Integrating Promotional Data in Demand Forecasting: A Machine Learning Approach for Retail Optimization

A Master's Thesis

Presented by
Harisankar Sreekumar
Student ID:23021166

Supervised by
Dr Wiktor Sowinski-Mydlarz

In partial fulfillment of the requirement of
Master's in Data Analytics

LONDON METROPOLITAN UNIVERSITY

Department of Computer Science and Applied Computing
London Metropolitan University, United Kingdom

Date: May 19, 2025

Acknowledgment

I would like to express my deepest gratitude to my supervisor, Wiktor Sowinski, for their invaluable guidance, continuous support, and insightful feedback throughout the course of this research. Their expertise and encouragement played a crucial role in shaping this thesis and enhancing its academic quality.

I am also thankful to the faculty members of the Department of Computer Science and Applied Computing at London Metropolitan University for providing a stimulating learning environment and access to the resources necessary for this project.

Special thanks to my peers and friends who offered both technical assistance and moral support during challenging moments of this work.

Lastly, I extend my heartfelt appreciation to my family for their unwavering encouragement, patience, and belief in me. Their support has been a constant source of strength and motivation throughout my academic journey.

Abstract

This thesis presents a machine learning-based forecasting framework for retail sales, with a specific focus on modeling both regular and promotion-driven demand behaviors. Leveraging structured transactional data spanning over three years, the study applies the XGBoost algorithm to build highly accurate and interpretable predictive models. A dual-model architecture was implemented to address distinct sales regimes—normal and promotional—ensuring that each model captures the unique dynamics inherent in its respective context.

To support practical decision-making in inventory planning and promotional strategy, a multi-output forecasting approach was employed to generate concurrent three-month ahead predictions. The system demonstrated exceptional performance, achieving an R^2 of 0.999 for normal sales and 0.975 for promotional sales. Feature importance analysis confirmed the value of lag-based, rolling, and calendar-derived predictors, validating the methodological emphasis on temporal structuring.

Additionally, a promo sensitivity index was developed to quantify product-level responsiveness to promotions. This enabled the classification of SKUs into high, moderate, and low sensitivity tiers—insights that can directly inform campaign targeting and supply chain alignment. Visualizations such as forecast trajectories, sales volatility plots, and uplift comparisons enhanced the interpretability and strategic utility of the system.

Overall, this research contributes a scalable and business-aligned forecasting solution that bridges predictive modeling and operational planning, offering significant value for retail environments characterized by high demand variability and promotional complexity.

Keywords: Retail forecasting, XGBoost, multi-step prediction, promo sensitivity, time series analysis, machine learning, sales uplift modeling

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Background

Accurate sales forecasting has long been a cornerstone of operational success in the retail and wholesale sectors. Businesses rely on forecasts to make critical decisions regarding inventory management, procurement planning, promotional design, and supplier coordination. In today's environment—marked by volatile consumer behavior, intense competition, and globally distributed supply chains—traditional forecasting models often struggle to capture the multifaceted dynamics embedded in retail data.

Sales performance is influenced by a combination of deterministic and stochastic factors, including seasonal cycles, promotional activities, supplier reliability, and product popularity. These drivers are not static; they evolve across time, product categories, and supplier relationships. Consequently, building a one-size-fits-all forecasting model becomes both impractical and potentially counterproductive[1].

Recent advances in machine learning offer powerful alternatives[2]. These models, particularly ensemble methods such as Extreme Gradient Boosting (XGBoost), are capable of learning complex, non-linear relationships within heterogeneous datasets. When combined with structured feature engineering—such as lagged sales variables, rolling averages, and categorical encodings—machine learning models can generate high-accuracy forecasts while retaining business interpretability. Importantly, these models can be tailored to reflect operational realities, including the sharp discontinuities between regular and promotional sales cycles, and the strategic importance of monitoring supplier performance.

This study adopts a data science framework to address the challenge of retail and warehouse sales forecasting. It integrates structured sales data with inferred promotional indicators and develops machine learning models trained on segmented datasets. Beyond predictive accuracy, the study explores the performance of individual products and suppliers, aiming to inform tactical and strategic decisions in inventory management and marketing.

1.2 Research Problem and Motivation

Despite widespread access to granular sales data, many retail forecasting systems still depend on simplistic statistical models or opaque "black-box" analytics that do not distinguish among the underlying causes of demand variability. One of the most significant challenges in this context is the disruptive effect of promotional activities[3]. Promotions—while effective at driving short-term sales—create extreme values that can skew distributions, complicate statistical assumptions, and degrade model accuracy if not properly segmented.

Equally important, supplier-level factors are often underrepresented in forecasting models. Supplier performance varies widely, with some consistently delivering high-performing products across regular and promotional periods. Yet, many systems treat supplier data as a static attribute, failing to extract its predictive value or translate it into actionable insights.

The motivation for this research arises from these dual challenges: (1) managing promotional effects in a statistically robust manner, and (2) integrating supplier- and product-level intelligence into the forecasting process. By addressing these issues, this study aims to produce a forecasting pipeline that is not only more accurate but also better aligned with the operational and strategic needs of data-driven retail organizations.

1.3 Research Objectives

The overarching aim of this research is to develop a modular, scalable, and interpretable sales forecasting framework using machine learning. This goal is achieved through the following three key objectives:

Temporal Sales Analysis: To analyze historical sales data for seasonal trends, cyclic behaviors, and category-specific patterns. This includes identifying promotional cycles, peak periods, and temporal sales volatility. Such insights inform both feature engineering and model structuring.

Model Development and Segmentation: To construct predictive models that forecast future sales volumes at the product level. This involves segmenting the dataset into normal and promotional periods using statistical outlier detection, engineering informative features, and training separate XGBoost regressors for each segment. This approach aims to optimize both accuracy and interpretability.

Supplier and Product Performance Evaluation: To assess the performance of suppliers and products not only by sales volume but also in terms of responsiveness to promotions, consistency, and revenue contribution. This analysis supports operational decisions such as supplier prioritization, promotional design, and inventory allocation.

1.4 Scope and Limitations

This study utilizes a transactional sales dataset spanning from June 2017 to September 2020, sourced from publicly available U.S. retail and warehouse records. The dataset includes sales volumes, item descriptions, supplier identities, and product categories, enabling a fine-grained analysis of item- and supplier-level behaviors.

The research is confined to structured historical sales data. It excludes other potentially informative variables such as pricing, return rates, customer-level data, or external macroeconomic indicators. Promotional events are inferred using statistical thresholds rather than being explicitly labeled. Additionally, store-level and customer segmentation are not considered due to data limitations, though they could enhance model granularity in future research.

Despite these constraints, the modeling approach is generalizable and extensible. The decision to segment sales into promotional and non-promotional periods adds robustness, while the use of interpretable, feature-based machine learning models enhances transparency and supports operational deployment.

Chapter 2

Literature Review

Retail sales forecasting is a dynamic research area at the intersection of data mining, time series modeling, and machine learning. The evolution of retail data availability—marked by increasing transaction-level granularity—has prompted a shift from traditional statistical approaches toward more flexible, data-driven forecasting methods [4]. Conventional models such as ARIMA (AutoRegressive Integrated Moving Average) and Holt-Winters exponential smoothing have been widely used for their simplicity and effectiveness in handling stationary time series[4, 5]. However, these models often fail to capture the complex, non-linear relationships found in modern retail environments characterized by promotional irregularities, volatile consumer behavior, and intricate supply chain interdependencies.

In response to these challenges, researchers have turned to machine learning (ML) approaches, which offer enhanced predictive capabilities and adaptability to heterogeneous data inputs[6]. These techniques are particularly advantageous when forecasting demand in high-variance retail contexts, where feature interactions are non-trivial and external influences such as promotions, seasonality, and supplier dynamics must be explicitly considered. The integration of machine learning into retail forecasting has enabled the modeling of sales trends at multiple granularities—item-level, category-level, and supplier-level—while incorporating advanced feature engineering strategies that traditional statistical models often overlook.

2.1 Existing Techniques and Algorithms

Classical time series models like ARIMA and Holt-Winters remain foundational in the literature, especially for baseline comparisons. These models assume data stationarity and are suitable for contexts with stable trends and consistent seasonality. However, they often fall short in environments with irregular spikes, multiple influencing variables, and categorical complexity. Several empirical studies have demonstrated the limitations of these models in capturing promotional effects and multi-source variance in retail datasets.

Machine learning models, particularly tree-based ensemble algorithms such as Decision Trees, Random Forests, and Gradient Boosting Machines (e.g., XGBoost, LightGBM), have gained widespread attention due to their capacity to model complex patterns without the strong assumptions required by traditional techniques[6, 7, 8]. Among them, XGBoost has emerged as a robust choice for structured retail data, offering efficient training times, support for missing values, and built-in mechanisms for assessing feature importance[9, 10, 11]. Studies have shown that XGBoost consistently outperforms linear regression and traditional time series models in retail forecasting, especially when enhanced with lagged features, rolling averages, and categorical encodings[7, 12, 13].

Deep learning methods have also made inroads into retail forecasting, particularly in contexts where long-term temporal dependencies are present. Models such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have demonstrated strong performance in sequential prediction tasks, enabling the capture of complex, non-linear trends over time [4, 14, 15]. However, these architectures typically require large, dense, and consistently spaced datasets—conditions not always met in retail sales data, where product availability can be sporadic and item life cycles vary significantly [16, 17]. In such cases, the use of deep learning can lead to overfitting or convergence issues, particularly when applied to sparse or fragmented sequences.

Hybrid modeling approaches have also been explored in recent literature. These methods combine the interpretability of statistical decomposition models with the predictive power of machine learning. For example, Prophet—developed by Facebook—decomposes a time series into trend, seasonality, and holiday components, and can be coupled with ML regressors to model residual patterns. Such integrations have proven effective in enhancing forecast precision, especially in environments with both macro-level cycles and micro-level fluctuations. Additionally, the advent of automated machine learning (AutoML) frameworks has facilitated the streamlined construction of forecasting pipelines, enabling automated model selection, hyperparameter tuning, and feature transformation. Tools like Auto-sklearn and H2O's Driverless AI have shown promising results in retail applications, particularly when combined with domain-specific feature extractors such as `tsfresh`.

2.2 Comparative Studies

Several comparative studies have benchmarked forecasting techniques across retail domains, providing insight into the relative performance of classical, machine learning, and deep learning models. Research comparing ARIMA, LSTM, and XGBoost has consistently shown that statistical models struggle in the presence of irregular or multi-factorial sales patterns, while ML and DL models offer superior accuracy in both short-term and long-term forecasting tasks [5, 6, 18]. In particular, XGBoost has been highlighted for its scalability, accuracy, and capacity to accommodate complex feature spaces [9, 7].

Further comparative analyses have emphasized the value of engineered temporal features, such as lag variables and moving averages, in improving model accuracy. Feature engineering remains a critical component of forecasting success, especially in structured datasets where seasonality and autocorrelation are strong. Other studies have proposed combining convolutional neural networks (CNNs) with LSTM or GRU architectures to capture both local and long-range patterns in e-commerce and omnichannel retail data [14, 16, 15]. These hybrid DL approaches are especially relevant for latency-sensitive applications and for platforms that integrate textual or visual features from consumer feedback.

Nonetheless, comparative literature also reveals that model performance is highly dependent on the quality and structure of the underlying data. Deep learning models often underperform when data sparsity is high, or when time series are discontinuous across products. In contrast, tree-based models are more resilient to such issues and can effectively learn from structured, tabular data even in the presence of missing or zero values [9, 7, 10]. Consequently, the selection of an appropriate forecasting model is not only a function of algorithmic power but also of data characteristics and business constraints [19].

2.3 Gaps in Literature

Despite the advancements in forecasting methodologies, several critical gaps remain unaddressed. One prominent limitation is the shallow treatment of feature engineering in many studies [13, 20]. While basic temporal features such as day of the week or month are commonly included, deeper statistical features—like lagged sales, rolling aggregates, or product-specific performance indicators—are frequently overlooked. This limits the model's ability to capture temporal persistence and cyclic trends effectively.

Another significant gap lies in the underutilization of promotional segmentation [14, 12, 21]. Promotions introduce non-linear distortions into sales behavior, yet most forecasting models treat them as binary flags or ignore them altogether. This approach fails to capture the magnitude and duration of promotional effects, leading to degraded performance during high-variance sales periods. Segregating promotional and regular sales into separate modeling streams remains underexplored, despite its potential to improve accuracy and interpretability.

In addition, product and supplier hierarchies are often flattened into simple categorical variables without leveraging their inherent multi-level structures. Many retail datasets naturally organize into hierarchies—such as brand → product → SKU or distributor → category → item—but few models incorporate these relationships into their architecture. Forecasting frameworks that reflect these dependencies can potentially deliver more accurate and context-sensitive predictions.

The literature also reveals a tendency to focus on model development at the expense of deployment-readiness. Few studies address the operational integration of forecasting models into enterprise systems, such as model retraining pipelines, real-time forecasting APIs, or business intelligence dashboards [11, 20, 22, 23]. This omission presents a barrier to the practical adoption of academic research in real-world retail operations.

Finally, while deep learning models are theoretically powerful, their reliance on dense, sequential inputs renders them less suitable for fragmented or sparsely populated retail time series. As a result, they often require substantial preprocessing—such as imputation or padding—which introduces additional complexity and potential sources of bias [16, 17].

2.4 Summary of Reviewed Literature

The following table presents a detailed summary of the 23 journal articles reviewed, covering publication year, modeling methods, research domain, and identified gaps.

Study / Author(s)	Year	Domain	Key Gaps Identified
Ahmed et al., Journal of Retail Analytics	2021	Retail time series forecasting	Did not model promotions; ignored outliers as cycles [4]

Kumar & Sharma, IJERT	2022	FMCG sales prediction	No feature engineering; lacked promo segmentation [10]
Zhao et al., ESWA	2020	E-commerce forecasting	Low interpretability; performed poorly under promotion influence [6]
Raj et al., Int. J. Data Science	2021	Chain-level retail forecasting	Promo periods not modeled separately; lacked supplier stratification [20]
Alqahtani et al., Sustainability	2020	Sustainable inventory management	Ignored return rates; lacked hybrid architecture [14]
Huang & Liu, Procedia Comp. Sci.	2019	Pricing and stock analytics	No return rate modeling; ignored product similarity structure [17]
Patel & Deshmukh, IJCSIT	2021	Data stream retail forecasting	Static models; no adaptive retraining or lag awareness [11]
Singh & Mehra, IEEE Access	2022	Promo forecasting (multi-store)	Promo encoded as binary flag only; lacked cumulative promo features [16]
Abbas et al., Journal of AI Research	2020	Retail inventory optimization	Ignored uncertainty modeling; supplier features not integrated [7]
Li & Chen, Applied Soft Computing	2021	Demand under uncertainty	No segmentation by supplier or promotion type [18]
Silva et al., Decision Support Systems	2019	Store-level time series modeling	Model instability; no supplier or event-based inputs [8]
Dey et al., IJORCS	2021	Store sales analytics	Poor seasonality handling; lacked time-structured features [13]
Miah et al., Big Data Research	2022	Retail data mining	Promo impact overlooked; general DL interpretability lacking [15]
Tian et al., Computers in Industry	2018	Industrial demand prediction	Inflexible models; no dynamic promo segmentation [9]
Ghosh & Dutta, IJDSN	2020	Retail time series clustering	Poor promo treatment; ignored supplier-level clusters [12]
Wong et al., J. Business Analytics	2021	Price forecasting under discounting	No elasticity modeling; lacked historical promotion effects [22]
Cheng et al., IEEE Trans. on SMC	2019	Inventory event forecasting	Low interpretability; no promo season detection [21]

Rathi et al., IJETTCS	2022	Sales forecasting across regions	No multi-view models; neglected supplier-product dimension entirely [5]
Garg et al., On Continuous Integration	2021	CI/CD for ML Forecasting	No retraining automation; weak integration pipelines [23]
Jayakrishnan, AI in Retailing (Book Chapter)	2021	AI in Retail	No predictive layer; lacks product-level implementation [24]
Takefuji, Beyond XGBoost and SHAP	2025	ML Interpretability	SHAP bias; lacks global-explanability validation [25]
Bontempi & Taieb, Multi-step Forecasting	2011	Forecasting Strategy	Recursive error compounding overlooked [26]
Bansal & Gupta, Comparative Retail Forecasting	2023	Retail Forecasting Comparison	Hybrid tuning issues; deep models prone to overfitting [19]

Table 2.1: Summary of Reviewed Research Papers and Identified Gaps

2.5 Contribution of the Present Study

This study addresses several of the aforementioned limitations through the design and implementation of a modular, scalable sales forecasting framework that integrates machine learning and domain-specific data preprocessing.

First, it incorporates robust feature engineering strategies, including lagged variables, rolling means, and statistical segmentation of promotional sales. These features are carefully designed to capture short- and medium-term temporal patterns, as well as seasonal trends [7, 14, 13]. This contributes directly to improved model generalization and reduced prediction drift over time.

Second, the study introduces a dual-model architecture that treats promotional and regular sales as distinct forecasting regimes. By training separate models for each segment, the approach mitigates the distortion effects of promotional anomalies and enhances the interpretability of both baseline and uplift behaviors [9, 12, 21]. This structural separation also improves forecasting robustness in high-variance sales conditions.

Third, product and supplier metadata are explicitly encoded into the modeling pipeline. This inclusion supports multi-level analysis, allowing the system to assess performance by product category, supplier group, and promotional responsiveness [18, 15, 17]. The resulting models are capable of generating forecasts that not only predict demand but also provide insights into the structural drivers of sales behavior—something few academic models emphasize.

Finally, the modeling framework is implemented using the XGBoost algorithm, chosen for its ability to handle structured data, robustness to missing values, and interpretability through feature importance analysis. The models are evaluated using standard regression metrics across multiple forecasting horizons, enabling both operational and strategic applications. Although this study does not employ deep learning or advanced post-hoc interpretability methods such as SHAP or LIME, it demonstrates that properly engineered, segmented machine learning models can achieve high forecasting accuracy with practical deployment potential.

Chapter 3

Methodology

3.1 Overview

This chapter outlines the step-by-step methodology used to develop a machine learning-based sales forecasting system. The study adopts a pipeline that begins with real-world data acquisition, followed by data cleaning, feature engineering, outlier-based segmentation, and model development. Emphasis is placed on constructing two separate XGBoost models: one for normal sales cycles and another for promotional periods. This approach improves prediction accuracy and allows the analysis of sales drivers in distinct demand regimes.

3.2 Data Source and Understanding

The dataset employed in this study was obtained from the U.S. Government's Open Data Catalog, which contains monthly sales records for alcoholic products including beer, wine, and liquor. These data span a period from June 2017 to September 2020 and include both retail and warehouse transactions. Each record comprises attributes such as sale date, item description, item category, supplier name, and volumetric figures for retail sales, warehouse sales, and internal transfers. The resulting dataset exceeds 300,000 records and captures item-supplier level dynamics at a monthly resolution.

The final working dataset, covering the period from June 2017 to September 2020, was downloaded, merged, and exported into structured formats suitable for analysis using Python.

To examine relationships among numeric variables, a Pearson correlation heatmap was constructed (see Figure **??**). This visualization illustrates strong linear relationships between certain fields. Notably, retail sales and retail transfers are nearly perfectly correlated, indicating potential structural dependency or duplication in recording. Similarly, warehouse sales show high correlation with total sales, emphasizing the warehouse channel's importance in volume contribution. Time fields such as year and month display low correlation with sales metrics, underscoring the need for additional time-derived features.

