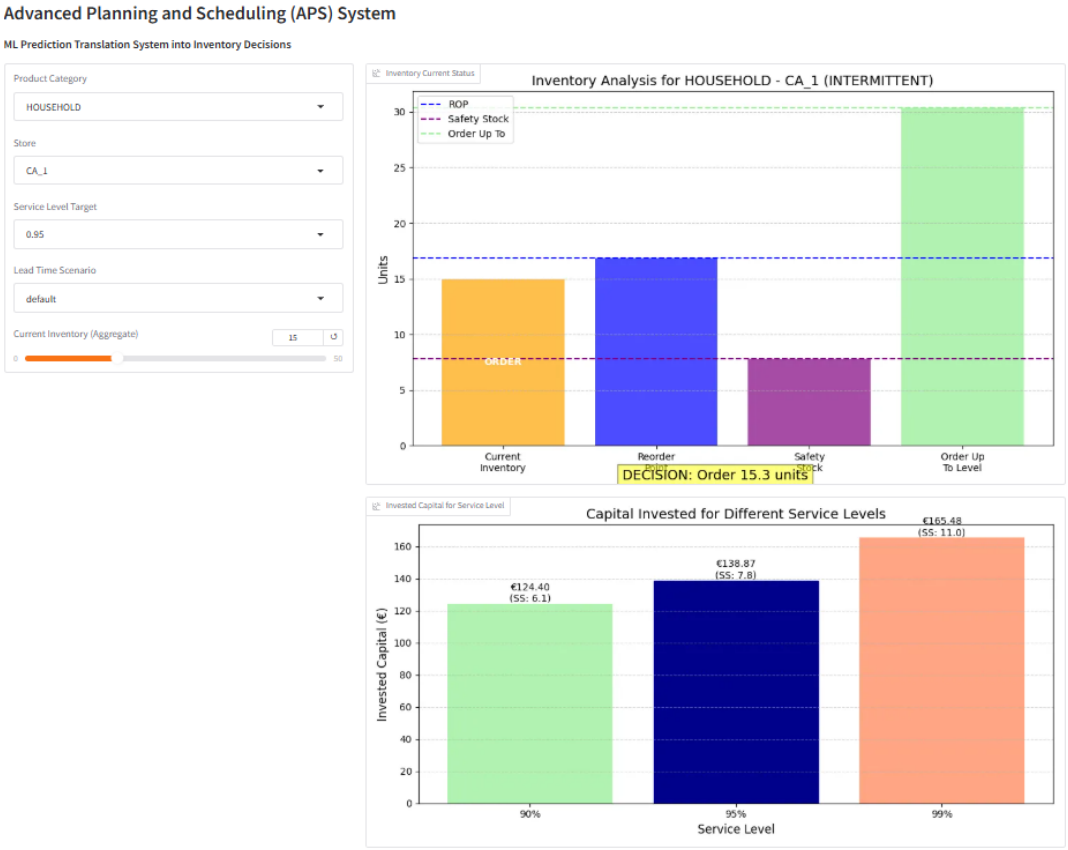

Figure 4.3: APS System analysis for HOUSEHOLD category at CA_1 store

As illustrated in Figure 4.3, the system provides a comprehensive analysis for the HOUSEHOLD category at store CA_1, where the ML models have identified an "INTERMITTENT" demand pattern. This exemplifies how the digital twin adapts its recommendations to different pattern characteristics and provides clear visual guidance for inventory decisions.

The upper visualization displays the critical inventory status:

- Current inventory stands at 15 units (orange bar), indicating inventory has fallen below the reorder point
- Reorder Point is calculated at approximately 17 units (blue dashed line), representing the threshold for initiating replenishment
- Safety Stock level is set at approximately 7 units (purple dashed line), providing a buffer against demand variability
- Order Up To Level is displayed as a light green bar at approximately 30 units
- The system recommendation is clear: "DECISION: Order 15.3 units"

The 15.3-unit recommendation follows the enhanced order-up-to logic: when the order arrives, the inventory will reach 30.3 units, providing substantial coverage above the ROP while maintaining efficient capital utilization.