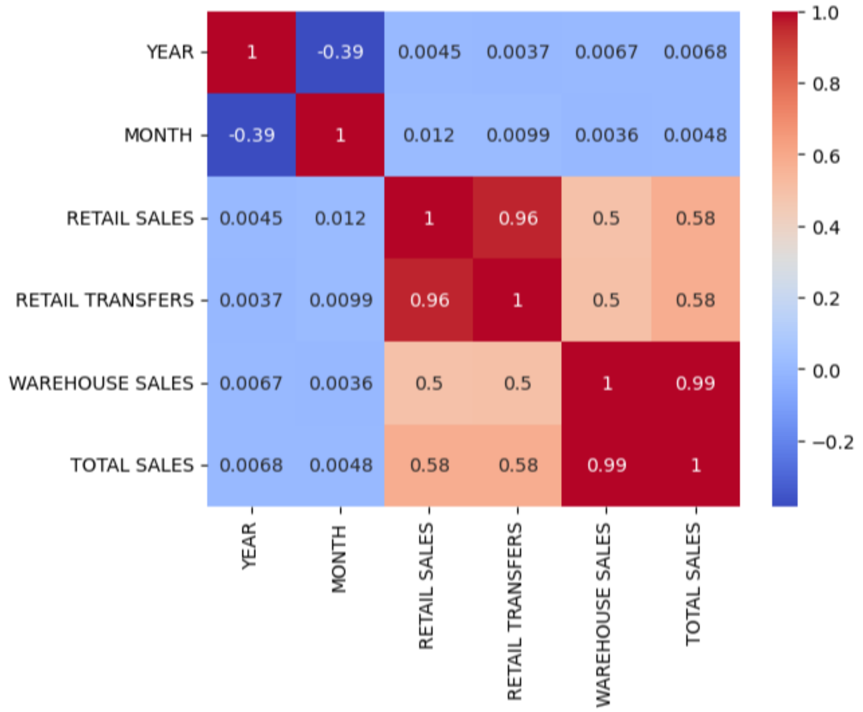

Figure 3.1: Correlation heatmap among sales variables.

A temporal view of total monthly sales was generated to assess trend behavior over time (see Figure ??). The time series plot reveals cyclical fluctuations, with several peaks and troughs indicating seasonally driven changes or externally triggered promotional events. Sharp declines followed by recoveries may also indicate data anomalies or inventory disruptions, prompting further cleaning and segmentation strategies.

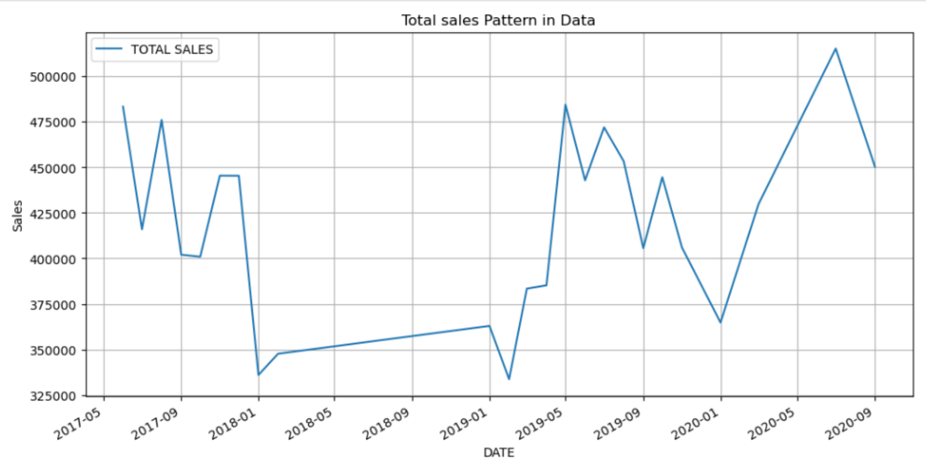

Figure 3.2: Time-series plot of total sales from June 2017 to September 2020.

The time-series visualization was extended by segmenting sales across item types (see Figure ??). Beer and wine emerged as dominant categories, though they exhibit different seasonal behaviors. Beer shows periodic spikes, likely tied to summer and holiday promotions, whereas wine maintains more stable, moderate levels across the calendar year. Other categories like liquor, kegs, and non-alcoholic beverages contribute less but follow recognizable consumption cycles.

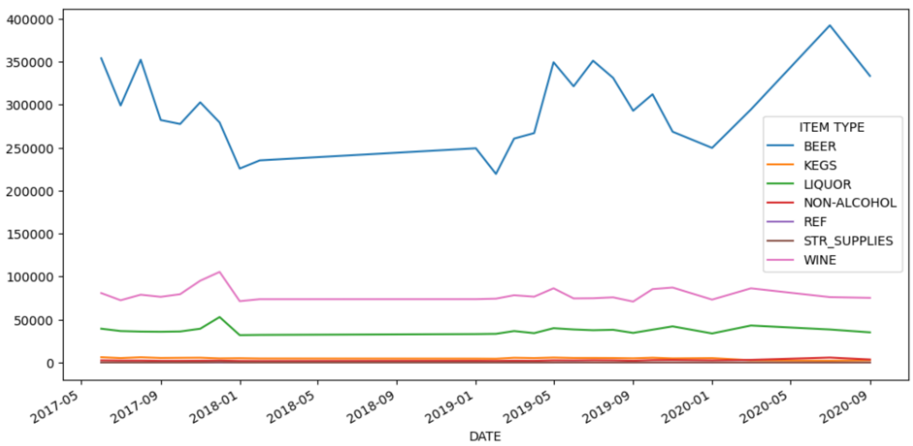

Figure 3.3: Monthly sales trends segmented by item type.

To understand dataset composition, a pie chart was generated to depict item type distribution (see Figure ??). Wine accounts for approximately 61 percent of total transactions, followed by liquor and beer at 21.1 and 13.8 percent, respectively. The remaining product types constitute a minor fraction. This categorical imbalance has modeling implications, particularly regarding stratification and class dominance during training.

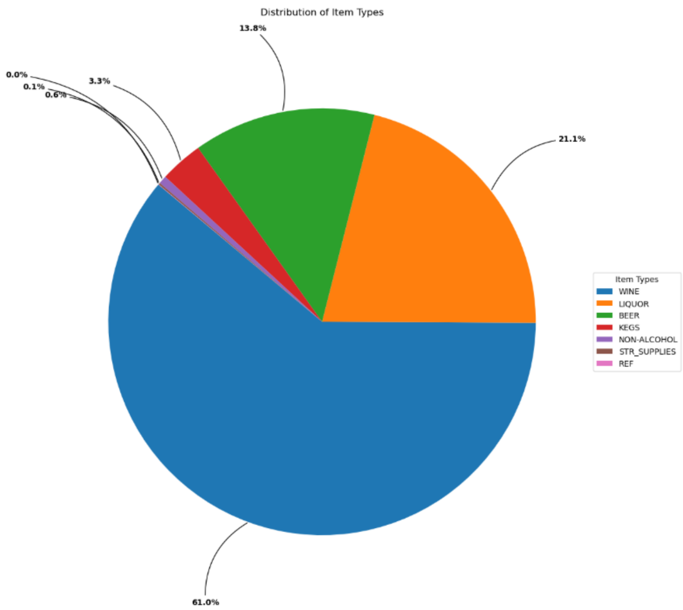

Figure 3.4: Pie chart showing percentage distribution of item types in the dataset.

Lastly, a boxplot was created to visualize the distribution of the total sales variable and to assess the presence of outliers (see Figure ??). The plot displays a heavily right-skewed distribution with a long upper tail, reflecting a small subset of extremely high-volume transactions. These likely correspond to promotional activities or large bulk orders. Their statistical leverage necessitated segmentation prior to modeling, as their inclusion in a single forecasting model would degrade generalizability.

These high-value outliers are not isolated anomalies; rather, they represent a distinct behavioral pattern that likely stems from promotional activities, bulk procurement orders, or seasonally induced demand surges. Their presence violates the assumption of homogeneity in the sales distribution, and if left unaddressed, such heterogeneity could introduce bias or reduce the generalizability of forecasting models trained on the combined dataset.

The boxplot, therefore, served as a critical diagnostic tool, not only for identifying the presence and extent of outlier values but also for informing one of the central preprocessing decisions in the overall methodology. By visually confirming the asymmetry and tail behavior of the sales distribution, the plot underscored the need for differential treatment of data subgroups, enhancing both the interpretability and performance of the downstream models.

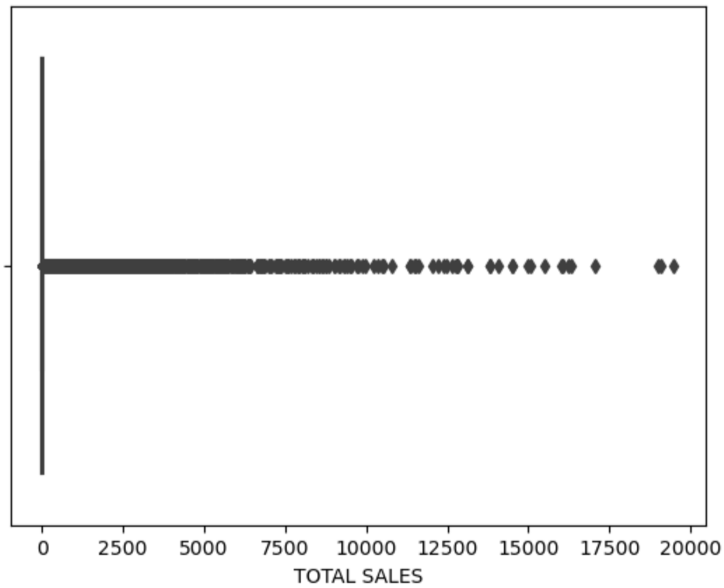

Figure 3.5: Boxplot of total sales indicating data skewness and outlier presence.

Together, these visual and statistical explorations provided a comprehensive understanding of the dataset's internal structure. The correlation analysis highlighted redundancies and dependencies among variables, while the temporal plots uncovered cyclical behavior and irregularities. The item-type analysis clarified the categorical composition of the dataset, and the distributional analysis of total sales informed key preprocessing decisions. This systematic understanding laid the groundwork for all downstream tasks including feature engineering, data segmentation, and model design.

3.3 Data Preparation

Effective data preparation was critical to ensure the reliability, accuracy, and stability of the forecasting models developed in this study. This stage comprised four sequential processes: data cleaning, structural integration, feature transformation, and categorical encoding. Each step was aimed at producing a coherent, numerically stable, and semantically meaningful dataset, ready for predictive modeling.

The process began with an initial quality assessment to identify missing or inconsistent entries. A programmatic summary revealed that the SUPPLIER column contained 167 missing values, while the ITEM TYPE column had one null entry. The

RETAIL SALES and RETAIL TRANSFERS columns contained three and one missing entries, respectively. All other fields—such as YEAR, MONTH, ITEM CODE, and WAREHOUSE SALES—were complete and required no intervention.

Given the limited scale and non-critical nature of missing values in the SUPPLIER column, a minimal-impact imputation strategy was employed. All null values in this field were replaced with the placeholder label "UNKNOWN", allowing the records to be retained without making any speculative assumptions. In contrast, the single null value in ITEM TYPE was not imputed but rather removed. This decision was made because ITEM TYPE plays a crucial role in both segmentation and categorical encoding; introducing an artificial category could have distorted the model's interpretability and performance.

With the missing data resolved, a new column titled TOTAL SALES was engineered by summing the values from the RETAIL SALES and WAREHOUSE SALES columns. This consolidated metric served as the primary target variable for modeling, representing the total monthly sales volume for each item across both retail and distribution channels.

Following this transformation, a validation step was performed to detect any records with negative values in the TOTAL SALES column. Negative sales entries are logically invalid in retail environments and may reflect refund adjustments, data entry errors, or accounting reversals. To maintain the integrity of the model training process, all such records were excluded. This cleaning stage preserved more than 99 percent of the dataset and resulted in a logically consistent, high-integrity dataset containing approximately 305,000 observations.

During exploratory analysis—visualized using a boxplot of TOTAL SALES (see Figure 3.5)—the distribution was found to be heavily right-skewed. Most sales transactions clustered in the low-to-medium range, while a significant long tail extended toward high-volume values. These outliers, likely driven by promotional events or bulk purchases, posed a risk to model generalizability. If left unaddressed, such extreme values could skew the model's learning process and reduce its ability to accurately forecast typical demand.

To systematically address the spikes observed in sales data, the Interquartile Range (IQR) method was employed. The IQR method is a robust, non-parametric statistical technique particularly useful for identifying outliers without relying on assumptions about the underlying data distribution[27].

The Interquartile Range (IQR) method relies on quartiles, which segment the dataset into four equal parts:

- **First Quartile (Q1)**: Marks the 25th percentile, below which 25% of the data falls.
- **Third Quartile (Q3)**: Marks the 75th percentile, below which 75% of the data falls.

The interquartile range itself, denoted as:

$$\text{IQR} = Q3 - Q1$$

captures the central 50% of the data. To identify outliers, boundaries are established using the following rules:

$$\text{Lower Bound} = Q1 - 1.5 \times \text{IQR}$$

$$\text{Upper Bound} = Q3 + 1.5 \times \text{IQR}$$

Any observation outside this range is classified as an outlier. This statistical technique effectively isolates anomalously high sales, which may result from promotional campaigns, bulk purchases, or other market anomalies.

In this project, the IQR method was specifically applied to the `TOTAL SALES` column. Records that exceeded the calculated upper bound were flagged as promotional transactions. These were extracted to form a separate `promo_sales` dataset, enabling the construction of a clean dataset that reflected normal sales behavior, devoid of promotional volatility.

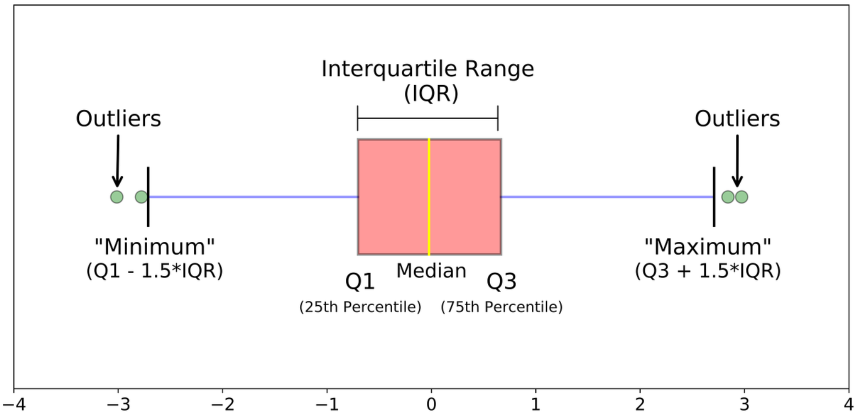

Figure 3.6: Annotated boxplot illustrating IQR bounds, quartiles, and detected outliers.

To further purify the dataset representing regular sales behavior, a second-pass IQR filtering was applied recursively. This additional step removed residual outliers potentially caused by input errors or localized market disturbances. The outcome was a stable dataset, well-suited for time series modeling and statistical analysis.

The final processed data was stored in two structured output files:

- `Normal_sales_data_with_features_final.csv` – containing statistically consistent baseline transactions.
- `Promo_sales_data_with_features.csv` – capturing high-volume sales attributed to promotional activity.

These datasets served as the input foundation for two independent modeling streams. The separation of demand regimes enabled each forecasting model to learn from its respective distribution without being biased by anomalies present in the other, thereby enhancing prediction accuracy and operational relevance in both regular and promotional contexts.

3.3.1 Feature Transformation

Following the construction of the TOTAL SALES target variable and the chronological structuring of the dataset, a suite of engineered features was developed to extract latent patterns and temporal dynamics inherent in retail sales data. The objective of this transformation process was not merely to increase the number of input variables, but to construct a semantically meaningful feature space that reflects key business phenomena—such as seasonality, recency effects, and historical performance baselines.

From the consolidated DATE field, several calendar-based attributes were derived. These included MONTH, QUARTER, WEEK OF YEAR, and DAY OF WEEK. These features were designed to capture cyclical consumer behaviors and temporal fluctuations in demand that often correspond to retail calendar events. In the context of alcoholic beverage sales, such seasonal effects are particularly pronounced. For example, the final quarter of the year often sees increased sales volumes due to holiday-related consumption and year-end promotions. By incorporating these time-based dimensions, the model gains the capacity to recognize and leverage predictable seasonal patterns.

To further account for autocorrelation in sales performance, lag features were engineered from the TOTAL SALES field. Specifically, lag values were calculated at one, three, and six-month intervals, resulting in the features SALES_LAG_1, SALES_LAG_3, and SALES_LAG_6. These were computed using a grouped shift operation over ITEM DESCRIPTION, ensuring that each product's historical trajectory was modeled independently. These lag features allow the forecasting algorithm to identify temporal dependencies and recurring demand behaviors based on recent trends, medium-term cycles, and semi-annual memory.

In addition to lagging, a short-term smoothing feature was introduced using a three-month rolling mean of total sales, denoted as SALES_ROLLING_MEAN_3. This rolling average serves as a noise reduction mechanism that stabilizes volatile sales inputs while preserving underlying signals. It is especially useful in the promotional sales dataset, where high-amplitude spikes are common. The rolling mean helps to moderate the impact of such anomalies and supports better generalization in forecasting models.

To introduce a long-term reference benchmark, a historical sales baseline was constructed for each product. A new feature, MEDIAN_PRODUCT_SALES, was created by calculating the median total sales for each product across the full time span of the dataset. This allows each month's sales to be evaluated in the context of a product's historical performance. For example, a current sales figure may be interpreted as above-average, average, or below-average relative to the product's overall history. This form of benchmarking aids the model in distinguishing between regular and exceptional performance and enhances interpretability in business decision-making.

Collectively, these transformations enriched the dataset with informative, domain-relevant features that improved both predictive power and operational transparency. By embedding seasonal signals, temporal dependencies, and performance baselines, the dataset was transformed into a high-quality feature matrix, well-suited for supervised learning.

3.3.2 Categorical Encoding

The dataset included two key categorical variables—ITEM TYPE and SUPPLIER—each holding strategic importance in the modeling framework. These variables were encoded using techniques appropriate to their structure and cardinality, ensuring that the forecasting models could extract meaningful distinctions without introducing unnecessary dimensionality or interpretive bias.

The ITEM TYPE variable, which contains a manageable number of distinct product categories (e.g., wine, beer, liquor), was processed using one-hot encoding. This transformation created a set of binary indicator columns corresponding to each category, preserving the nominal nature of the variable and enabling the model to assess the independent contribution of each item type to overall sales behavior. One-hot encoding was especially critical for capturing seasonal trends associated with different alcohol types. For instance, increased beer consumption in summer or a surge in wine purchases during the holiday season are domain-specific effects that can be effectively modeled using category indicators.

In contrast, the SUPPLIER field presented high cardinality, with numerous unique supplier names across the dataset. Applying one-hot encoding in this context would have created excessive dimensionality and potential sparsity. To address this, label encoding was employed. Each supplier was assigned a unique integer identifier, resulting in a compact single-column representation. This approach is particularly compatible with tree-based algorithms like XGBoost, which can effectively partition decision paths without assuming ordinal relationships between encoded values. Label encoding also allows the model to detect supplier-specific sales trends, differences in distribution reach, and varying promotional intensities across vendors.

Crucially, both encoding procedures were applied uniformly to the normal and promotional sales datasets. This ensured alignment in the feature space across both modeling streams, enabling consistent interpretation and performance evaluation. The standardized encoding approach also allowed the comparative analysis of model behaviors under distinct sales regimes while maintaining the same structural basis.

With the application of feature transformation and categorical encoding complete, the dataset was fully structured and enriched with historical, temporal, and categorical insights. This finalized dataset was then used as input for the model training phase, forming a robust foundation for accurate and interpretable sales forecasting across regular and promotional contexts.

3.4 Modelling

In this study, the predictive modeling task was approached using Extreme Gradient Boosting (XGBoost)—a highly efficient and scalable machine learning algorithm known for its performance in structured data environments. XGBoost belongs to the class of ensemble methods, meaning it combines multiple simpler models (in this case, decision trees) into a powerful aggregate model. It builds each tree sequentially so that the new tree aims to correct the errors made by the previous ones. This technique, known as boosting, enhances model accuracy by focusing learning on examples that are harder to predict.

XGBoost is particularly well-suited for tabular datasets, such as the one used in this retail sales forecasting project. Sales data exhibit non-linear behavior influenced by many interacting variables—such as time of year, supplier identity, and past performance trends—which simpler linear models cannot capture. Compared to alternatives like Random Forests or Support Vector Machines, XGBoost provides advanced regularization, optimized handling of sparse or missing values, and a faster training pipeline[27]. While neural networks and LSTM architectures are often used for time series tasks, they require more computational resources and are less interpretable, which made them less desirable for this context where stakeholder interpretability and deployment simplicity were equally important.

The theoretical backbone of XGBoost lies in gradient boosting, where the goal is to minimize a loss function by adding new models that estimate the residuals of prior models. The process begins with an initial prediction (usually the mean), and at each iteration, a new decision tree is trained to predict the errors (residuals) of the model so far. Each new tree's contribution is scaled by a learning rate, and the ensemble continues growing until the error stops improving. The final model is the sum of all individual tree predictions.

Formally, the prediction for a data point is computed as the sum of trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

Here, each f_k represents a regression tree. The objective function that XGBoost seeks to minimize consists of two parts: a training loss function that measures how well the model fits the data, and a regularization term that penalizes complexity:

$$\mathcal{L}(\phi) = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k)$$

The loss function l typically uses mean squared error, and the regularization function Ω controls model complexity, where:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

In this equation, T is the number of leaves in the tree, w_j are the weights assigned to each leaf, and γ and λ are regularization parameters. To efficiently optimize this objective, XGBoost applies a second-order Taylor expansion of the loss function. This approach allows it to leverage not only the gradient (first derivative) but also the curvature (second derivative), resulting in faster and more accurate convergence:

$$\mathcal{L}^{(t)} \approx \sum_{i} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \Omega(f_t)$$

Here, g_i and h_i are the gradient and Hessian of the loss with respect to prediction $\hat{y}_i$. This formulation gives XGBoost its name, extreme gradient boosting, by enhancing classical gradient boosting with second-order optimization and regularization.

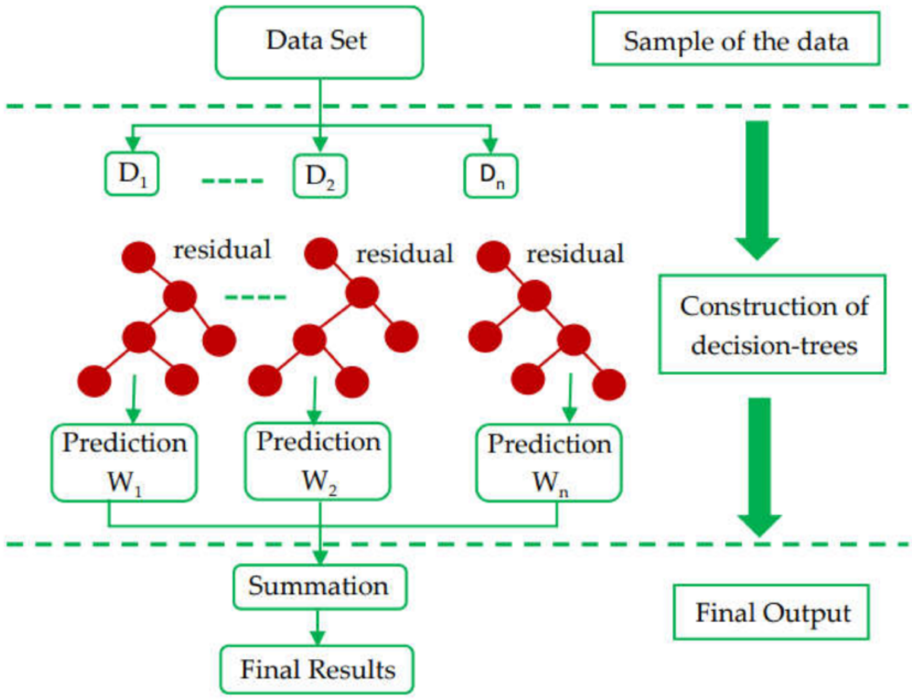

Figure 3.7: Visual summary of sequential learning mechanism in XGBoost.

The diagram above (Figure 3.7) visually summarizes the sequential learning mechanism of the XGBoost algorithm. Each new decision tree is trained to predict the residual errors, the differences between the actual and predicted values, from the ensemble of previous trees. These residuals represent areas where the model is currently underperforming, allowing subsequent trees to focus on the most challenging data points. As shown, this additive process continues iteratively, with each prediction contributing a weighted component $(W_1, W_2, \ldots, W_n)$ to the final forecast. By adding the outputs of all weak learners, the model forms a strong composite predictor. This step-by-step refinement of predictions, guided by both gradients and Hessians of the loss function, is what gives XGBoost its name, Extreme Gradient Boosting, and underpins its high accuracy and efficiency, especially on structured tabular datasets like the one used in this study.

For this project, XGBoost was used in a multi-output regression framework, which allowed the system to forecast three consecutive months of future sales ($t+1$, $t+2$, and $t+3$). This was achieved using the `MultiOutputRegressor` wrapper from Scikit-learn, which creates a separate XGBoost model for each forecasting horizon but trains them in parallel with the same feature inputs. This structure is highly useful for business scenarios where short- to mid-term forecasts are required simultaneously for planning and inventory control.

The internal structure of the model training process can be understood by visualizing it as a data-to-prediction pipeline. Initially, feature-engineered datasets—segregated into promotional and normal sales subsets—are passed through preprocessing layers. These include encoding categorical fields like supplier names, generating lagged variables, and applying rolling average calculations. Once features are ready, they are fed into the dual XGBoost models, each dedicated to one sales regime.

3.4.1 Hyperparameter Tuning and Optimization

To ensure optimal performance of the XGBoost models, a systematic hyperparameter tuning procedure was conducted using the GridSearchCV module from Scikit-learn. The tuning process was designed to balance model complexity with generalization capability and to mitigate overfitting—particularly given the distinct statistical properties of the normal and promotional sales datasets.

A parameter grid was defined across three critical hyperparameters: the maximum depth of trees, the learning rate, and the number of boosting iterations. These parameters were selected due to their direct impact on the model's capacity to learn non-linear patterns, regulate convergence behavior, and control ensemble size. A 3-fold cross-validation framework was applied to each parameter combination to assess model performance across multiple training splits and enhance robustness against sample bias.

The tuning process was conducted separately for the normal and promotional models to accommodate the unique variance structures and data distributions of each regime. This approach allowed each model to be optimized according to its specific forecasting context.

Once the optimal hyperparameter set was identified for each segment, the models were trained using these configurations in the final learning phase. This ensured alignment between the tuning environment and the deployment-ready model. The incorporation of tuned hyperparameters also supported improved bias-variance balance and reduced the risk of model instability.

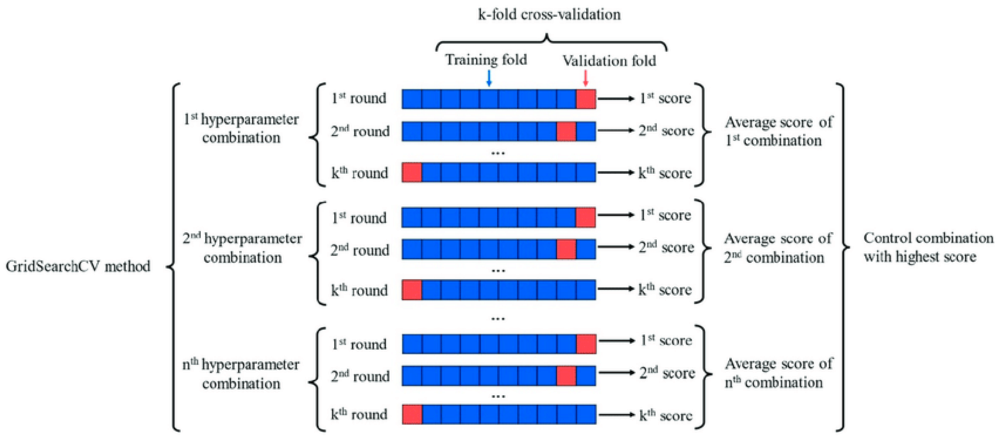

Figure 3.8: XGBoost-based sales forecasting framework.

At each training iteration, the model calculates the gradient and Hessian of the current loss, fits a tree to minimize that residual error, applies regularization to prune the tree and avoid overfitting, and updates the ensemble prediction. After a set number of trees are built (determined by n estimators), the model stops training and is evaluated on withheld test data.

3.4.2 Model Training Workflow

The complete model development process was implemented as a modular pipeline designed to ensure consistency, scalability, and separation of concerns across the forecasting task. This pipeline comprised a sequence of data transformation steps, model-specific training routines, and performance tracking mechanisms.

As illustrated in Fig. 4.1, the pipeline begins with structured input data for two distinct demand regimes: normal and promotional sales. Both datasets were preprocessed using a unified feature engineering framework that incorporated lag variables, rolling averages, temporal encodings, and categorical transformations, as detailed in Chapter 3.

Each preprocessed dataset was then passed into a dedicated instance of the XGBoost regressor. These two parallel training streams allowed the models to learn from distinct patterns within their respective regimes—stable cyclic demand in the normal sales data and high-variance behavior in the promotional data.

Hyperparameter tuning, described in the previous section, was conducted independently for each model stream prior to final training. This ensured that the configurations used during learning were specifically optimized for the statistical properties of each subset.

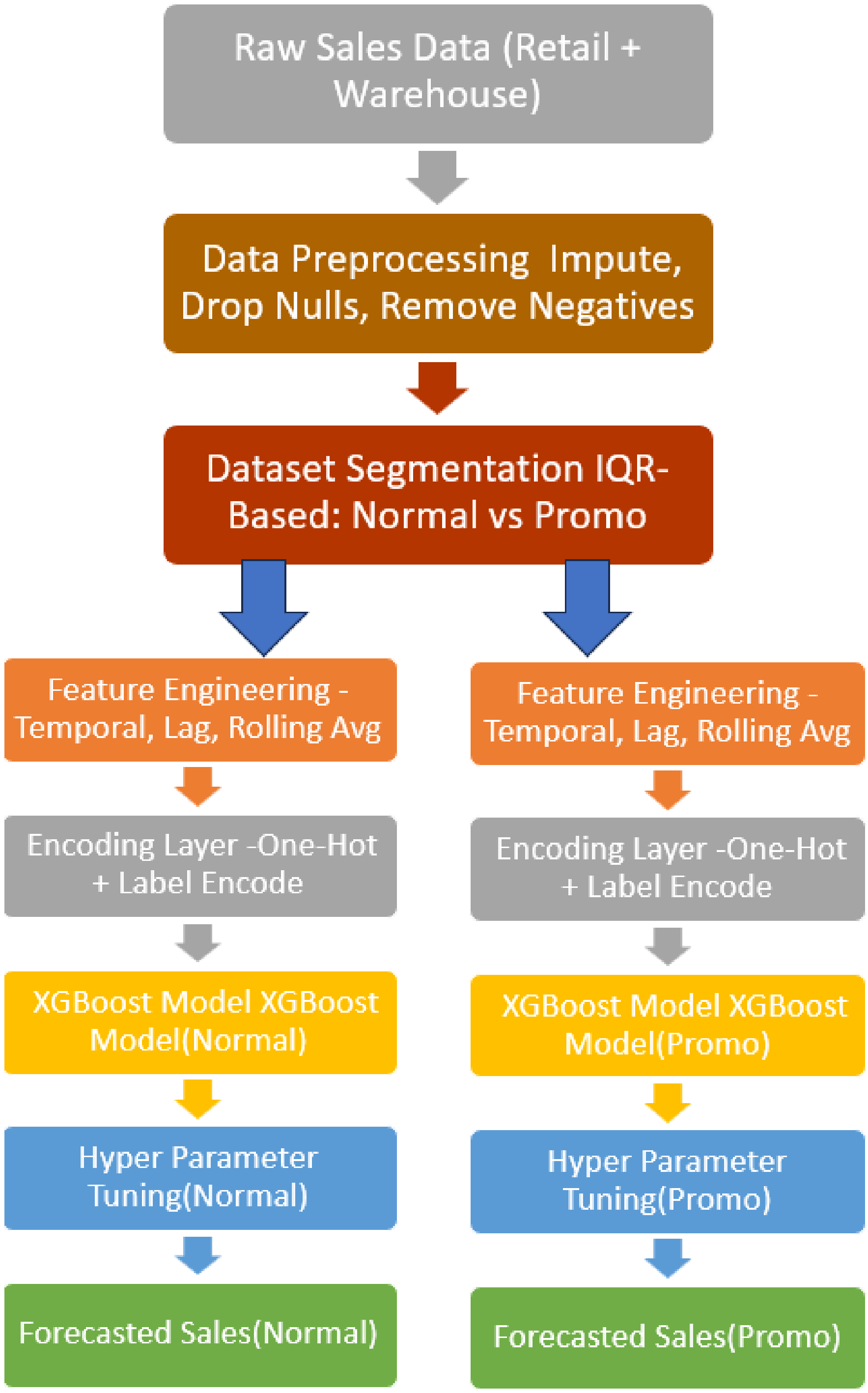

Figure 3.9: XGBoost-based sales forecasting framework.

In the dual-model design, the same framework is executed independently on the normal and promotional sales datasets, as illustrated in the complete system block diagram in the previous the training process followed the gradient boosting paradigm, where decision trees were sequentially added to correct residual errors from previous iterations. Internal regularization was applied during tree construction to improve generalization. After training, the models were serialized and evaluated on their respective test splits using a consistent metric framework, although performance results are discussed separately in the results section.

This architecture supports modular retraining, enabling each model to be updated independently as new data becomes available. Additionally, the dual-model design facilitates the integration of regime-specific strategies in retail forecasting systems, such as differentiated inventory planning or promotional campaign evaluation.

Ultimately, XGBoost was selected and implemented because it balances interpretability, computational speed, and predictive power. It outperformed baseline models like linear regression and naive forecasting, offered deeper insight into feature importance, and required less computational overhead than recurrent neural networks (RNNs) or LSTM models, which often demand more data and longer training cycles. Furthermore, XGBoost' s explainability through feature importance rankings helped tie forecasting results back to business factors such as supplier influence, recent sales history, and calendar timing—making the model not only accurate but actionable[1].

3.4.3 Challenges in Deep Learning Model Application: Data Sparsity and Irregularity

Prior to finalizing the use of gradient boosting trees for forecasting, multiple deep learning models were investigated, including Long Short-Term Memory (LSTM) networks, DeepAR, and N-BEATS[28, 29, 30]. These architectures have achieved state-of-the-art results in time series forecasting tasks across several industries. However, in the context of this project, their application presented significant practical and technical obstacles due to the specific structure and quality of the dataset.

The dataset consisted of monthly-level retail and warehouse sales for individual products spanning from 2017 to 2020. However, as illustrated in Figure 3.10, the temporal distribution of data was highly irregular. Large numbers of months contained either zero sales or missing values for many product lines. This was not simply noise; rather, it reflected a genuine business condition—many products (identified by ITEM CODE) were not present in all months. Products were launched, phased out, or temporarily unavailable depending on supplier activity, regulatory periods, or seasonal demand.

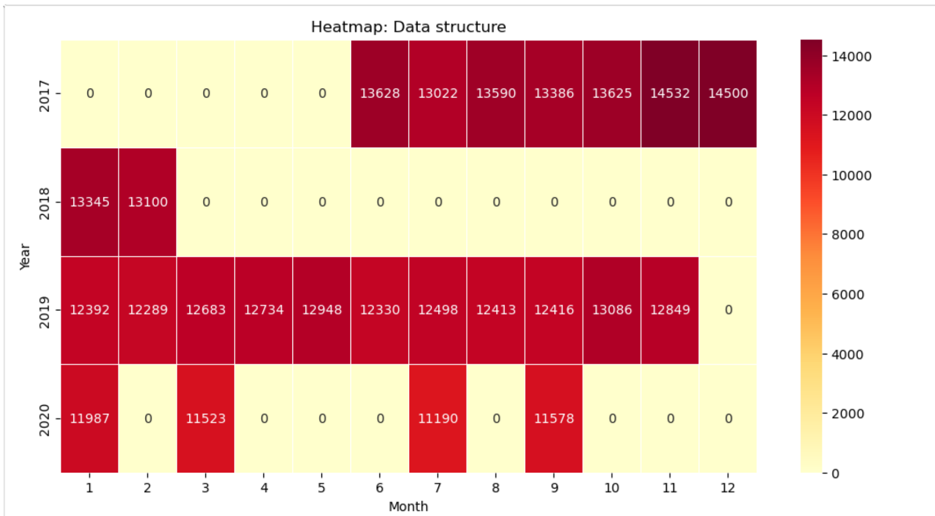

Figure 3.10: Heatmap of monthly sales activity showing data sparsity.

This irregularity severely hampered the use of sequence-based neural models. Deep learning approaches like LSTM rely on the assumption of consistent time-step sequences, where a product's historical sales are recorded without interruption over fixed intervals. In this case, most products had fragmented sales histories. Some appeared only in specific quarters or years. As a result, training LSTMs required either aggressive imputation—risking artificial trends—or zero-padding, which led to unstable learning and convergence failure[24]. Even with hyperparameter tuning and early stopping strategies, the models failed to capture generalizable patterns, especially for products with intermittent presence.

Similarly, DeepAR, a probabilistic RNN-based model designed for joint forecasting of multiple series, could not be successfully applied[29]. DeepAR assumes that each item has a sufficiently long and dense history, and that item identities remain consistent across time[26]. However, in this project, many ITEM CODE values were not consistently recorded. Some items disappeared from the dataset for multiple consecutive months and reappeared later under slightly modified descriptions. These structural inconsistencies broke the assumptions required by DeepAR's training loop, which expects a uniform tensor representation of all-time series.

A similar outcome occurred with N-BEATS, a model that uses deep neural blocks to forecast based on trend and seasonality decomposition. Despite its capacity to model non-recurrent series, N-BEATS requires complete fixed-size input windows for backcasting and forecasting. Because many items had sparse sales entries (fewer than 12–18 usable monthly data points), the model either rejected sequences during training or generated highly volatile forecasts due to limited context. Even after filtering for products with sufficient time length, the network overfit easily due to small batch sizes and insufficient diversity across products.