The lower chart in Figure 4.3 presents the capital investment implications across different service levels:

- 90% service level requires €124.35 of invested capital (15.1 units total inventory)
- 95% service level requires €138.87 of invested capital (16.8 units total inventory)
- 99% service level requires €165.48 of invested capital (20.1 units total inventory)

The capital investment figures are calculated by multiplying the total inventory requirement by the unit cost:

$$Capital\_Invested = ROP \times Unit\_Cost = (\mu_{ML} \cdot L + SS) \times Unit\_Cost \tag{4.5}$$

Average unit costs are applied per category as describe before:

- FOODS: €5.50 per unit (acquisition and holding cost)
- HOBBIES: €12.75 per unit (higher-value items)
- HOUSEHOLD: €8.25 per unit (standard products)

In this research, a decision was made to implement inventory management at the store category level rather than for individual SKUs. This approach was chosen for several reasons:

- **Data structure constraints:** The forecasting framework outputs predictions aggregated at store-category level, providing a natural foundation for inventory decisions.
- **Statistical robustness:** Aggregated forecasts typically exhibit lower error rates than individual SKU forecasts due to the portfolio effect.
- **Operational relevance:** Store managers typically make initial decisions at the category level before refining to individual SKUs.
- **Computational efficiency:** This approach enables faster decision-making while capturing essential demand patterns.

Key advantages of this digital twin implementation include:

- **Real-time adaptation**: Dynamic recalculation as demand patterns evolve

- **Financial transparency**: Immediate visualization of service level trade-offs
- **Pattern-specific optimization**: Adaptive calculations for different demand characteristics
- **Scenario simulation**: Instant "what-if" analysis capabilities
- **Logical ordering consistency**: Ensures post-order inventory always exceeds reorder thresholds

The digital twin architecture ensures that inventory decisions are based on dynamic ML predictions that capture complex patterns revealed in the exploratory analysis, closing the loop between advanced analytics and operational execution. By guaranteeing that orders bring inventory above the reorder point, the system maintains operational stability while optimizing inventory investment.

While this chapter focused on demonstrating the practical implementation of the forecasting models, the following chapter will evaluate the performance of this approach and assess its reliability in different retail scenarios.

4.2 Evaluation of Operational KPIs and Results

This section evaluates the practical impact of the digital twin APS system on key operational metrics, focusing on forecasting accuracy and its implications for inventory management. An important aspect is understanding not just how accurate the forecasts are, but how the system manages inevitable prediction errors in practice.

Before presenting the results, it's important to clarify the choice of forecasting accuracy metric. During preliminary model selection in Section 3.1, Was initially employed Root Mean Squared Error (RMSE) to evaluate individual SKU performance. However, when scaling to the full implementation across diverse product categories, RMSE proved inadequate for cross-category comparisons due to significant differences in demand scales. To address this limitation, was adopted the Root Mean Squared Scaled Error (RMSSE), defined as:

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{N}\sum_t (y_t - \hat{y}_t)^2}{\frac{1}{N-1}\sum_t (y_t - y_{t-1})^2}} \tag{4.6}$$

The RMSSE normalizes forecast errors relative to a naïve random-walk model that simply predicts tomorrow's sales will equal today's. An RMSSE below 1 indicates superior performance compared to this baseline, making it ideal for comparing accuracy across product categories with vastly different sales volumes.

The comprehensive evaluation revealed consistently strong performance across all store-category combinations. As shown in Tables 3.5, 3.6, and 3.7, all RMSSE values remained below 1.0, confirming the superior performance of ML-based forecasting over naïve approaches:

- **FOODS**: Average RMSSE of 0.8308, with all stores achieving RMSSE < 1.0 despite dealing with intermittent demand patterns
- **HOBBIES**: Average RMSSE of 0.6692, representing the strongest performance despite exclusively lumpy demand patterns
- **HOUSEHOLD**: Average RMSSE of 0.8112, with mixed pattern types (60% lumpy, 40% intermittent)

The pattern-specific performance revealed important insights. Notably, the HOBBIES category, characterized entirely by lumpy demand patterns, achieved the lowest average RMSSE. This counterintuitive result demonstrates the effectiveness of the Two-Stage forecasting approach specifically designed for intermittent demand. The specialized handling of zero-inflation through SMOTE balancing enables the model to capture the sparse but predictable patterns in hobby products.

While RMSSE demonstrates that the models outperform baseline approaches, it doesn't reveal the practical scale of forecast errors in daily operations. The absolute error metrics from the evaluation provide insights into the magnitude of prediction uncertainties the APS system must manage:

- **FOODS**: Average RMSE of 2.55 units/day, MAE of 1.16 units/day
- **HOBBIES**: Average RMSE of 1.46 units/day, MAE of 0.46 units/day
- **HOUSEHOLD**: Average RMSE of 1.29 units/day, MAE of 0.51 units/day

These values indicate daily prediction errors ranging from 0.46 to 1.16 units - less than 2 products per day for any category. The relationship between these three error metrics reveals important insights about forecast patterns:

MAE-RMSE Relationship: RMSE values consistently exceed MAE across all categories (FOODS: 2.20x, HOBBIES: 3.17x, HOUSEHOLD: 2.53x), indicating:

- **MAE** (Mean Absolute Error) indicates how much, on average, the forecasts deviate from the actual values each day.
- **RMSE** (Root Mean Square Error) gives extra weight to large errors by squaring the differences before calculating their average.

- The significant gap between RMSE and MAE suggests that most daily forecasts are accurate (low MAE), but occasionally the system makes larger errors, particularly during promotional events or sudden spikes in demand.

MAPE Limitations: MAPE values ranging from 62-92% appear concerning but are misleading due to mathematical characteristics: MAPE (Mean Absolute Percentage Error) calculates errors as a percentage of actual values, for intermittent demand with frequent zero sales, small absolute errors (e.g., predicting 1 unit when actual is 0) result in infinite percentage errors, when actual sales are very low (1-2 units), even small errors create inflated percentages (e.g., predicting 2 when actual is 1 = 100% error)

This pattern is characteristic of retail environments where:

- Regular days show predictable sales patterns with minimal errors
- Special events or unexpected circumstances create temporary but substantial forecast misses
- Zero-sales days, common in intermittent demand (>60% for many products), make MAPE mathematically unstable

The critical question becomes: **how does the APS system ensure these forecast uncertainties don't disrupt operations?**

To illustrate how these protection mechanisms work in practice, let's revisit the HOUSEHOLD CA1_1 example from Section 4.1, now examining how the system maintains operational stability despite forecast uncertainties. While the previous section demonstrated the calculation methodology, here we analyze the robustness of these calculations against prediction errors, The system employs a multi-layered approach to manage forecast uncertainty:

1. Dynamic Safety Stock Protection

The safety stock formula directly incorporates forecast uncertainty through σ_{ML} (the standard deviation of predicted demand values):

$$SS = z \cdot \sigma_{ML} \cdot \sqrt{L} \cdot pattern\_factor \tag{4.7}$$

Example (HOUSEHOLD at store CA_1):

- Demand pattern: Intermittent ($\phi = 1.0$).
- Daily demand prediction: $\mu_{ML} = 1.11$ units/day.
- Standard deviation of predicted values: $\sigma_{\text{pred}} = 1.71$.
- Lead time: $L = 8$ days; Service level: 95% ($z = 1.65$).

- $$SS = z \cdot \sigma_{\text{pred}} \cdot \sqrt{L} \cdot \phi = 1.65 \times 1.71 \times \sqrt{8} \times 1.0 \approx 8.0 \text{ units}$$

- $$ROP = \mu_{ML} \cdot L + SS = 1.11 \times 8 + 8.0 = 16.88 \approx 17 \text{ units}$$

2. Pattern-Adaptive Error Management

The pattern factor adjusts protection based on demand characteristics:

- Lumpy patterns (factor = 1.2): Additional 20% safety stock for unpredictable spike patterns
- Intermittent patterns (factor = 1.0): Standard protection for moderate irregularity
- Regular patterns (factor = 0.9): Reduced protection for predictable demand

3. Order-Up-To Buffer System

Beyond safety stock, the system adds a 50% buffer during ordering:

$$OrderUpToLevel = ROP + \mu_{ML} \cdot L + \mu_{ML} \cdot L \cdot 0.5 \tag{4.8}$$

As detailed in Section 4.1, this 50% buffer ensures that when the order arrives, inventory will be comfortably above the reorder point, preventing the need for immediate reordering and providing coverage for sustained operations.

4. Practical Error Tolerance in Action

Let's examine how these protections work with actual forecast errors the example of the picture in chapter 4.1. For a HOUSEHOLD store with:

- Daily demand forecast: 1.13 units
- Lead time: 8 days
- Pattern type: Intermittent

When current inventory (15 units) falls below ROP (16.83 units):

- Safety stock protection: 7.82 units
- Order Up To Level = 16.83 + (1.13 × 8) + (1.13 × 8 × 0.5) = 30.35 units
- Order Quantity = 30.35 - 15 = 15.35 units
- Total protection after order: 15.35 units above immediate needs

This multi-layered protection ensures the system remains operational even when forecasts miss significantly, as evidenced by the high MAPE values.