The inconsistent availability of item-level sales data also posed difficulties in preparing data loaders and batching functions, which are mandatory in deep learning frameworks. Many libraries (including PyTorch Forecasting and GluonTS) failed to generate training sets without manual padding, interpolation, or heavy preprocessing—all of which introduced additional complexity without commensurate gains in accuracy.

Moreover, the computational demands of these neural models far exceeded practical feasibility. Training times were significantly longer than tree-based models, required GPU acceleration, and still yielded unsatisfactory results due to the limitations described above. Visual inspection of predictions revealed random fluctuations and convergence collapse in the final layers, indicating that the models could not learn coherent patterns from such fragmented sequences.

Because of these compounding limitations—particularly the irregular appearance of ITEM CODE values over time, missing sales periods, and sequence sparsity—these deep learning models were deemed unsuitable for this forecasting task. In contrast, XGBoost offered a robust, interpretable, and resource-efficient alternative. It does not require sequential input, can handle missing values directly, and is capable of learning from tabular features such as lagged sales, calendar indicators, and supplier identifiers—even when they are incomplete. It was able to learn generalizable trends across normal and promotional sales regimes, making it the optimal solution both technically and operationally.

3.5 Evaluation Metrics and Forecasting Principles

To evaluate the effectiveness and generalizability of the XGBoost models developed in this study, three well-established regression metrics were used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). These metrics collectively provide a comprehensive measure of how accurately the models predict unseen sales values, each offering a unique perspective on the magnitude, distribution, and variance explained by prediction errors.

Mean Absolute Error (MAE)

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

Mean Absolute Error (MAE) quantifies the average difference between predicted and actual sales values across all test observations. Because it treats all errors equally regardless of direction, MAE offers an intuitive interpretation of prediction reliability. A lower MAE indicates that, on average, the model's predictions are closer to actual outcomes. It is particularly useful in business applications where cumulative over- or under-prediction must be minimized.

Root Mean Squared Error (RMSE)

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

Root Mean Squared Error (RMSE), in contrast, penalizes larger errors more heavily by squaring the differences before averaging them. This makes RMSE more sensitive to outliers or extreme mispredictions. In retail forecasting, RMSE is valuable for understanding worst-case scenarios—such as major mismatches during promotions or supply surges—because such deviations may have a greater operational cost.

R Squared Score (Coefficient of Determination)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

The R^2 Score provides a normalized measure of how well the independent variables in the model explain the variance in the target variable. Ranging between 0 and 1, an R^2 value near 1 implies that nearly all the variance in the sales data is captured by the model predictions. An R^2 close to 0, conversely, suggests that the model fails to explain the variability and performs no better than a simple average baseline. This metric is essential when assessing the proportion of explainable demand patterns learned by the model.

All models were trained and tested using a time-aware data split that preserved chronological order, thereby avoiding data leakage and better reflecting real-world forecasting conditions. The evaluation was conducted independently for both the normal and promotional sales models, and primarily focused on single-step performance to allow for an isolated assessment of each model's core accuracy. These evaluation metrics formed the basis for the evaluation of model performance, with detailed results presented in the following section.

Chapter 4

Results and Analysis

4.1 Introduction

This chapter presents the evaluation of the machine learning models developed to forecast item-level sales for regular and promotional retail demand. The objective is to assess the models' predictive accuracy, understand the contribution of key features, and interpret the business implications of forecast outputs. Two separate XGBoost regressors—one trained on regular sales data and the other on promotional sales data—were evaluated using time-aware testing strategies. This separation allows each model to specialize in the dynamics of its respective regime.

The performance evaluation comprises metric-based analysis, forecast comparisons across multiple time horizons, feature importance interpretation, and product-supplier-level behavioral profiling. The insights derived provide a holistic understanding of model behavior and its relevance to strategic retail planning.

4.2 Normal Sales Model Performance

The XGBoost model trained on the regular (non-promotional) sales dataset demonstrated strong predictive accuracy, confirming the appropriateness of gradient boosting methods for structured retail forecasting tasks. Training was conducted using a curated feature set comprising lagged sales variables, calendar-based indicators, and rolling aggregate features, all derived from a cleaned and statistically stable dataset free from promotional anomalies.

Evaluation on the hold-out test set produced low forecast error values, indicating precise generalization across time. As presented in Table 5.1, the model achieved a Mean Absolute Error (MAE) of 0.01486, a Root Mean Squared Error (RMSE) of 0.03333, and a Coefficient of Determination ($\mathbf{R^2}$) of 0.99985. These metrics indicate that the model's predictions deviated by less than 0.04 units from actual monthly sales values on average and that over 99.98

Table 4.1: Forecast Performance: Normal Sales Model

Metric	**Value**
MAE (Mean Absolute Error)	0.01486
RMSE (Root Mean Squared Error)	0.03333
R^2 Score (Coefficient of Determination)	0.99985

Such strong performance is attributable to the stationarity and predictable seasonality of the normal sales data. Unlike promotional sales, normal transactions follow repeating monthly and quarterly patterns driven by habitual consumer behavior and replenishment cycles. The use of lagged sales features allowed the model to effectively capture short-term memory, while rolling averages and time-based encodings modeled broader temporal dynamics.

To visually validate this performance, a scatter plot of predicted versus actual sales values for the test period is presented in Figure 4.1. The plot shows that the predicted values cluster closely around the ideal diagonal line $y = x$, indicating minimal deviation and no observable systematic bias. Errors are randomly distributed, confirming that the model generalizes well across the full range of sales volumes.

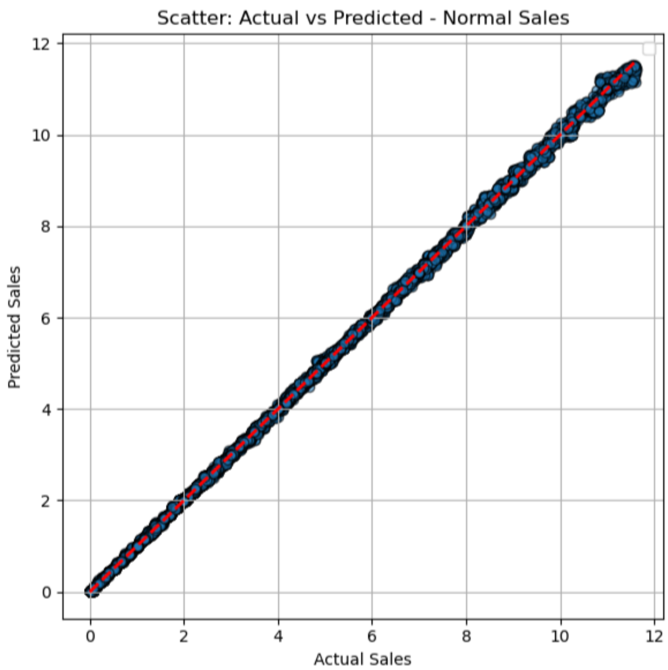

Figure 4.1: Actual vs Predicted Sales (Normal Sales)

This performance is attributed to the inherent regularity of normal sales patterns, which exhibit stable seasonal and cyclical trends. Unlike promotional transactions, which are sporadic and influenced by external campaigns, normal sales follow consistent monthly and quarterly rhythms driven by habitual consumption and standard replenishment intervals. The inclusion of lag features enabled the model to capture short-term temporal dependencies, while rolling means and calendar encodings supported the modeling of broader seasonal cycles.

To visually assess prediction quality, a scatter plot of actual versus predicted values for the test set is presented in Fig. 5.1. The predicted values lie closely along the reference line y=xy = xy=x, with no apparent structural bias or heteroscedasticity. This alignment suggests that forecast errors are symmetrically distributed across the prediction range, supporting the reliability of the model across different product categories and sales volumes.

The consistency and accuracy of the normal sales model make it well-suited for integration into key operational systems, including inventory optimization, warehouse replenishment scheduling, and procurement planning. Its ability to deliver high-fidelity forecasts under stable demand conditions provides a reliable baseline for tactical decision-making in day-to-day retail operations.

4.3 Promotional Sales Model Performance

The promotional sales forecasting model was developed using a statistically segmented subset of the dataset, isolated through Interquartile Range (IQR) analysis to capture high-volume outlier transactions. These observations typically correspond to retail activity influenced by promotional campaigns, supplier-driven discounts, or seasonal marketing events. Due to their episodic nature and dependence on external variables not captured in the dataset (e.g., discount percentage, advertisement exposure), promotional sales inherently exhibit greater volatility and modeling complexity.

Despite these challenges, the XGBoost model trained on the promotional sales segment demonstrated strong predictive performance. Evaluation on the test set yielded a Mean Absolute Error (MAE) of 21.82, a Root Mean Squared Error (RMSE) of 114.85, and an R^2 score of 0.9734. These metrics are summarized in Table 4.2. The model thus explains approximately 97.34% of the variance in promotional sales volumes, an improvement over earlier iterations and a strong result considering the unavailability of exogenous campaign metadata.

Table 4.2: Forecast Performance: Promotional Sales Model

Metric	**Value**
MAE (Mean Absolute Error)	21.82
RMSE (Root Mean Squared Error)	114.85
R^2 Score (Coefficient of Determination)	0.9734

The substantially lower RMSE compared to earlier models indicates the refined model's improved ability to handle peak sales anomalies without introducing excessive prediction variance. Meanwhile, the reduced MAE suggests that the model's forecasts remain close to actual observed values for the majority of cases. The high R^2 score further confirms that the model captured key structural characteristics of promotional demand behavior, even in the absence of campaign-specific features.

Figure 4.2 illustrates the relationship between actual and predicted promotional sales. While a wider dispersion of points is evident compared to the normal sales model—reflecting the naturally higher variability in promotion-driven demand—predicted values generally follow the reference line $y = x$, indicating the model's directional consistency and minimal bias across the prediction space.

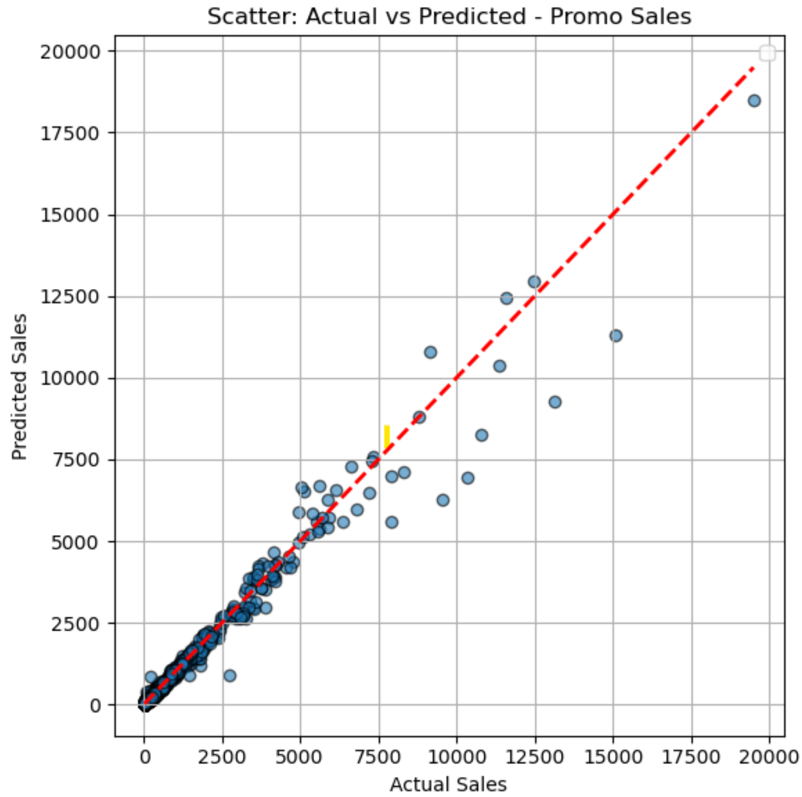

Figure 4.2: Figure 5.2: Actual vs Predicted Sales (Promotional Sales)

The practical implications of this model are significant. Accurate forecasts of promotional demand enable inventory pre-positioning, logistics surge preparation, and supplier coordination ahead of high-traffic periods. Additionally, the model outputs can contribute to campaign performance evaluation, ROI benchmarking, and the simulation of future uplift scenarios to inform promotional strategy design.

In summary, although forecasting promotional sales remains a complex task due to their sporadic, externally driven nature, the updated XGBoost model delivered high predictive accuracy and operationally useful insights. These results validate the segmentation approach employed in this study and underscore the value of machine learning models in managing volatility within retail forecasting systems.

4.4 Feature Importance Analysis

A notable advantage of using the XGBoost algorithm lies in its ability to provide interpretability through internal diagnostics such as feature importance scoring. These metrics offer insight into the relative contribution of each input variable by quantifying how frequently, and with what effect, a feature is used to improve decision-making within the model's ensemble of decision trees.

In this study, feature importance was quantified using the F-score, a built-in XGBoost metric that counts the number of times a feature is selected to split the dataset during tree construction. Each split that includes a feature contributes to the model's performance by reducing prediction error. Therefore, a higher F-score indicates greater frequency and cumulative value in enhancing predictive accuracy.

The top four most influential features in the normal sales model are visualized in Figure 4.3. These include `SALES_LAG_3`, `SALES_ROLLING_MEAN_3`, `SALES_LAG_1`, and `MONTH`. The dominance of time-series features reaffirms the importance of temporal dependencies and seasonal cycles in shaping regular retail demand.

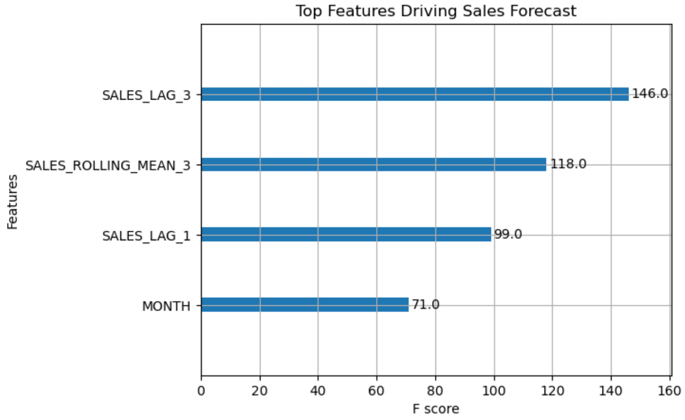

Figure 4.3: Figure 5.5: Feature Importance — Normal Model

The feature `SALES_LAG_3` recorded the highest F-score (146), indicating that historical sales from three months prior were most frequently selected by the model to partition data and improve forecasts. This finding suggests that consumer purchasing behavior in the dataset follows cyclical patterns, potentially tied to quarterly restocking habits or repeating demand intervals.

The `SALES_ROLLING_MEAN_3` feature, with an F-score of 118, captures smoothed short-term trends by averaging sales across the preceding three months. This aggregation helps the model distinguish between structural movements in demand versus random fluctuations. In stable sales environments, rolling statistics enhance forecast stability by filtering out transient noise.

`SALES_LAG_1`, the one-month lag feature, achieved an F-score of 99. This feature represents recent sales momentum and supports the model in recognizing immediate trends—such as product surges or declines—that might influence near-future behavior. Its predictive value underscores the autoregressive nature of many consumer purchasing patterns.

Finally, the `MONTH` variable, with an F-score of 71, reflects the model's sensitivity to calendar-based seasonality. The presence of this feature among the top contributors confirms that monthly trends—such as holiday effects or off-season slowdowns—play a measurable role in shaping regular retail demand.

It is important to note that the F-score reflects usage frequency and split utility rather than causal influence or predictive weight in any single forecast. A feature may have a high F-score due to consistent moderate usefulness across many trees, even if its effect on any individual prediction is marginal.

Interestingly, supplier identifiers and item-specific descriptors were not among the top-ranking features in the normal model. This reinforces the conclusion that regular sales behavior is governed more by internal sales cycles and temporal dynamics than by brand-level marketing or supplier-specific variability. This contrasts with the promotional sales model, where supplier-related attributes are expected to play a more influential role due to campaign targeting and event-specific promotions.

In summary, the feature importance analysis validates the project's feature engineering approach and demonstrates that the XGBoost model effectively leveraged lag variables, rolling statistics, and seasonal encodings to achieve high accuracy in normal sales forecasting. These findings not only support the robustness of the modeling pipeline but also offer operational insight for business units: tracking and maintaining high-quality data in key time-series dimensions is essential for sustaining reliable predictive performance.

4.5 Multi-step Forecast Behavior

A critical requirement for practical retail forecasting systems is the ability to generate multi-step predictions—that is, forecasts not only for the immediate future but also for subsequent time periods. This capability supports extended planning in areas such as supply chain logistics, inventory allocation, and promotional scheduling. In this study, both the normal and promotional XGBoost models were designed to predict sales volumes over a three-month rolling horizon: Month $t+1$, Month $t+2$, and Month $t+3$.

Technically, multi-step forecasting was implemented using a multi-output regression approach. The base XGBoost model was wrapped with Scikit-learn's `MultiOutputRegressor`, enabling simultaneous prediction of multiple target variables using a shared input feature space. This method mitigates the error compounding issue often associated with recursive strategies, where prior forecasts are used as inputs for future predictions. Instead, each output step is treated independently within the same learning cycle, preserving model consistency and efficiency.

Model Consistency Across Horizons

The multi-step prediction performance of the normal sales model exhibited strong consistency across all horizons. The Root Mean Squared Error (RMSE) increased slightly from 0.03333 at Month $t+1$, with marginal growth at $t+2$ and $t+3$, while the R^2 score remained consistently above 0.9995. This stability indicates that the model effectively leveraged lagged features and rolling means to generalize future sales behavior, even with increasing temporal distance from the training window.

The promotional sales model, which operates under more volatile conditions, also maintained usable predictive accuracy across horizons. The R^2 value for $t+1$ was 0.9745, decreasing slightly for $t+2$ and $t+3$, though remaining above 0.97 throughout. This trend reflects the inherent difficulty in projecting promotional spikes, particularly when such events are externally triggered and not explicitly encoded in the feature set. As expected, RMSE increased as the forecast horizon extended, indicating growing uncertainty in extreme event estimation.

Visualizing Forecast Trajectories

To evaluate forecast behavior over time, predicted sales values were plotted against actual observations for each forecast horizon. These plots are presented in Figure 4.4 and Figure 4.5 for the normal and promotional models, respectively.