The digital twin APS system's effectiveness stems from treating forecasts as probability distributions rather than point estimates. By maintaining RMSSE < 1.0 while strategically managing absolute error ranges through calibrated buffers, the system achieves:

- Consistent service levels despite forecast errors ranging from 60-90% (MAPE)
- Optimized inventory levels that adapt to pattern-specific error characteristics
- Reduced stockout events through proactive error management
- Efficient capital utilization by avoiding excessive inventory accumulation

Regional variations also emerged from the analysis. Wisconsin stores generally showed better forecasting performance across all categories, potentially reflecting the stronger impact of SNAP disbursement patterns that the model was specifically designed to capture. The pattern factors and buffer calculations adapt automatically to these regional differences, ensuring consistent performance across geographic locations.

These results validate the practical value of integrating machine learning forecasts with dynamic inventory optimization. The system's ability to maintain operational stability despite significant forecast errors, as measured by RMSE, MAE, and MAPE, demonstrates its readiness for real-world implementation. Rather than pursuing perfect prediction accuracy, the APS system achieves robust performance through intelligent error management, transforming forecast uncertainty from a critical vulnerability into a manageable operational parameter. A comprehensive and detailed discussion on the overall quantitative impact of the proposed strategies, including the limitations and trade-offs identified in practical applications, will be presented in the next chapter.

Chapter 5

Limitations and Future Research Directions

5.1 Trade-Offs and Limitations

While this research demonstrates the feasibility of integrating machine learning with inventory optimization, several important limitations constrained the scope and depth of the implementation, revealing critical trade-offs between theoretical potential and practical constraints.

The implementation was conducted on a personal laptop (Intel i5-1345U @ 1.60 GHz with 16GB RAM) due to budget problem, the computational capacity severely limited model optimization opportunities. The available processing power forced a compromise between model sophistication and computational feasibility, requiring simplified architectures that could train within reasonable time. Access to GPU-accelerated infrastructure would unlock exhaustive hyperparameter search.

With additional computational resources, several enhancements could have been explored:

- Systematic hyperparameter optimization for LightGBM and Two-Stage models (an attempt was made but computation time exceeded 4 days, forcing termination, due to the huge number of rows)
- Deeper feature engineering with automated selection and interaction term discovery. This would include implementing genetic algorithms for feature selection, automated creation and testing of polynomial interactions between temporal and categorical features, and exploration of complex feature combinations (e.g., price × seasonality × SNAP interactions). Such automated feature discovery could potentially identify non-obvious relationships between variables but would require substantial computational

power for exhaustive search and validation

Several operational parameters were selected based on industry averages rather than empirical optimization:

- **Lead time distributions**: Category-specific ranges (FOODS: 3-5 days, HOBBIES: 5-12 days) were assumed from typical retail operations rather than derived from actual supplier data
- **Lead time variability**: The implementation used fixed lead time values for each product category, while real-world supply chains experience significant variability in procurement times. This simplified approach may underestimate the safety stock requirements needed to accommodate delivery uncertainties.
- **Safety buffer percentages**: The 50% buffer in the order-up-to level was selected based on conservative retail practices, not optimized through simulation
- **Pattern factors**: The multipliers for different demand patterns (lumpy=1.2, intermittent=1.0, regular=0.9) were calibrated through limited trial runs rather than exhaustive testing
- **Cost assumptions**: Unit costs per category were estimated from typical retail markups (€5.50 for Foods, €12.75 for Hobbies, €8.25 for Household), lacking specific supplier agreements or actual procurement data

These assumptions, while reasonable, represent a trade-off between implementation speed and optimal performance calibration. The lack of real-world calibration data meant that these parameters, though based on industry standards, may not reflect the specific operational realities of different retail environments or supply chain configurations.

The store-category level aggregation, though operationally practical, represents a compromise that affects scalability:

Granularity limitation: Although the forecasting models were initially designed for individual SKU-level predictions, challenges with categorical variable encoding and memory constraints led to aggregation at the store-category level. This decision was appropriate for the research context given the nature of the available data and computational limitations. For an initial proof-of-concept demonstrating ML viability in inventory optimization, the store-category level provided sufficient granularity to capture important demand patterns while remaining computationally feasible. However, this aggregation inherently sacrifices product-specific precision and may not fully capture individual SKU behaviors or cross-product interactions within categories

Despite the richness of the Walmart M5 dataset, certain limitations affected model performance:

- **Missing metadata**: Lack of detailed product attributes such as product descriptions, names, or images constrained similarity-based forecasting for new products
- **External factors**: Limited visibility into competitor actions, marketing campaigns, and local economic indicators
- **Supply chain visibility**: Absence of supplier reliability data and transportation delays in the dataset
- **Product lifecycle modeling**: This research did not explicitly model product maturity stages or lifecycle effects (introduction, growth, maturity, decline). All SKUs were treated uniformly in the forecasting model, assuming stationarity in demand behavior over time. In real-world retail, especially for seasonal or trend-driven items, incorporating lifecycle indicators could further refine the predictions.

The Two-Stage approach, while effective for intermittent demand, introduced several trade-offs:

- **Model complexity**: Maintaining two specialized models for product category increased maintenance overhead. The Direct approach was never utilized as the data characteristics consistently indicated the need for Two-Stage modeling across all patterns
- **Training time**: Sequential training of classification and regression models extended overall development cycles

Several factors would complicate real-world deployment:

- **Change management**: Transitioning from traditional inventory systems requires substantial organizational adaptation
- **Integration costs**: Connecting ML forecasts with existing supply chain systems demands significant IT and hardware investment
- **Skill requirements**: Operating and maintaining the system necessitates specialized data science expertise

- **Validation challenges**: Proving return on investment (ROI) in production environments remains difficult without extensive testing. This limitation stems from the inability to run controlled experiments where one store operates with the ML system while another maintains traditional methods for direct comparison. Without such parallel testing, attributing performance improvements specifically to the ML system becomes challenging, as numerous external factors (seasonal changes, market conditions, supply chain disruptions) can influence inventory metrics simultaneously

A more extensive temporal validation was not possible, as the validation dataset provided only 30 days of data, limiting the ability to be sure about model performance over longer time horizons.

These limitations highlight a fundamental trade-off in AI implementation: while machine learning can theoretically optimize supply chain decisions and help managers make "informed" decisions, the practical benefits are highly dependent on available computational resources, organizational readiness, and data quality. Current research demonstrates the feasibility of the approach in constrained environments with the assumptions noted above, demonstrating that even modest hardware can yield significant improvements, while also revealing the significant potential for organizations with the appropriate technical infrastructure.

5.2 Future Research Directions

This research has demonstrated the potential of machine learning for demand forecasting and inventory optimization in retail environments characterized by intermittent demand patterns. The implementation of a Two-Stage forecasting approach integrated with an Advanced Planning and Scheduling system has shown promising results across different product categories and store locations. However, as a research endeavor, this work represents only a starting point from which several future research directions can be identified.

The methodological framework developed in this thesis could be extended in various ways to enhance both the theoretical understanding and practical application of machine learning in inventory management. Several high-potential research avenues emerge from the current limitations and findings, representing opportunities for significant advancement in this field.

Advanced modeling approaches could transform how forecasting integrates with inventory decisions. The current research relied primarily on gradient boosting methods, which offer an effective balance between accuracy and computational efficiency. However, more sophisticated architectures could potentially capture more complex patterns in retail data.

- **Deep learning architectures for complex patterns**: Recurrent neural networks with attention mechanisms or transformer-based models could better capture seasonality and event impacts in retail data. These architectures excel at identifying long-term dependencies that simpler models miss, such as how an event 12 months ago affects current demand. With additional data like product descriptions and attributes (unavailable in the current dataset), these models could further improve by learning rich product representations and their relationship to demand patterns.

- **Reinforcement learning for direct inventory optimization**: Instead of using machine learning only for forecasting, reinforcement learning could directly learn optimal inventory policies thanks to the feedback. This would work by simulating thousands of inventory scenarios where an AI agent makes ordering decisions, receives rewards based on profits and service levels, and gradually learns to balance conflicting objectives like minimizing holding costs while avoiding stockouts thanks to learing from the problematic situation with feedbacks. The key advantage is optimization directly for business metrics rather than forecast accuracy.

Beyond methodological enhancements, future research should address the multi-echelon nature of retail supply chains. The current implementation focused exclusively on store-level inventory decisions, treating each location as an independent entity. This approach, while computationally efficient, doesn't capture the interconnected nature of modern retail networks. A more comprehensive framework would consider the entire supply chain as an integrated system, optimizing inventory placement throughout the network rather than at individual nodes in isolation.

Such a network-oriented approach could reduce overall inventory requirements while maintaining or improving service levels through more intelligent allocation of stock. By considering distribution centers, warehouses, and stores simultaneously, the system could leverage geographic diversification to reduce safety stock requirements. This direction is particularly promising as it addresses a fundamental limitation of current inventory systems, which often optimize locally rather than globally.