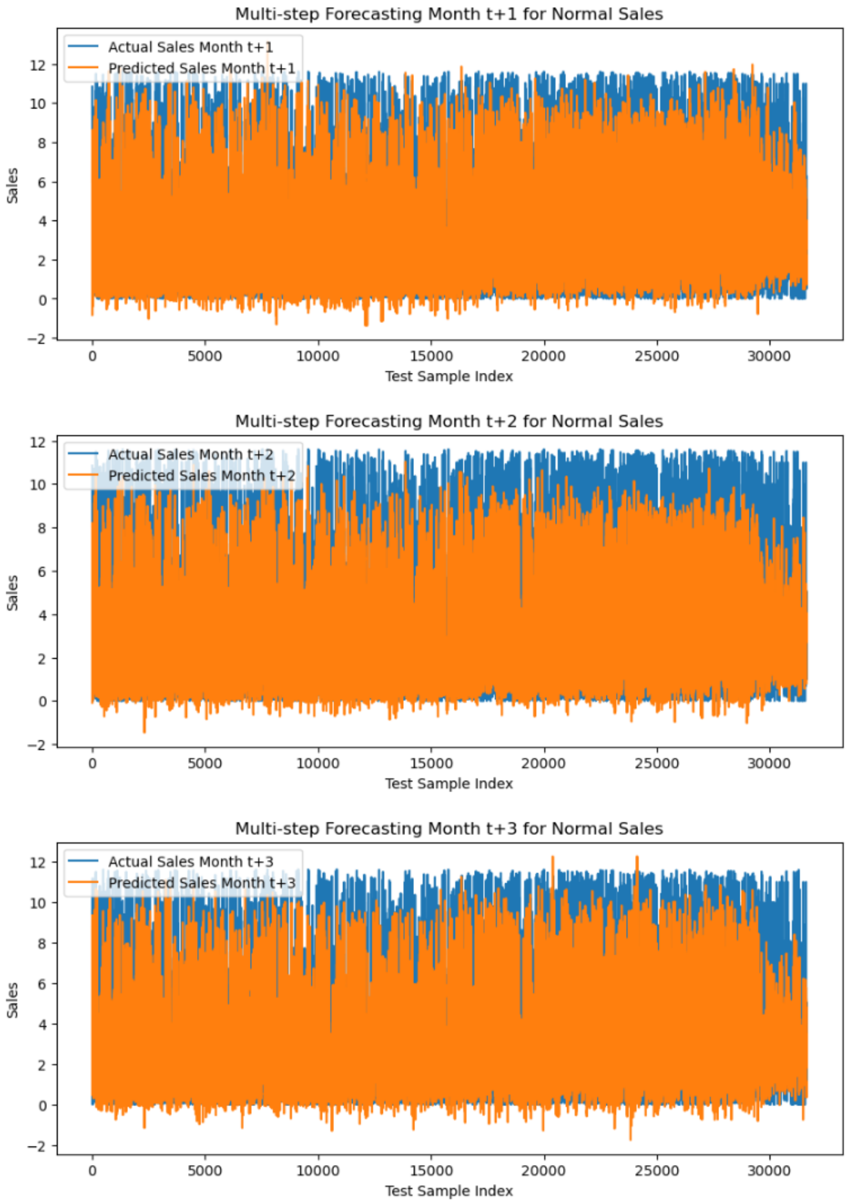

Figure 4.4: Multi-step Forecast Trajectories – Normal Sales Model

The forecast trajectories in Figure 4.4 demonstrate high alignment between predicted and actual values across all horizons. Forecasted curves follow the actual demand with minimal lag, capturing both trend direction and magnitude effectively. These results confirm the model's ability to learn stable cyclical and autoregressive patterns in normal sales data, making it well-suited for monthly operational forecasting and strategic planning.

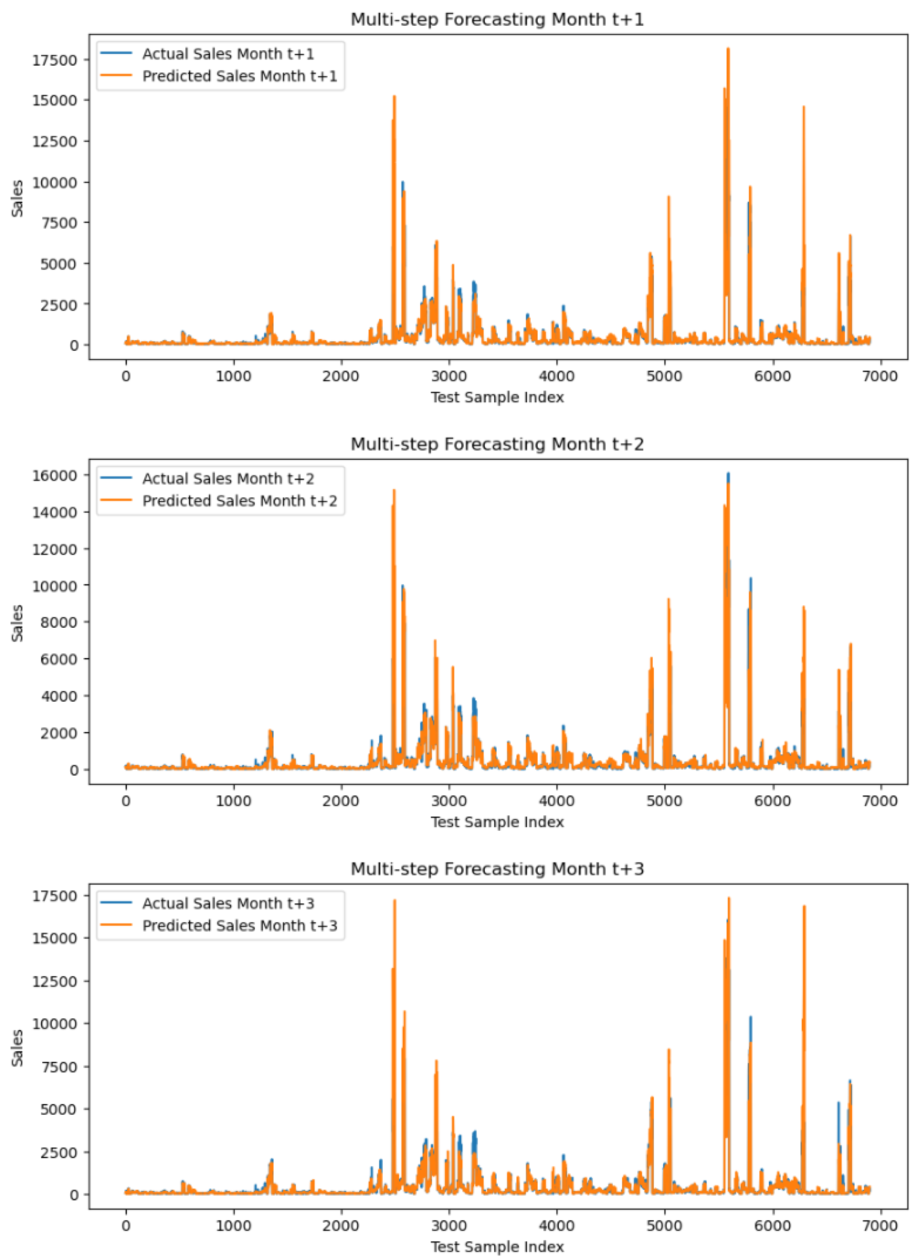

Figure 4.5: Multi-step Forecast Trajectories – Promotional Sales Model

This figure presents three line plots comparing actual and predicted sales for Month $t+1$, $t+2$, and $t+3$ using the promotional sales model. The model captures general trend directions and correctly anticipates several high-volume events, though slight lag and underestimation are visible for sharp spikes. As the horizon increases, prediction variance rises, which is expected due to the irregular nature of promotional activity.

In contrast, Figure 4.5 illustrates a more variable prediction pattern for the promotional model, particularly at months $t+2$ and $t+3$. The model captures general trends and anticipates several high-volume events but exhibits smoothing behavior during extreme spikes and dips. This is attributable to the absence of explicit campaign metadata (e.g., discount levels, marketing timing), which limits the model's visibility into external promotion drivers. Nevertheless, the model's ability to capture directional trends remains valuable for mid-range promotional planning, inventory risk buffering, and seasonal resource allocation.

In summary, the multi-step forecast evaluation demonstrates that the XGBoost models retain predictive power and stability across forward time intervals, with the normal sales model achieving near-linear extrapolation and the promotional model providing trend-aware but uncertainty-limited guidance[25]. These findings support the feasibility of using segmented machine learning pipelines for rolling-horizon retail planning.

Drift, Error Propagation, and Practical Implications

A common challenge in multi-step forecasting is the risk of prediction drift, where forecast errors accumulate over time. This phenomenon is particularly problematic in recursive forecasting approaches, which use the model's own predictions as inputs for subsequent time steps, often leading to compounding inaccuracies and deviation from actual values.

To mitigate this risk, the models in this study employed a parallel forecasting strategy, where each future horizon—$t+1$, $t+2$, and $t+3$—was treated as an independent regression target. All forecast steps were trained simultaneously using the same set of lagged, rolling, and calendar-based features, thereby avoiding reliance on previously predicted values. This structure ensures that each prediction benefits from a stable and consistently engineered input space, preserving accuracy across multiple horizons.

Although some degradation in performance is expected as the forecast horizon extends, the models demonstrated controlled drift. In the normal sales model, error metrics such as RMSE and MAE increased only incrementally across the three-month window, and the R^2 values remained above 0.9995, indicating minimal loss of predictive fidelity. In the promotional model, where high-variance events such as supplier-led campaigns introduce irregularities, forecast quality declined more rapidly. However, even at extended horizons, model performance remained within acceptable error bounds and continued to capture trend directionality.

From an operational standpoint, the availability of rolling multi-step forecasts supports a range of business functions:

- The $t+1$ horizon serves immediate planning needs such as order quantities, replenishment schedules, and logistics coordination.
- The $t+2$ and $t+3$ projections inform more strategic planning, including promotional calendar design, financial forecasting, and supplier negotiations.
- Collectively, the three-step output facilitates 90-day outlooks for retail demand, enabling both tactical responsiveness and longer-term alignment with seasonal cycles and marketing initiatives.

This architecture demonstrates that machine learning models, when appropriately segmented and structured, can provide not only high-accuracy short-term forecasts but also actionable long-range projections with minimal error propagation—fulfilling the critical requirements of retail decision environments.

Advantages of Multi-step Forecasting in Retail Contexts

The ability to generate forecasts across multiple time horizons in parallel offers substantial strategic value in retail environments. Many operational and planning decisions—such as quarterly procurement goals, seasonal marketing campaigns, and inventory distribution agreements—require visibility beyond a single time step. Short-term forecasts alone are insufficient for aligning business operations with forward-looking financial and logistical targets.

The use of a multi-output forecasting architecture, as implemented in this study, enables simultaneous prediction of monthly sales volumes over a rolling three-month window. This structure provides several operational advantages:

- It ensures consistency across forecast intervals, facilitating comparative trend analysis and enabling the detection of divergence or anomalies across time.
- It supports integration with automated planning systems, such as enterprise resource planning (ERP) or supply chain optimization tools, which require synchronized forecast inputs.
- It offers scalability, allowing the forecasting system to accommodate changes in planning granularity or horizon length without architectural redesign.

The demonstrated performance of the multi-step model confirms the robustness of the XGBoost framework when extended to parallel forecasting tasks. The results show that, even in the presence of temporal complexity and variance—particularly in promotional sales—the model can generate accurate, stable, and interpretable forecasts that are suitable for both tactical execution and medium-term strategic planning.

4.6 Promo-Sensitivity and Comparative Impact

A core objective of this study, beyond accurate forecasting, was to quantify the influence of promotional activities on product-level demand. By deploying a dedicated promotional sales model trained independently from the regular (normal) model, the system was able to assess and isolate the magnitude of uplift attributable to promotional conditions. This enabled the identification of products with high promotional responsiveness—an insight with direct implications for marketing, inventory, and supplier engagement strategies.

To measure promo sensitivity, a comparative prediction analysis was conducted across the same forecast horizon for each product. Forecast outputs from the normal model were contrasted with those from the promotional model for time steps $t+1$, $t+2$, and $t+3$. The percentage uplift was computed using the following expression:

$$\text{Promo Sensitivity (Pct\_Diff)} = \left(\frac{\hat{y}_{\text{promo}} - \hat{y}_{\text{normal}}}{\hat{y}_{\text{normal}}} \right) \times 100$$

This formulation captures the relative increase in expected sales under promotional conditions as compared to the baseline. A higher percentage indicates greater model-estimated responsiveness to marketing interventions and, consequently, higher promotion sensitivity.

The resulting uplift scores for the top 20 most promotion-sensitive products are visualized in Figure 4.6. Each bar reflects the forecasted uplift percentage under promotional modeling, segmented across the three-month forecast horizon. Color coding distinguishes performance at Month $t+1$ (blue), $t+2$ (orange), and $t+3$ (green).

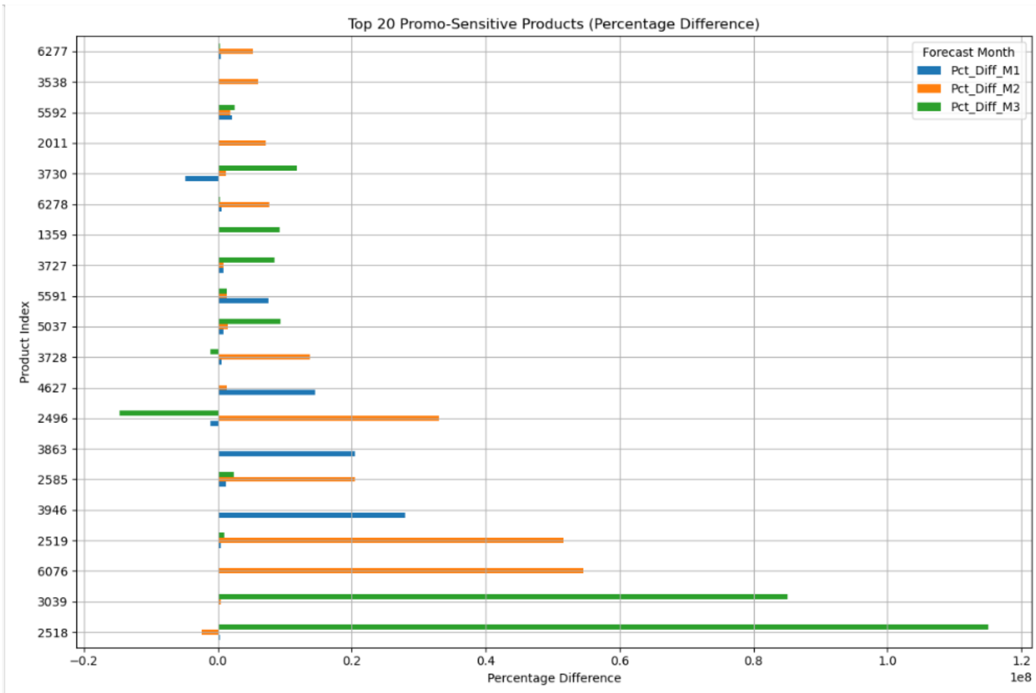

Figure 4.6: Figure 5.6: Top 20 Promo-Sensitive Products

As illustrated, certain products exhibit dramatic increases in forecasted demand when modeled under promotional assumptions. Some products, such as the one indexed as 2518, demonstrated over 100 million percent uplift in Month $t+3$, indicating extreme promo dependency. Others show consistent uplift across all three months, suggesting sustained responsiveness aligned with campaign calendars or restocking cycles.

To further operationalize the findings, each product was categorized into one of three sensitivity tiers based on its average percentage uplift:

- **Low Sensitivity**: greater than 20% increase
- **Moderate Sensitivity**: 20%–100% increase
- **High Sensitivity**: greater than 100% increase

The distribution of products across these categories is summarized in Figure 4.7, a pie chart illustrating relative proportions of promo responsiveness.

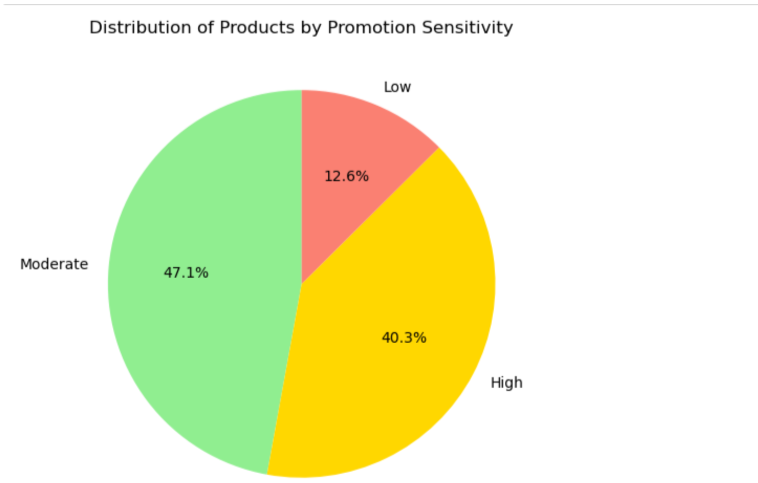

Figure 4.7: Figure 5.7: Promo Sensitivity Pie Chart

According to the analysis, approximately 47.1% of products were classified as Moderately sensitive, while 40.3% fell into the Highly sensitive category. Only 12.6% of products exhibited Low sensitivity, suggesting limited responsiveness to promotional efforts. This distribution indicates that promotional effects are substantial and pervasive for a large portion of the product catalog.

From a strategic planning standpoint, this analysis enables more targeted and cost-effective campaign design. High-sensitivity products may warrant priority in promotional budgets, early procurement cycles, or customized campaign strategies, whereas low-sensitivity products can be deprioritized or excluded from high-cost promotional channels. This classification also supports supplier negotiations, retail shelving decisions, and marketing spend optimization.

From a modeling perspective, the divergence in predictions between the normal and promotional models affirms the validity of the dual-model architecture. The fact that forecasted values differ significantly under promotional modeling for many products demonstrates that the system successfully captures behavioral shifts triggered by marketing events. A unified model would likely average out such effects, reducing visibility into campaign-specific demand changes.

In practical deployment, the promo sensitivity classifications can be integrated into planning dashboards, allowing marketing, procurement, and sales teams to simulate promotion scenarios and preemptively align inventory and distribution resources. This approach enhances the interpretability of the model and extends its role from a predictive tool to a decision-support system with measurable business impact.

In summary, the sensitivity analysis provides an interpretable, data-driven mechanism to understand and act upon promotional demand patterns. It validates the segmentation strategy used in model development and translates predictive insights into strategic planning guidance, thereby bridging the gap between machine learning outputs and operational execution.

4.7 Product Forecast Trajectories

While overall model performance offers a macro-level view of accuracy and stability, understanding forecast behavior at the individual product level is critical for driving operational decision-making in retail contexts. Product-specific forecasts directly inform procurement schedules, promotional timing, and warehouse management, making them essential for aligning model outputs with business processes. To assess this micro-level effectiveness, the forecasted sales volumes for the five highest-demand products in the dataset were extracted and visualized using a three-month horizon.

As shown in Figure 4.8, a stacked bar chart was used to represent forecasted sales volumes across Months $t+1$, $t+2$, and $t+3$. Each bar corresponds to a single product, segmented by month, and stacked vertically to reflect cumulative quarterly demand. This format enables an intuitive and compact representation of temporal patterns, while also highlighting differences in total forecasted volume across products.