- **Integrated multi-echelon inventory management**: Future research could expand beyond store-level decisions to simultaneously optimize inventory across stores, distribution centers, and warehouses. This approach would determine not just when to order, but also where to position inventory throughout the network, potentially reducing total safety stock requirements through risk pooling while maintaining service levels.

- **Dynamic sourcing decisions**: More advanced models could incorporate transportation disruptions (strikes, weather events, port congestion), supplier reliability metrics,

and geographical factors to make intelligent sourcing decisions. This would enable the system to recommend not just order quantities but also optimal sourcing locations based on current supply chain conditions, balancing proximity, lead time reliability, and cost factors dynamically.

The current research demonstrated the value of incorporating calendar events and SNAP disbursement data into forecasting models. This success suggests that further data integration could yield additional improvements in prediction accuracy. Future implementations can explore more comprehensive data integration strategies, particularly for products with demand heavily influenced by external factors like weather conditions and other environmental and non-controllable variables. The goal would be to capture not just historical sales patterns but also the contextual environment in which these patterns exist.

Particularly promising is the modeling of inter-product relationships. The current implementation treats each product independently, but consumer purchasing behavior often involves complementary or substitute products. Capturing these relationships could significantly improve forecast accuracy, especially during promotional periods when cross-product effects are most pronounced.

- **Complementary and substitute product modeling**: Beyond treating each product independently, future research could model relationships between products that are frequently purchased together (complements) or instead of each other (substitutes). This approach would capture how a price change or promotion on one product affects demand for related items, improving forecast accuracy especially during promotional periods. For example, a model could learn that a promotion on pasta sauce increases pasta sales by 30%, automatically adjusting inventory for both products.

- **Real-time environmental factors**: Though the current research incorporated calendar events and SNAP impacts, future implementations could integrate real-time factors like weather conditions (capturing how rain affects in-store traffic), local COVID rates (affecting shopping patterns), or macroeconomic indicators (inflation affecting purchase behavior). These external variables could be especially valuable for improving forecasts during unusual market conditions.

For machine learning systems to be effectively adopted in practice, they must not only make accurate predictions but also engender trust among users. This is particularly important in inventory management, where decisions have significant financial implications and managers often rely on years of experience and intuition. Future research should address the human-system interaction aspects of AI-enhanced inventory management, developing frameworks that make model recommendations transparent and interpretable to non-technical users.

Effective inventory systems of the future will likely operate as human-AI (AI agent) partnerships rather than fully automated solutions. This collaborative approach requires interfaces that explain recommendations in business terms rather than technical metrics, allowing managers to understand not just what the system recommends but why. Additionally, these systems should be capable of learning from human expertise, gradually incorporating the knowledge embedded in manual overrides and adjustments.

- **Visual decision explanation interfaces**: For inventory systems to gain trust, managers need to understand why particular decisions are recommended. Future research could develop interfaces that visually explain recommendations, showing specifically how factors like upcoming events, recent sales trends, and lead time uncertainty contributed to an order quantity. These interfaces could include interactive "what-if" scenarios, allowing managers to see how changing assumptions would affect recommendations.

- **Learning from human expertise**: Advanced systems could observe when managers override recommendations, learn from these interventions, and gradually incorporate this expertise. For example, if managers consistently increase order quantities before school holidays beyond what historical data suggest, the system could learn this pattern and automatically adjust future recommendations, blending human knowledge with data-driven insights.

The cold start problem for new products represents one of the most challenging aspects of retail forecasting. Without historical sales data, traditional forecasting methods cannot be applied directly. Section 3.3 discussed similarity-based approaches conceptually, but this area requires significant further research. Future work could adapt techniques from computer vision and natural language processing to calculate product similarity based on attributes, descriptions, and images, enabling a more effective transfer of demand patterns from existing products to new introductions.

Another promising direction involves dynamic safety stock allocation based on forecast confidence. The current implementation used fixed pattern factors (1.2 for lumpy, 1.0 for intermittent, 0.9 for regular) to adjust safety stock calculations. A more sophisticated approach would continuously vary the levels of the safety stock based on the accuracy of the predicted forecast, maintaining higher buffers for products where the predictions have historically been less reliable. This would create a more efficient allocation of inventory protection, potentially reducing overall stock levels while maintaining service targets.

- **Similarity-based product matching**: To address the cold-start problem for new products, future research could adapt techniques from computer vision and natural

language processing to calculate product similarity. By analyzing product descriptions, images, and attributes, the system could identify the closest matching existing products and use their demand patterns as a starting point for forecasting. This would be particularly valuable for fashion, electronics, and seasonal items where historical data doesn't exist but similar products provide useful signals.