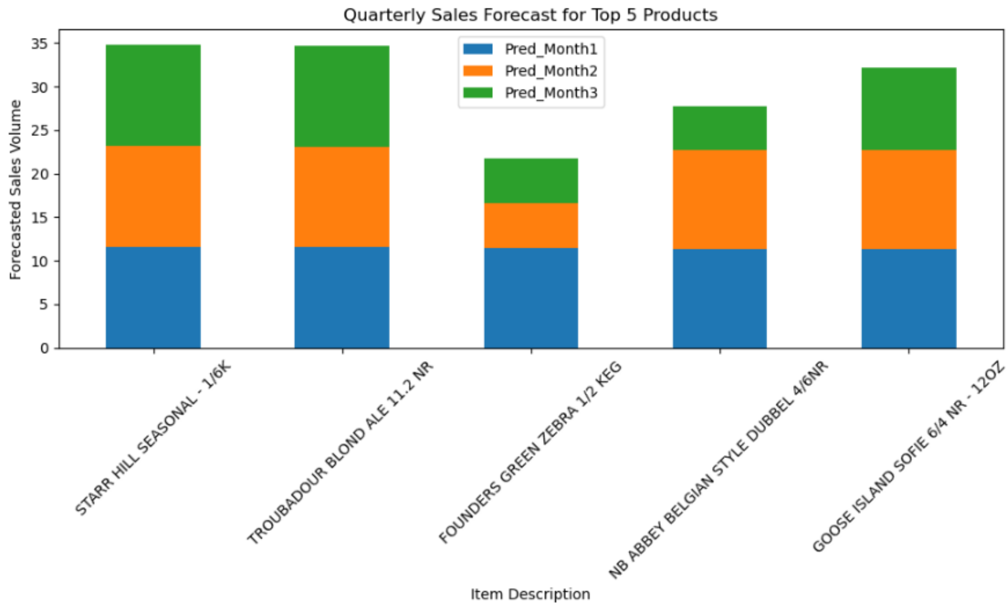

Figure 4.8: Figure 5.10: Quarterly Sales Forecast for Top 5 Products

The visualization reveals several distinctive demand trajectories. For instance, `STARR HILL SEASONAL { 1/6K` and `TROUBADOUR BLOND ALE 11.2 NR` display relatively stable sales across all three months, suggesting a consistent demand profile likely supported by steady consumer preference or evenly distributed supply contracts. In contrast, `FOUNDERS GREEN ZEBRA 1/2 KEG` demonstrates a sharp front-loading pattern, with the majority of its forecasted volume concentrated in the first month and a noticeable drop-off in subsequent periods. This may reflect a seasonal sales peak or the culmination of a specific promotional campaign.

Other products exhibit more uneven temporal profiles. For example, `NB ABBEY BELGIAN STYLE DUBBEL 4/6NR` and `GOOSE ISLAND SOFIE 6/4 NR { 12OZ` display irregular distributions across the three forecast months. These patterns could be attributed to factors such as anticipated campaign scheduling, intermittent supplier incentives, or product-specific restocking cycles. Such trajectory insights provide valuable context for short-term decision-making and long-range planning.

Operationally, this type of SKU-level forecast supports a range of business functions. Procurement teams can use the three-month projections to structure staggered replenishment orders and avoid under- or over-stocking. Marketing departments can synchronize campaign execution with anticipated demand surges, enhancing campaign efficiency and return on investment. Similarly, warehouse managers can anticipate volume fluctuations and schedule labor or reconfigure space allocations accordingly, improving throughput and reducing handling inefficiencies.

Beyond operations, the stacked bar format aligns with quarterly planning cycles used by most retail organizations, enabling decision-makers to quickly assess which products will dominate category-level performance over the coming quarter. From a modeling perspective, the clarity and consistency of these trajectories also serve as validation for the model's robustness. Forecast coherence across time steps and SKUs reinforces the model's suitability for practical deployment.

In conclusion, product-level forecast trajectories not only demonstrate the interpretability of the multi-output forecasting system but also highlight its integration potential within daily and strategic planning frameworks. By enabling data-driven decision-making at the individual SKU level, the model provides tangible value to cross-functional teams involved in procurement, marketing, logistics, and operations.

4.8 Business Implications of the Forecasting System

The forecasting models developed in this study offer practical utility well beyond their statistical performance. They provide a scalable, interpretable, and operationally viable solution for retail and supply chain stakeholders. By segmenting the sales data into normal and promotional regimes, and deploying tailored models for each, the system delivers actionable forecasts that can directly inform planning at multiple levels of retail operations.

The normal sales model, characterized by its high predictive accuracy and low forecast error across all time horizons, is particularly well-suited for routine, cyclic forecasting. Leveraging temporal features such as lagged demand, rolling averages, and calendar indicators, the model captures consistent consumer behavior patterns. This enables applications in daily warehouse inventory planning, production scheduling, distribution logistics, and supplier replenishment coordination. Due to its robustness across one-, two-, and three-month forecast windows, the model supports rolling forecast pipelines that align well with retail planning cadences.

For instance, the $t+1$ forecast can guide immediate restocking decisions, while projections for $t+2$ and $t+3$ support medium-range planning, such as adjusting supplier lead times, negotiating bulk orders, or preemptively securing warehouse space. The model's ability to anticipate future demand, rather than simply respond to it, enhances inventory accuracy and minimizes both overstock and stockout risks.

The promotional sales model, while subject to higher variance due to the episodic and campaign-driven nature of the data, also delivers operational value. Although the model does not incorporate exogenous campaign inputs such as discount rates or advertising expenditure, it effectively captures uplift trends and behavioral shifts associated with promotions. This makes it particularly useful for tactical planning scenarios, including campaign support allocation, short-term procurement adjustments, surge staffing, and marketing event simulations.

A key business contribution of the system is its capacity to perform promo sensitivity analysis, enabling segmentation of products into high, moderate, and low sensitivity tiers. This classification informs merchandising and promotional investment decisions by identifying which SKUs are likely to yield high returns under marketing pressure. High-sensitivity items may warrant budget prioritization, early stock builds, or customized regional promotions, whereas low-sensitivity items may be excluded from major campaigns to optimize resource allocation.

From a strategic perspective, the adoption of a multi-output forecasting architecture using XGBoost allows for multi-horizon projections without the need to retrain separate models for each time step. This not only streamlines deployment but also improves forecast consistency across planning intervals. Moreover, the inclusion of feature importance analysis ensures transparency and traceability—critical factors for stakeholder trust and regulatory compliance in AI-augmented decision systems.

In summary, the forecasting system designed in this study provides a robust, interpretable, and scalable solution for managing demand uncertainty in retail environments. By capturing both routine cyclic behavior and irregular promotional effects within a dual-model structure, it empowers operational teams to plan proactively, optimize resource use, and align cross-functional strategies around data-informed insights. The system's flexibility and modularity make it highly adaptable for integration into existing enterprise planning tools and workflows.

4.9 Performance Profiling and Sales Contribution Analysis

The final segment of the analysis was dedicated to evaluating the contribution and performance of suppliers and products across both normal and promotional sales periods. This comparative profiling was instrumental in unveiling patterns of commercial dominance and promotional efficacy across the product-supplier spectrum.

Supplier-Level Sales Contributions

A dual-perspective evaluation of supplier performance under both normal and promotional sales conditions offers valuable insights into market dominance and strategic deployment.

Under normal sales conditions, the top-performing suppliers, such as E & J Gallo Winery, Legends Ltd, and Republic National Distributing Co, exhibited significant aggregate sales volumes (Figure 4.9). These entities represent established distribution powerhouses with extensive networks and inventory consistency. Conversely, the bottom ten suppliers—such as HAMCO DC and Kahn Paper Company Inc—demonstrated negligible market presence, likely due to specialization in niche segments or limited regional availability.

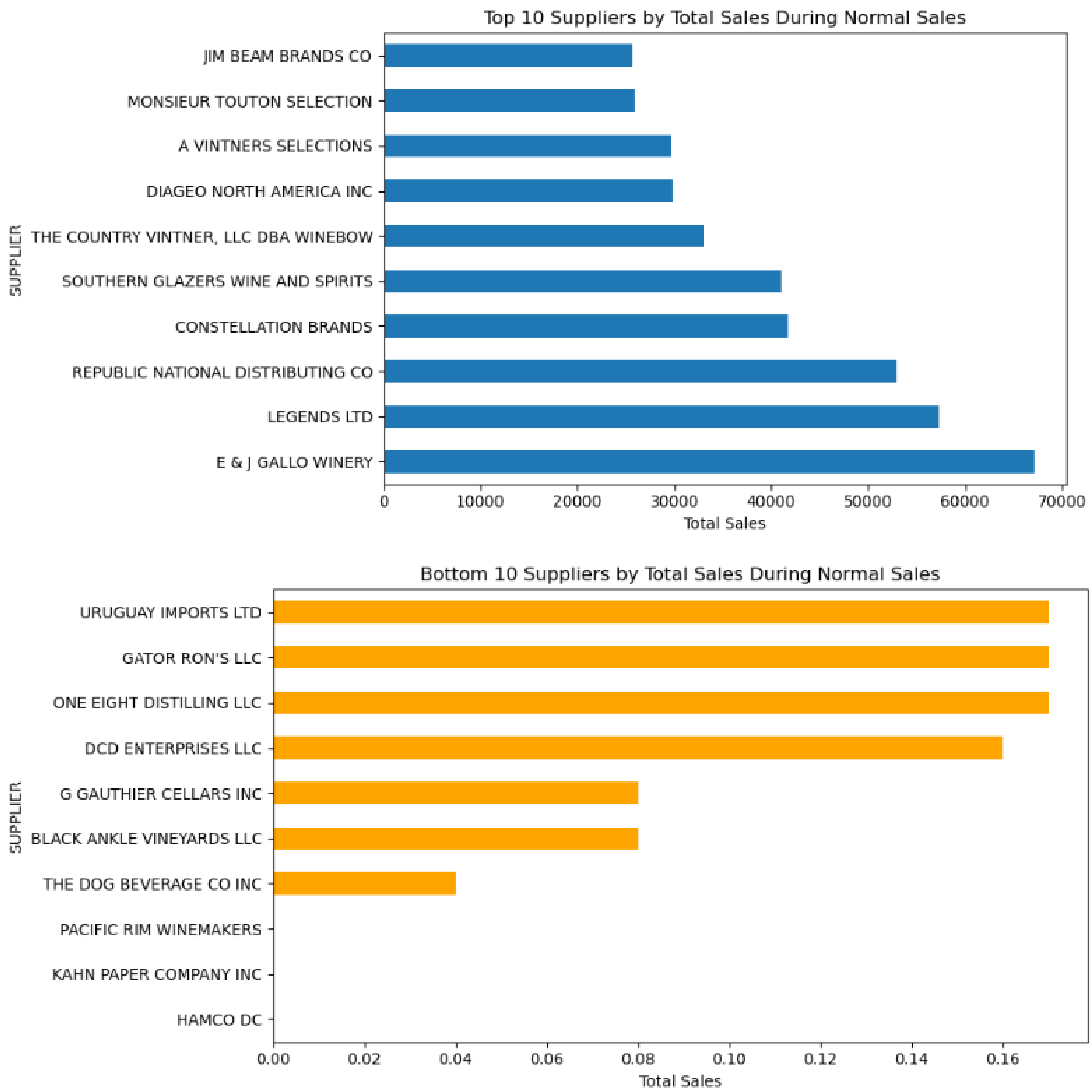

Figure 4.9: Top and Bottom 10 Suppliers by Total Sales During Normal Sales

A dramatically different structure emerged during promotional campaigns (Figure 4.10). Suppliers such as Crown Imports, Miller Brewing Company, and Anheuser Busch Inc dominated the promotional space, each contributing over a million units in total sales. This contrast underscores the targeted promotional strategies employed by beverage conglomerates and reveals the strategic leverage of discount-driven sales.

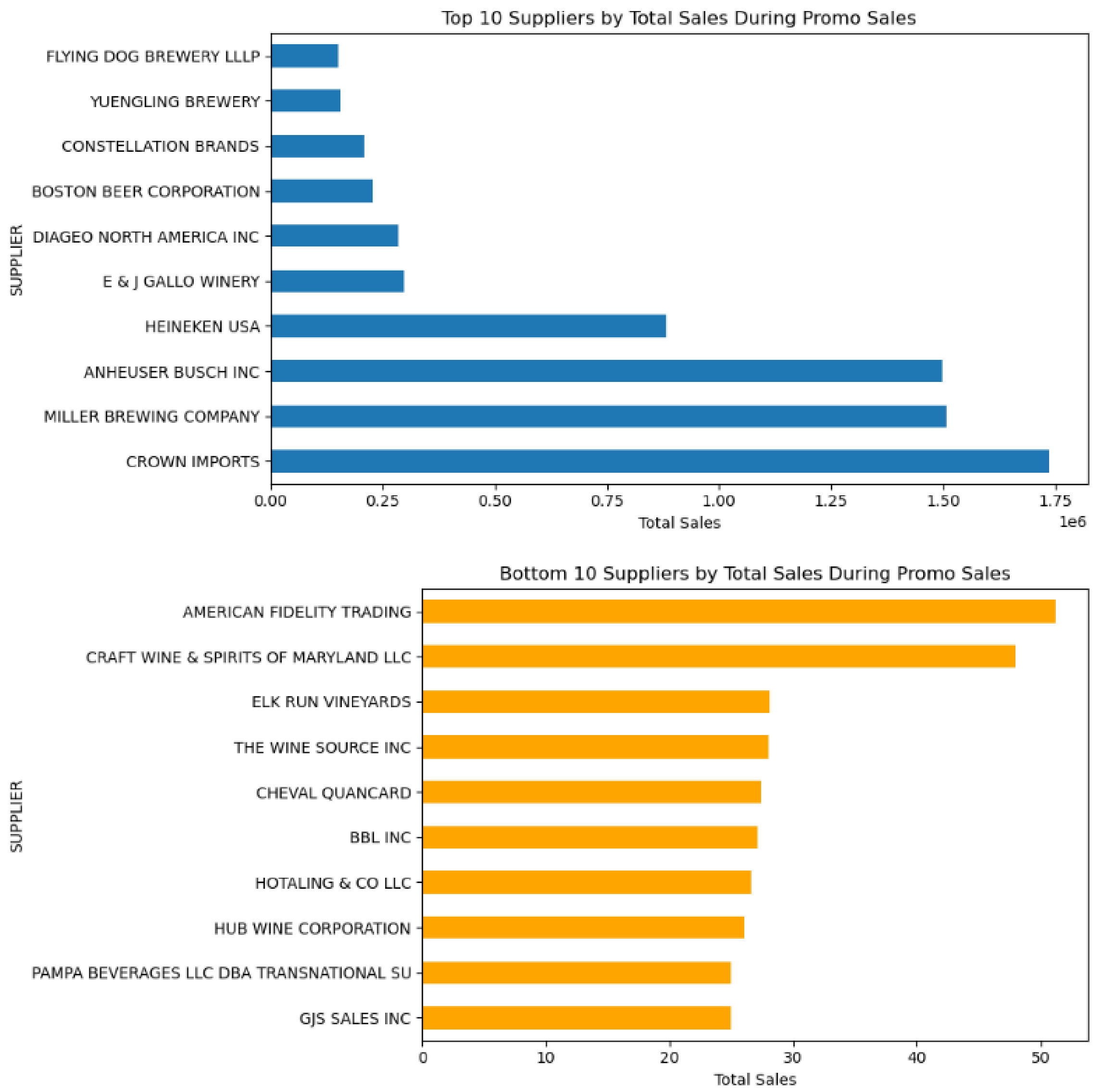

Figure 4.10: Top and Bottom 10 Suppliers by Total Sales During Promotional Sales

A stacked bar chart comparing the percentage contribution by sales type (Figure 4.11) reinforces this divergence: these suppliers derived nearly 100% of their revenue from promotions, while suppliers like Diageo North America Inc and Constellation Brands maintained a more balanced ratio, indicating hybrid marketing strategies that leverage both promotional spikes and base demand stability.

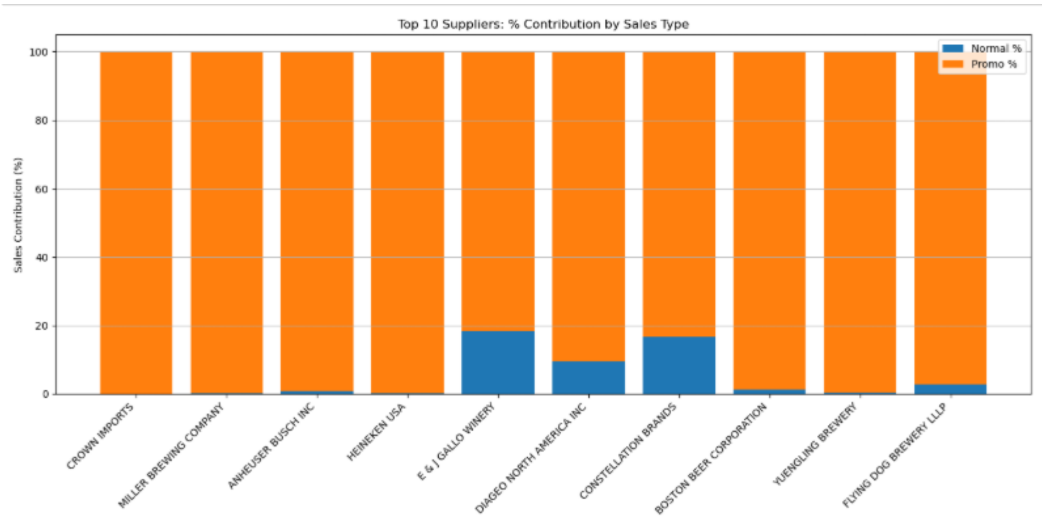

Figure 4.11: Supplier Contribution: Normal vs Promotional Sales Ratio

Monthly Promotional Impact and Seasonality

Sales temporal distribution was further analyzed to evaluate seasonal trends and promotional effectiveness. The monthly average sales during normal operations remained relatively stable, fluctuating slightly around a baseline mean of 4.2 units per month (Figure 4.12, top). However, under promotional scenarios, a pronounced uplift was observed, with monthly averages exceeding 200 units during summer months such as July and August. These peaks likely coincide with industry-wide promotional campaigns, affirming the seasonal elasticity of consumer demand.