- **Dynamic safety stock based on forecast confidence**: Future implementations could adjust safety stock levels based on prediction uncertainty, maintaining higher buffers where forecasts have historically been less accurate. This approach would create a more intelligent buffer system that allocates inventory protection based on demonstrated forecast error patterns, potentially reducing overall inventory while improving service levels.

Finally, longitudinal validation studies conducted in partnership with retailers would provide valuable insights into long-term performance and adaptation of machine learning inventory systems. The current research was limited to historical data analysis and simulation, but real-world implementation would yield important insights about system robustness across multiple seasonal cycles and changing market conditions. Such studies thanks to continuous reinforcement learning updates would help bridge the gap between theoretical potential and realized business value, informing both research directions and implementation strategies.

Another promising research direction involves adapting demand forecasting and inventory optimization models for SMEs with limited resources. Previous research has highlighted how SMEs face specific challenges in aligning their IT strategies with business objectives due to limited financial resources, insufficient IT skills, and inadequate infrastructure (Kazemargi and Spagnoletti 2020). Despite these limitations, analysis of Italian SMEs' responses to government policies for Industry 4.0 shows growing interest in system integration technologies (12.73% of investments) and simulation (3.95%), which are fundamental components for implementing APS systems like the one presented in this research (Kazemargi and Spagnoletti 2020). Future studies could explore how to simplify and make machine learning-based forecasting models more accessible to SMEs, possibly through cloud services or shared platforms that could reduce financial and technical barriers to implementation.

Through these focused research directions, the foundation established in this work could evolve into more sophisticated inventory management systems that better balance efficiency and resilience in retail supply chains. By concentrating on these high-impact areas, future research can bridge the gap between theoretical possibilities and practical implementation in real-world retail environments.

Chapter 6

Conclusions

This research began with a practical challenge observed in retail environments: the persistent gap between machine learning models and operational inventory decisions. Through applying Design Science Research principles, I sought not only to develop a technical solution but to bridge theoretical knowledge with practical implementation and address a fundamental challenge optimizing the supply chain from factory to retailer. My driving motivation was to reduce product waste, minimize utilization of warehouse space, conserve resources, and streamline processes throughout the supply network.

The central question that inspired this thesis was: "How can we optimize the entire production chain in the most effective way?" This optimization would not only yield corporate improvements through reduced costs and increased efficiency but could also decrease waste and potentially lower consumer prices by reducing resources tied up in inventory. By targeting this optimization challenge, the goal is to create value at multiple levels: operational, economic, and environmental.

Working with the Walmart dataset revealed patterns that challenged initial assumptions about retail demand. The stark regional differences in consumer response to identical events, from SNAP disbursements to holidays underscored that even within a single retailer, universal forecasting approaches are fundamentally limited. This observation shifted my research focus from pursuing perfect prediction accuracy toward developing adaptable systems that acknowledge and accommodate local variability.

The prevalence of intermittent demand across product categories reinforced my conviction that retail inventory management requires specialized approaches. While literature often treats zero sales as anomalies, my analysis revealed they are the norm in many retail contexts. This realization led to the Two-Stage methodology that embraces rather than avoids this reality.

The implementation of the forecasting models and their integration with the APS system revealed the effectiveness of decomposing the problem into two fundamental questions:

whether a product will sell on a given day, and how much will sell if a transaction occurs. This Two-Stage approach proved particularly valuable for addressing the intermittent demand patterns that dominated the dataset, especially in the Hobbies category where over 80% of products showed zero sales for more than 60% of days. The methodology used SMOTE to balance the classification dataset in the first stage, addressing the challenge of class imbalance, before proceeding to the prediction of the quantity in the second stage when sales were expected to occur.

Empirical validation confirmed the superiority of this approach, with RMSSE values consistently below 1.0 in all product categories, MAE under 1.1643, demand patterns, demonstrating up to 33% improvement over traditional methods like ARIMA in the preliminary model selection. These performance improvements translated directly into more efficient inventory management through the Advanced Planning and Scheduling system, which dynamically adjusted safety stocks and reorder points based on forecast patterns.

Beyond measurable performance improvements, this research shifted my understanding of how machine learning creates value in operational contexts. The greatest value often came not from incremental accuracy improvements but from consistency, transparency, and reduced manual intervention.