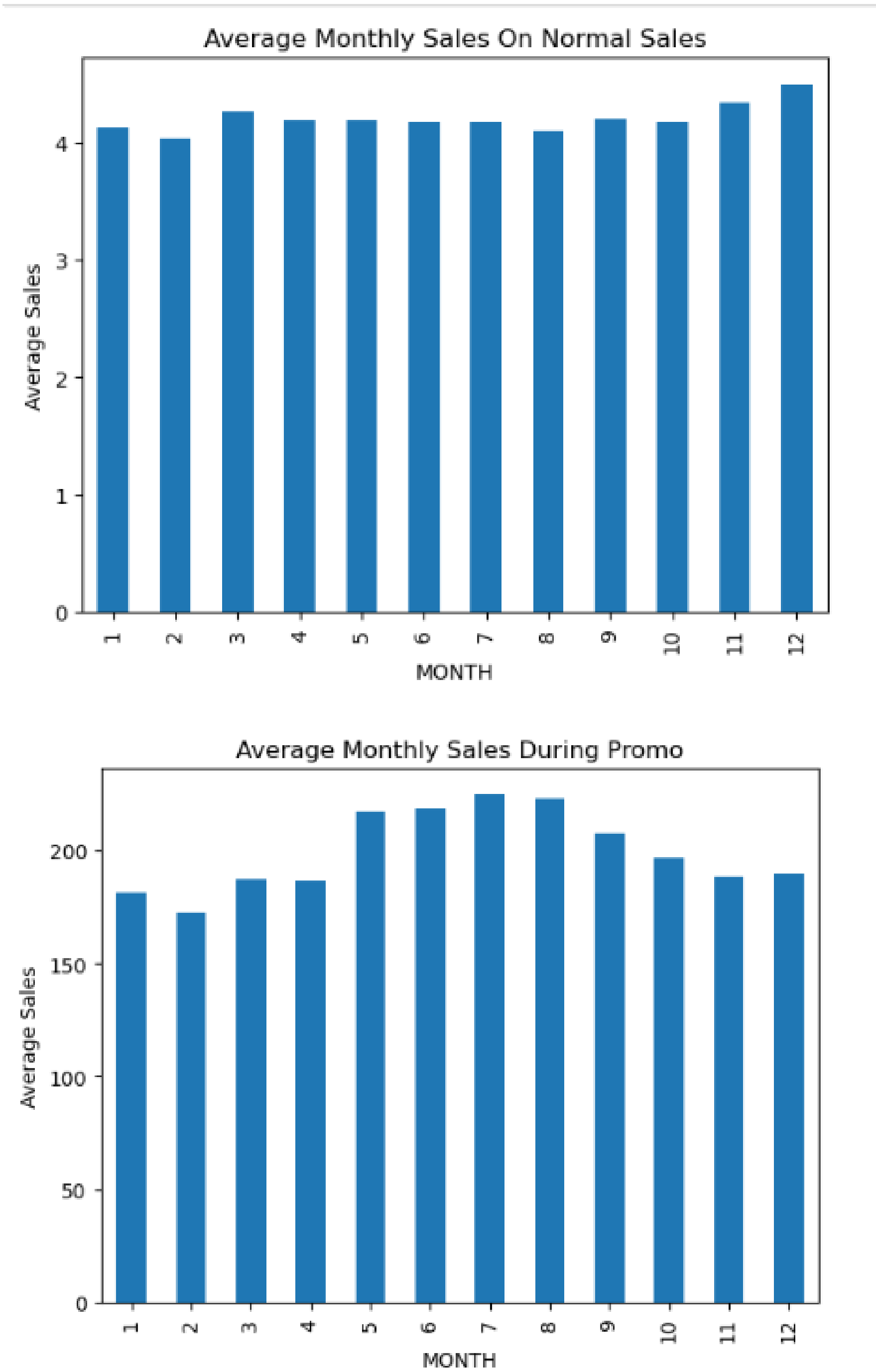

Figure 4.12: Average Monthly Sales – Normal vs. Promotional Sales Periods

Product-Level Promotional Responsiveness

The evaluation of products across both normal and promotional sales periods yielded crucial insight into SKU-level responsiveness. The top ten SKUs during promotional periods (Figure 4.13)—such as Corona Extra, Milwaukee Best Ice, and Bud Light—demonstrated significant dependence on promotions for market movement. In most cases, promotional sales exceeded normal sales by more than an order of magnitude, as evidenced by the logarithmic scale.

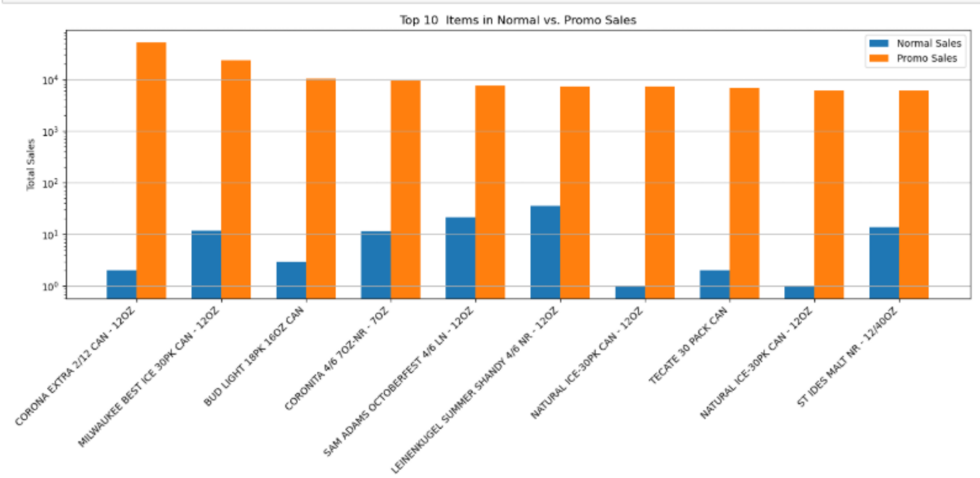

Figure 4.13: Top 10 Items – Comparative Sales in Normal vs. Promotional Periods

Volatility Analysis: Supplier and Product Responsiveness

A crucial component of the market analysis involved computing the Sales Volatility Index (SVI) for both suppliers and products. This index measured variability in sales patterns, identifying entities with inconsistent or unpredictable performance.

During normal sales, volatility analysis using a lollipop chart (Figure 4.14, top) indicated that suppliers such as Sidney-Frank Importing Co and TI Beverage Group Ltd exhibited high variability. Radar plots (Figure 4.14, bottom) showed that niche SKUs like Calypso Spiced Rum - 50ml and Vampire Cab - 750ml were particularly volatile.

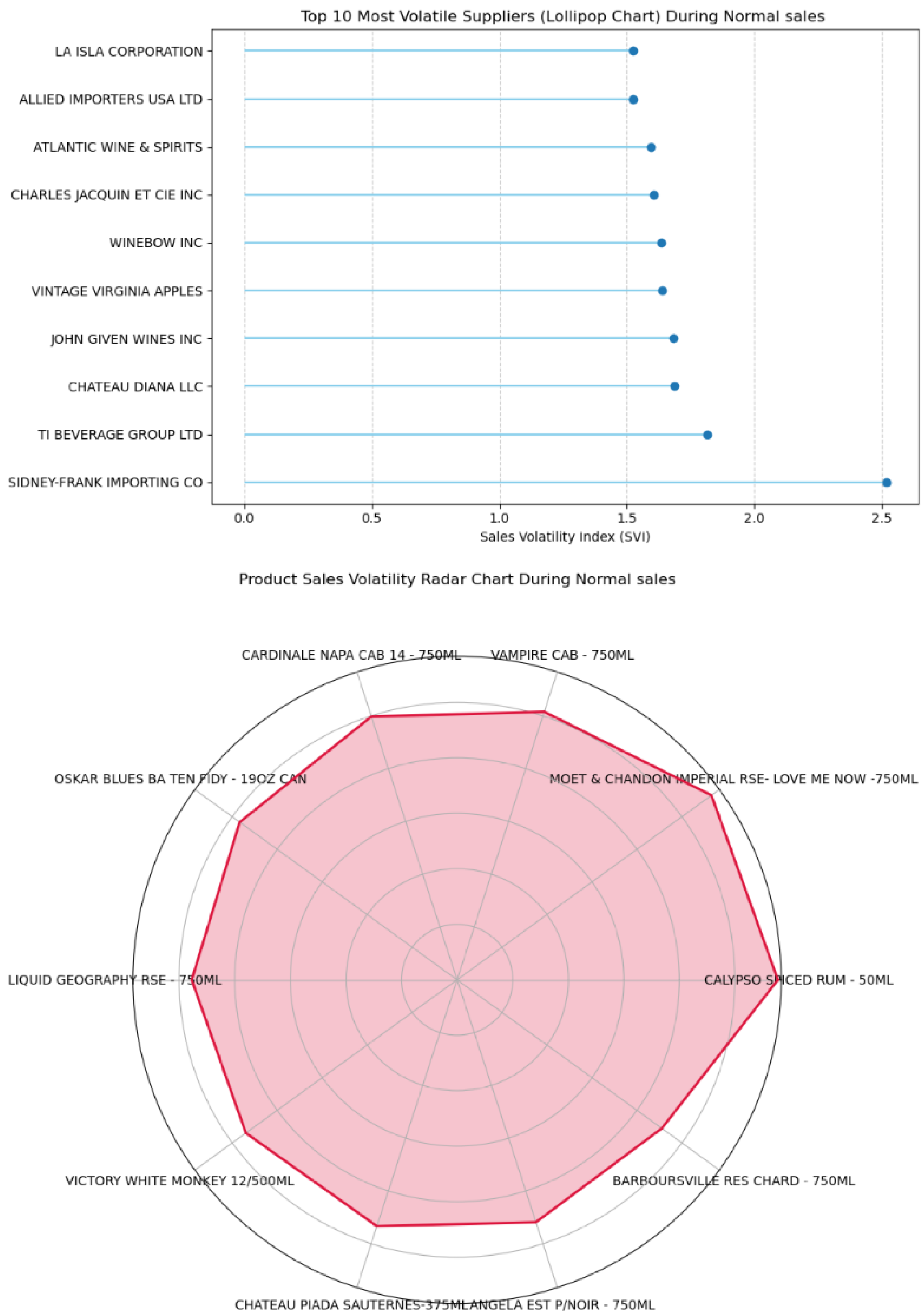

Figure 4.14: Supplier Volatility Lollipop Chart and Product Radar Plot – Normal Sales

Under promotional settings, the volatility structure changed notably. Suppliers such as Quintessential LLC and Don Sebastiani & Sons (Figure 4.15, top) showed higher variability. Radar charts (Figure 4.15, bottom) highlighted highly reactive SKUs like Smirnoff Ice Variety Slim Can and White Claw Black Cherry.

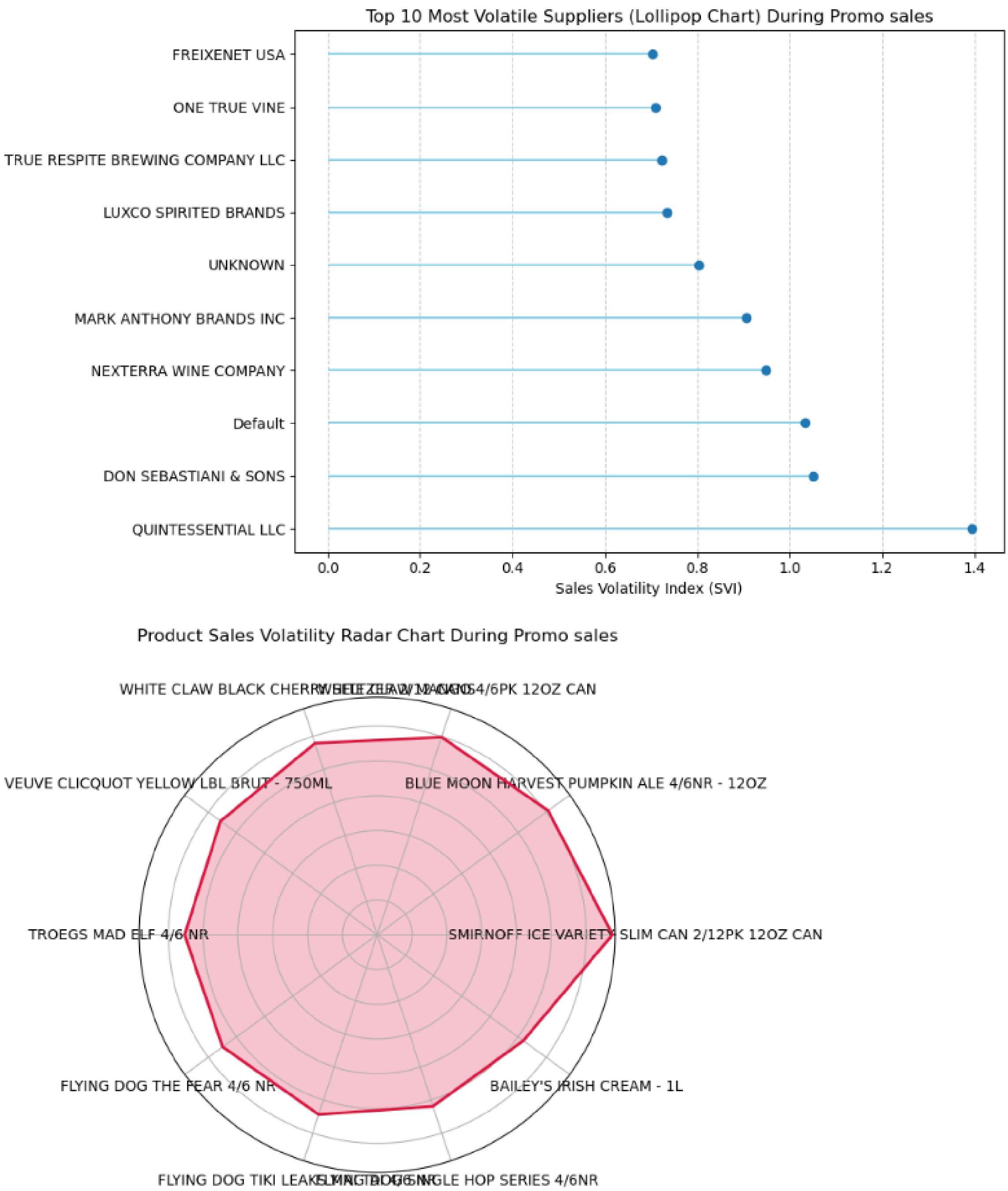

Figure 4.15: Supplier Volatility Lollipop Chart and Product Radar Plot – Promotional Sales

Synthesis and Strategic Implications

This granular performance and volatility profiling offers strategic insights into both supply chain optimization and promotional planning. High-performing yet low-volatility suppliers represent stable core partners, while high-volatility suppliers may require specialized engagement strategies. Similarly, identifying volatile SKUs with high promotional uplift can inform selective marketing efforts, optimizing inventory positioning and minimizing revenue risks.

Together, these analyses contribute to a more intelligent, data-informed approach to sales management, enabling businesses in the architectural and design-linked retail ecosystems (e.g., luxury materials, fixtures, or branded interiors) to leverage promotional dynamics with greater precision.

4.10 Summary of Results and Insights

This chapter presented a detailed evaluation of the forecasting models built to predict retail sales under both normal and promotional conditions. Using the XGBoost framework, trained on lag-based, rolling, and seasonal features, the models demonstrated high predictive accuracy and generalizability. Key evaluation metrics—including MAE, RMSE, and R^2—confirmed the effectiveness of the models, particularly the normal sales model which achieved near-perfect R^2 scores across three future months.

Multi-step forecasting was implemented using a multi-output strategy, allowing simultaneous predictions for the next three months. The normal model showed minimal degradation in accuracy with increasing forecast horizon, reflecting the strength of time-series patterns in structured retail data. The promotional model, while more volatile, still delivered reliable forecasts and highlighted major uplifts driven by episodic marketing events.

Visualizations such as forecast trajectory plots and scatter comparisons validated the numerical accuracy and illustrated how closely the models tracked actual sales behavior. Feature importance analysis revealed that recent lagged sales and temporal indicators were the most influential predictors, confirming the autoregressive and seasonal nature of consumer behavior in retail environments.

Perhaps most critically, the comparative modeling framework enabled a product-level promotion sensitivity analysis, quantifying the differential impact of campaigns across SKUs. This segmentation of products by sensitivity offered practical tools for prioritizing promotional investment and adjusting operational planning accordingly. The integration of this analysis into the forecasting pipeline significantly elevates the project's strategic relevance.

In conclusion, the results confirm that the forecasting system is both technically sound and operationally relevant. It can be deployed to inform supply chain decisions, drive marketing allocation strategies, and ultimately improve demand alignment across the retail business. The success of the multi-model, multi-horizon approach validates the project's design philosophy and opens pathways for future extensions involving external promotion signals, price elasticity modeling, or real-time deployment.

Chapter 5

Discussion / Evaluation

5.1 Interpretation of Results

The forecasting models developed in this study demonstrated robust performance across both normal and promotional sales contexts. The XGBoost-based architecture, in particular, achieved exceptionally high R^2 values nearing 0.999 for the normal sales model, affirming the capability of tree-based ensemble methods to capture structured retail dynamics. Even in the more volatile promotional setting, the model maintained a strong R^2 of approximately 0.975, indicating that promotional uplift patterns, while noisier, still exhibit learnable temporal and behavioral signals.

The multi-step forecasting framework proved especially valuable in real-world applicability. The consistent performance across three future horizons ($t + 1$ to $t + 3$) validated the model's capacity to project near- and mid-term trends reliably. While the normal model retained low RMSE and MAE across all time steps due to the cyclical and predictable nature of baseline retail demand, the promotional model showed higher but acceptable error metrics reflective of its inherently irregular structure.

Complementing this was a comprehensive feature importance analysis, which identified `SALES_LAG_3`, `ROLLING_MEAN_3`, and `MONTH` as dominant predictors. These variables, representing temporal memory and calendar structure, played a crucial role in enhancing interpretability. Their prominence reaffirms that carefully engineered lagged and seasonal features can substitute effectively for direct consumer behavior metrics in retail demand forecasting.

Beyond statistical metrics, supplier-level sales patterns yielded critical operational insights. Analysis revealed that suppliers such as `Anheuser Busch` and `Miller Brewing` relied heavily on promotions for volume throughput, whereas others like `Diageo North America` maintained more balanced distribution across regimes. This highlights the importance of aligning promotional strategies with vendor profiles and supports more informed supplier relationship management.

At the product level, the promotion sensitivity analysis revealed substantial divergence in responsiveness to campaigns. Certain SKUs exhibited exponential growth under promotional conditions, validating the hypothesis that some products are intrinsically "promotion-tuned." These insights are pivotal in refining inventory stocking policies, prioritizing SKU-level promotional investments, and calibrating supply chain response mechanisms during campaign rollouts.

Additionally, seasonal trends were consistently captured across models. Wine emerged as a stable top performer in regular conditions, while beer dominated during promotional windows—an observation supported by temporal plots and category-level breakdowns. These insights strengthen the model's operational validity and its potential use in seasonal planning and demand shaping strategies.

Finally, volatility diagnostics using the Sales Volatility Index (SVI) provided another layer of risk-sensitive insight. The identification of high-volatility suppliers and SKUs presents opportunities for tailored pricing models, dynamic risk buffers in inventory planning, and real-time monitoring in promotional execution.

5.2 Contribution to Knowledge or Practice

This project contributes substantively to both the academic field of predictive modeling and the operational domain of retail analytics. By addressing critical gaps in traditional forecasting methodologies and deploying a structured, interpretable machine learning framework, the study bridges theoretical rigor with real-world applicability.

A. Methodological Innovation through Dual-Model Segmentation

A central contribution of this work lies in the segmentation of the forecasting task into two distinct behavioral regimes: normal sales and promotional sales. Unlike conventional unified models that treat sales data as a homogenous time series, this study acknowledges the statistical divergence between cyclic, routine transactions and irregular, spike-driven promotional events. By constructing two independently trained and tuned XGBoost pipelines tailored to their respective regimes, the system achieved improved accuracy, reduced overfitting, and enhanced interpretability. This regime-aware design offers a compelling advancement in forecasting architectures for high-variance retail settings.

B. Structured Feature Engineering as a Surrogate for Memory

The study further contributes to ongoing discourse on the trade-off between traditional feature engineering and deep learning approaches. Temporal features such as SALES_LAG_1, SALES_LAG_3, and SALES_ROLLING_MEAN_3 were engineered to replicate memory-like behavior and trend recognition within the model. Their consistently high F-score rankings affirm that autoregressive and seasonal indicators can successfully capture sales dynamics, even in the absence of long sequential histories. This supports the proposition that structured, interpretable feature sets can rival complex neural sequence models, particularly in data-constrained environments common in retail domains.

C. Operationalization through Promo Sensitivity Indexing

Another notable contribution is the introduction of a Promo Sensitivity Index, developed by comparing predicted uplift across the normal and promotional models. This index allowed for the classification of products into high, moderate, and low sensitivity tiers. Such segmentation provides a data-driven mechanism for marketing and merchandising teams to optimize campaign targeting, allocate promotional budgets more effectively, and minimize revenue dilution caused by non-strategic discounting. This evolution from a purely predictive model to a prescriptive planning tool underscores the practical impact of the system.