The development process reinforced that effective DSR isn't merely about creating technological artifacts, it's about generating design knowledge that connects technical capabilities with human and organizational contexts. This perspective transformed how I approached validation, focusing not just on statistical performance but on whether the system effectively supported decision-making processes.

Looking beyond current limitations, envision inventory management systems that learn continuously from operational feedback, not just historical sales. Such systems would evaluate inventory decisions against actual outcomes, creating a reinforcement learning cycle that optimizes not just forecasts but the entire inventory policy.

As computational constraints diminish, the opportunity to implement truly personalized inventory strategies at individual SKU levels becomes viable. This granularity would unlock new optimization potential, particularly for retailers with diverse product portfolios.

The most promising frontier may be collaborative multi-echelon optimization, where retailers and suppliers share forecasting insights and collaboratively manage inventory positioning throughout the supply chain. This approach could transform traditionally adversarial supply relationships into data-driven partnerships.

This research demonstrates that the gap between advanced analytics and operational practice can be bridged through thoughtful design that respects both technical possibilities and practical constraints. The Two-Stage forecasting approach and its integration with

inventory systems represents not just a technical solution but a framework for thinking about how we model and respond to demand uncertainty.

Beyond business optimization, this work highlights how AI and forecasting, when properly implemented, can serve as powerful sustainability tools. The intelligent application of machine learning to inventory management represents an opportunity to address pressing environmental challenges through operational excellence. By precisely matching supply with demand, these systems can significantly reduce overproduction and waste issues that plague global supply chains and contribute to environmental degradation. In this sense, advanced forecasting becomes not merely a profit-maximizing tool but a means to steward resources more responsibly in our increasingly resource-constrained world.

As supply chains continue to face unprecedented volatility, the ability to make data-driven inventory decisions that adapt to local conditions and account for complex demand patterns will become increasingly vital. This work contributes one path toward a future where more efficient supply chains not only improve business performance but also reduce waste and resource consumption—potentially creating both economic and environmental benefits through the intelligent application of machine learning to inventory management.

Bibliography

Agrawal, N. et al. (2024). "How Machine Learning Will Transform Supply Chain Management". In: *Harvard Business Review*.

Amazon Web Services (2023). *Generate cold start forecasts for products with no historical data using Amazon Forecast*. URL: https://aws.amazon.com/blogs/machine-learning/generate-cold-start-forecasts-for-products-with-no-historical-data-using-amazon-forecast-now-up-to-45-more-accurate/.

Baskerville, Richard L. et al. (2018). "Design Science Research Contributions: Finding a Balance between Artifact and Theory". In: *Journal of the Association for Information Systems* 19.5, pp. 358–376.

Chen, T. and C. Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. URL: https://arxiv.org/pdf/1603.02754.

Chopra, S. and P. Meindl (2019). *Supply Chain Management: Strategy, Planning, and Operation*. 7th ed. Pearson.

GeeksforGeeks (2024). *Box-Jenkins Methodology for ARIMA Models*. URL: https://www.geeksforgeeks.org/box-jenkins-methodology-for-arima-models/.

Imbalanced-learn developers (2023). *SMOTE - Synthetic Minority Over-sampling Technique*. URL: https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.SMOTE.html.

Impact Analytics (2023). *AI-Driven Cold Start Modeling for Retail Demand Forecasting*. URL: https://www.impactanalytics.co/blog/ai-retail-demand-forecasts-cold-start-modeling-for-new-retail-products.

Kaggle (2020). *M5 Forecasting - Accuracy Competition*. URL: https://www.kaggle.com/competitions/m5-forecasting-accuracy.

Kazemargi, Niloofar and Paolo Spagnoletti (2020). "IT Investment Decisions in Industry 4.0: Evidences from SMEs". In: *Digital Business Transformation*. Ed. by Rocco

Agrifoglio et al. Vol. 38. Lecture Notes in Information Systems and Organisation. Springer, pp. 77–92.

Ke, G. et al. (2017). “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems*. Vol. 30.

Kelkar, A., V. Marya, and M. Mysore (2024). “An Early-Warning System Will Make Your Supply Chain More Resilient”. In: *Harvard Business Review*.

McKinsey & Company (2019). *Succeeding in the AI supply-chain revolution*. URL: https://www.mckinsey.com/industries/metals-and-mining/our-insights/succeeding-in-the-ai-supply-chain-revolution.

— (2020). *AI-driven operations forecasting in data-light environments*. URL: https://www.mckinsey.com/capabilities/operations/our-insights/ai-driven-operations-forecasting-in-data-light-environments.