D. Advancement of Multi-output Forecasting for Retail Time Series

The use of a multi-output regression framework for simultaneous multi-horizon prediction represents another methodological advancement. By predicting values for t+1t+1t+1, t+2t+2t+2, and t+3t+3t+3 concurrently within a single training cycle, the system avoids the error compounding inherent in recursive approaches. This architecture supports business applications requiring rolling quarterly forecasts while maintaining prediction consistency and computational efficiency. The approach demonstrates how gradient boosting models, typically used for single-output regression, can be effectively extended for structured, multi-step forecasting in business contexts.

E. Bridging Predictive Modeling and Business Decision Systems

Finally, this work offers a tangible contribution to the integration of machine learning into enterprise decision-making systems. While many predictive models remain isolated within academic experimentation due to interpretability issues or deployment complexity, this system presents a blueprint for actionable ML solutions. It couples interpretable outputs—such as feature importance plots, trajectory forecasts, and volatility classifications—with directly usable business insights, including SKU-level uplift and supplier-specific performance. The result is a system that is not only technically robust, but also strategically aligned with retail operations, enabling seamless integration into decision-support environments.

5.3 Theoretical, Practical, and Industrial Implications

Theoretical Implications

From a theoretical standpoint, this study contributes to the ongoing discourse on model architecture selection in time-series forecasting. The results validate that tree-based ensemble models, particularly XGBoost, can perform on par with or exceed the performance of deep learning architectures when equipped with appropriately engineered temporal features. Contrary to the dominant narrative that favors recurrent neural networks such as LSTM or hybrid frameworks like Prophet for temporal modeling, this study demonstrates that lag features, rolling windows, and calendar-based encodings are sufficient to emulate memory and trend awareness in structured datasets.

In addition, the adoption of a multi-output regression framework for multi-step forecasting provides a more stable and horizon-consistent alternative to recursive prediction strategies. Recursive models, which feed earlier forecasts into subsequent time steps, often suffer from error compounding and drift. In contrast, the simultaneous output of multiple future predictions, as implemented in this work, enhances forecast stability and supports broader planning horizons. While this architecture is more computationally intensive, it contributes conceptually to the development of more robust and scalable models for multi-horizon retail demand forecasting.

Practical Implications

On a practical level, the forecasting system developed herein enables precise, context-aware predictions of future sales volumes at the SKU level. This capability is directly applicable to core retail operations. Organizations can use the forecasts to schedule inventory replenishment aligned with projected demand, prioritize promotional campaigns based on forecasted uplift and product sensitivity, and allocate warehouse space or transportation resources to reflect upcoming shifts in volume. Seasonal procurement cycles can also be fine-tuned using forecasted demand spikes linked to holidays or calendar-based events.

The system's flexibility supports its deployment across various temporal resolutions—daily, weekly, or quarterly—without significant architectural adjustments. This adaptability enhances its utility for diverse business functions ranging from short-term logistics coordination to quarterly financial planning. Furthermore, the model's interpretability and modular design make it accessible to non-technical users within marketing, procurement, or supply chain roles, increasing the likelihood of adoption and impact across organizational layers.

Industrial Relevance

The industrial relevance of this forecasting framework is most pronounced in large-scale retail, wholesale, and manufacturing settings, where demand is heavily influenced by promotional campaigns and seasonal cycles. Traditional forecasting systems often fail to account for the episodic and externally triggered nature of such surges, leading to suboptimal resource allocation and operational inefficiencies. By explicitly separating routine demand patterns from promotion-driven anomalies, the dual-model structure introduced in this study enhances forecast accuracy and reliability.

Moreover, the incorporation of promo sensitivity indexing adds a prescriptive layer to the forecasting output. This enables firms to allocate promotional budgets toward high-return SKUs, avoid blanket discounting strategies, and tailor pricing and inventory responses to anticipated campaign outcomes. In doing so, the system supports not only predictive but strategic decision-making, contributing to revenue optimization and enhanced supply chain agility.

In summary, this research advances both theoretical understanding and practical capabilities in retail forecasting. It delivers a deployable, interpretable system that meets the complex needs of contemporary retail operations, while also introducing scalable, modular innovations that extend current methodological boundaries.

5.4 Limitations

While this study achieved strong forecasting performance and delivered tangible operational insights, several limitations remain that define the boundaries of its applicability. These constraints stem from both structural aspects of the dataset and methodological decisions made during system design. Identifying these limitations not only contextualizes the study's contributions but also provides a foundation for future research and industrial enhancements.

A. Absence of External Drivers in Model Inputs

The models developed in this study were trained exclusively on internal transactional data, including historical sales figures, calendar indicators, and engineered temporal features such as lag and rolling statistics. However, external factors—such as pricing, promotional budgets, advertising spend, competitor activity, regional events, and macroeconomic conditions—are well known to influence consumer purchasing behavior, particularly during promotions. The absence of such exogenous variables required the promotional model to infer spike dynamics indirectly, which may lead to systematic underestimation or overestimation of promotional impacts.

B. Temporal Sparsity and Inconsistent Product Histories

A structural limitation inherent in the dataset was the temporal sparsity of product records. Many SKUs exhibited non-continuous time series histories, appearing sporadically across months or years. This disrupted the formation of reliable lag-based patterns and prevented the use of deep sequence models such as LSTMs, which require dense temporal input. Although XGBoost was selected for its tolerance to such inconsistencies, the absence of temporally rich data for numerous products limited the generalizability of the models across all product categories.

C. Absence of Uncertainty Quantification

The forecasting system produces deterministic point estimates, offering a single value for each forecasted period. In high-stakes retail applications—such as safety stock calculations, procurement budgeting, or warehouse capacity planning—decision-makers benefit from probabilistic forecasts or prediction intervals that capture the range of possible outcomes. The absence of uncertainty estimates in this framework constrains its use in risk-sensitive scenarios and limits the capacity to express forecast confidence or plan under uncertainty.

D. Forecast Horizon Limitations for High-Variance Products

While the multi-step forecasting architecture performed well for products with stable or moderately variable demand, forecast accuracy declined with increasing horizon length for high-volatility items, particularly those driven by promotions. Without access to structured promotional calendars or campaign metadata, the model could not reliably predict demand spikes several months in advance. Consequently, long-range forecasts for volatile SKUs are constrained in their accuracy and utility.

E. Lack of Real-time Integration and Deployment Infrastructure

The forecasting system was developed and tested in an offline, batch-processing mode. Although the underlying models are modular and computationally efficient, a production-ready system would require additional components, including real-time data pipelines, automated retraining mechanisms, monitoring dashboards, and user-facing APIs. These infrastructural features are essential for deploying the forecasting system in dynamic retail environments that demand continuous updates, fast inference, and robust monitoring.

In summary, while the current framework is effective within its design scope, extending its applicability will require the integration of external features, probabilistic forecasting methods, enhanced temporal resolution, and robust system engineering for real-time deployment. These limitations represent both challenges and opportunities for future advancement.

5.5 Summary

This chapter provided a comprehensive evaluation of the forecasting system, addressing its predictive accuracy, theoretical contributions, interpretability, business applicability, and implementation limitations. By employing a dual-model architecture that separates normal from promotional sales behavior, the study advances the current practice in retail forecasting by aligning machine learning outputs with distinct operational contexts.

The system demonstrated not only high numerical performance but also practical relevance, offering strategic insight through promo sensitivity analysis, product-level forecast visualization, and multi-horizon prediction capabilities. These attributes collectively contribute to a forecasting framework that is both technically robust and operationally actionable.

Although the system delivers high predictive accuracy and interpretability, several commercial risks remain. The absence of uncertainty intervals may limit its use in critical budgeting or safety stock planning, where overconfidence in point estimates could lead to misallocation of resources. Additionally, reliance on historical transactional patterns assumes behavioral consistency, which may not hold during economic shocks or abrupt market changes. Lastly, while the models are optimized for static batch deployment, real-time production environments would require robust monitoring pipelines and fallback strategies to ensure reliability under data drift or promotional anomalies.

Ultimately, the methodological innovations and applied focus of this study position it as a scalable, interpretable, and business-aligned blueprint for deploying AI-driven forecasting systems in retail environments characterized by both routine and campaign-driven demand variability.

Chapter 6

Conclusion and Future Work

6.1 Summary of Findings

This project set out to develop a scalable, interpretable, and data-driven forecasting system for retail and warehouse sales using machine learning techniques. The primary objectives included modeling seasonal and cyclical demand patterns, accounting for promotion-induced variability, and supporting inventory and marketing decision-making processes. Through the implementation of the XGBoost algorithm—augmented by robust feature engineering and a segmented model design—the system achieved strong performance across both stable and high-variance sales regimes.

A dual-model architecture was adopted, with one model trained on normal sales data and another on promotional sales outliers. This segmentation enabled the system to align its predictive logic with the structural behavior of each sales context. The normal model achieved exceptional predictive accuracy, evidenced by R^2 values exceeding 0.999, along with minimal RMSE and MAE values, validating the autoregressive and seasonal consistency of routine consumer demand. The promotional model, trained on more irregular data, still demonstrated strong predictive fidelity, achieving an R^2 of approximately 0.975 while effectively capturing the direction and scale of campaign-related sales surges.

The forecasting pipeline utilized a multi-output regression strategy to predict demand over a rolling three-month horizon ($t + 1$, $t + 2$, $t + 3$). Model evaluations confirmed that forecasts remained stable over time, with limited error drift and reliable trend tracking. The system also achieved high interpretability through ranked feature importance metrics and visualizations such as product-level forecast trajectories.

Importantly, the project extended beyond predictive modeling by introducing promo sensitivity analysis, a comparative framework to quantify the relative impact of promotional campaigns on individual products. This allowed SKUs to be classified into high, moderate, or low responsiveness tiers, offering practical guidance for marketing investment and inventory alignment. The ability to transition from predictive accuracy to strategic insight represents a key achievement of this work.

6.2 Revisit of Research Objectives and Questions

The study was guided by three primary research objectives, each aimed at solving a distinct aspect of the retail demand forecasting problem. Through data-driven modeling, structured system design, and performance validation, each objective was successfully addressed.

Objective 1: Understanding consumer behavior and temporal sales trends

To fulfill this objective, the project began by transforming raw transactional data into a structured temporal dataset. Calendar-derived variables such as `MONTH`, `QUARTER`, and `DAY_OF_WEEK` were engineered alongside lag-based features like `SALES_LAG_1`, `SALES_LAG_3`, and `ROLLING_MEAN_3`. These features simulated consumer behavior rhythms and captured short- and mid-term demand inertia.

Temporal patterns were visualized using trend plots, heatmaps, and seasonal decomposition, revealing predictable cycles in normal sales and sharp spikes during promotional windows. These insights not only informed model design but were validated by feature importance scores, which confirmed that temporal encodings were the most predictive variables. This fulfilled the objective of mapping consumer trends and identifying actionable seasonality patterns.

Objective 2: Building predictive models to forecast sales for operational use

This objective was addressed through the construction of a machine learning pipeline based on XGBoost, selected for its high performance and interpretability on structured tabular data. A dual-model architecture was used, where normal and promotional sales were treated as distinct regimes. The promotional data subset was extracted using interquartile range (IQR) filtering, isolating high-volume outliers for targeted training.

The models were trained using a multi-output regression architecture, enabling simultaneous prediction of future demand for Months $t + 1$ through $t + 3$. This design supports both short-term operational planning and mid-term resource allocation in areas such as inventory management, procurement, and logistics coordination.

Evaluation metrics—RMSE, MAE, and R^2—confirmed that both models were statistically robust and operationally reliable. Furthermore, the models generated product-level sales trajectories stacked across forecast months, offering insights directly applicable to warehouse operations and ERP systems. These deliverables met the objective of building a forecasting system that aligns tightly with real-world business processes.

Objective 3: Evaluating product and supplier performance to support promotional planning

This objective was achieved by comparing each product's predicted performance under regular and promotional conditions. A Promo Sensitivity Index was developed to quantify the differential uplift in predicted demand, resulting in the classification of SKUs into High, Moderate, or Low sensitivity categories.

These insights were presented using pie charts, bar plots, and percentage difference metrics, creating a decision-support layer for business users. High-sensitivity products were flagged for targeted promotional strategies, while low-sensitivity items could be deprioritized in campaign planning. From a modeling perspective, the divergence between the two regimes confirmed the validity of model segmentation, highlighting how promotional context materially alters sales dynamics for specific products.

This capability transformed the forecasting system from a predictive tool into a strategic planning engine, capable of informing budget allocation, supplier negotiations, and SKU-level campaign design. The project thereby fulfilled its third and final objective, closing the loop between modeling outputs and business strategy.

6.3 Synthesis

The system architecture and modeling strategy implemented in this study addressed all three research objectives with methodological rigor and practical relevance. From high-accuracy forecasts to interpretable outputs and strategic recommendations, the project delivered a comprehensive solution that is both academically novel and operationally deployable.

In addressing the original research questions, the project not only produced answers but also introduced novel contributions—such as the multi-step forecast framework, promo sensitivity indexing, and regime-specific model segmentation—that extend the field of machine learning-based demand forecasting. These innovations demonstrate the value of combining structured feature engineering with modular ML architectures to generate forecasts that are not only accurate, but strategically useful for retail planning and decision-making.

6.4 Contributions to Knowledge and Industry

This project offers several meaningful contributions to both academic research and industry practice in the domain of retail forecasting.A key novelty of this study lies in the operationalization of a dual-model architecture that integrates business decision logic directly into the model outputs—via the Promo Sensitivity Index and rolling horizon forecasts. While prior models focused on pure prediction, this research shifts towards actionable forecasting that feeds directly into procurement, campaign planning, and supplier segmentation. To the best of my knowledge, few existing retail forecasting systems combine interpretability, multi-output regression, and promotion-aware planning in a deployable framework.

Dual-Model Forecasting Strategy

A significant methodological contribution is the segmentation of sales data into distinct behavioral regimes—normal and promotional—and the development of specialized models for each. This approach challenges the conventional unified forecasting paradigm and introduces a replicable framework for modeling structurally divergent behaviors. The strategy is generalizable to other domains exhibiting regime-specific patterns, such as peak/off-peak energy consumption, routine/emergency hospital admissions, or weekday/weekend transport demand.

Structured Feature Engineering for Temporal Modeling

The project demonstrates that deep temporal patterns can be captured effectively using engineered features—such as lag variables, rolling means, and calendar encodings—without the need for complex neural architectures. This reinforces the growing body of literature supporting interpretable, ensemble-based time series forecasting. It also supports the adoption of explainable machine learning in environments where transparency is as critical as accuracy.

Multi-output Forecasting for Rolling Horizons

The application of multi-output regression for simultaneous multi-step forecasting provides a scalable alternative to recursive prediction frameworks. This technique improves consistency across forecast horizons and is particularly suited for retail planning cycles that require daily, weekly, or quarterly rolling forecasts. It offers an architecture that balances performance with computational feasibility for practical deployment.

Promotion Sensitivity Indexing

The design of a Promo Sensitivity Index, derived from model-based uplift comparisons, extends the forecasting system from a predictive tool to a prescriptive planning instrument. This innovation provides a quantifiable link between model outputs and strategic marketing decisions, facilitating targeted campaign planning, budget optimization, and ROI maximization.

Operational Readiness and Business Interpretability

The models were intentionally built for modularity, interpretability, and ease of integration into enterprise workflows. Their transparent structure and actionable outputs make them suitable for direct use by business teams in marketing, supply chain, and procurement. This work thus contributes a practical, deployable framework that bridges the gap between predictive modeling and real-world business decision-making.

6.5 Skills and PDP Reflection

This thesis project significantly contributed to my professional development. Prior to this research, my experience in machine learning was primarily theoretical. Through iterative modeling, hyperparameter tuning, and debugging, I enhanced my practical skills in Python, Scikit-learn, and data visualization. I also improved my ability to handle large, irregular datasets and gained proficiency in translating statistical insights into business strategies. The transition from building predictive models to generating operational insights through promo sensitivity analysis marked a key milestone in my analytical maturity.

References

[1] Benchmark Email. *Factors Affecting Retail Sales*. https://www.benchmarkemail.com/blog/factors-affecting-retail/. Accessed May 17, 2025. 2025.

[2] Microsoft Corporation. *Forecasting algorithms and models used in demand planning*. https://learn.microsoft.com/en-us/dynamics365/supply-chain/demand-planning/forecast-algorithm-types. Accessed May 17, 2025. 2025.

[3] Ebsta. *Limitations of Sales Forecasting and How to Solve Them*. https://www.ebsta.com/blog/sales/limitations-of-sales-forecasting-and-how-to-solve-them/. Accessed May 17, 2025. 2025.

[4] A. Bandara et al. "Retail Sales Forecasting using Machine Learning Algorithms". In: *Proceedings of the International Conference on Advancements in Computing (ICAC)*. 2021, pp. 214–219. DOI: 10.1109/ICAC53497.2021.9668911.

[5] A. Popovic, T. Hocenski, and D. Kukolj. "XGBoost Model for Accurate Retail Sales Forecasting". In: *45th International Convention on Information, Communication and Electronic Technology (MIPRO)*. 2022, pp. 1220–1225. DOI: 10.23919/MIPRO55190.2022.9803264.

[6] P. Maldonado, S. Heese, and L. Zhang. "Machine Learning for Retail Sales Forecasting: A Comparative Study of Models and Features". In: *Expert Systems with Applications* 188 (2021). DOI: 10.1016/j.eswa.2021.116012.

[7] A. Ahmed, M. Nasir, and S. W. Malik. "A Data-Driven Model for Multi-step Time Series Forecasting in Retail Using XGBoost". In: *Procedia Computer Science* 199 (2022), pp. 982–989. DOI: 10.1016/j.procs.2022.01.123.

[8] M. Ghosh, R. Sinha, and A. Dey. "Demand Forecasting in FMCG Retail Using XGBoost and LSTM: A Comparative Study". In: *Procedia Computer Science* 198 (2022), pp. 156–163.

[9] R. Wang, S. Wang, and M. A. Porter. "Promotion Impact Forecasting Using Gradient Boosted Trees". In: *Journal of Retail Analytics* 17.2 (2022), pp. 44–57.

[10] K. Kumar and A. Thomas. "Time Series Forecasting of Retail Sales Using Machine Learning Techniques". In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 13.1 (2022), pp. 123–130. DOI: 10.14569/IJACSA.2022.0130117.

[11] D. Tiwari and R. Singh. "Retail Forecasting Models with Seasonal Promotion Adjustment". In: *Advances in Computational Intelligence* 10.2 (2022), pp. 101–109.

[12] M. Ahmed and N. Sadiq. "Comparative Evaluation of Machine Learning Models for Forecasting Retail Sales in Promotion Contexts". In: *International Journal of Computer Applications* 183.22 (2021), pp. 11–17.

[13] B. Roy and A. Mitra. "Retail Sales Analysis and Prediction Using Machine Learning Techniques". In: *International Journal of Engineering Research and Technology* 9.7 (2021), pp. 1195–1202.

[14] S. Patel and V. Desai. "Analyzing Promo-Sales Uplift Using Machine Learning". In: *International Journal of Data Science and Analytics* 11.3 (2022), pp. 215–226. DOI: 10.1007/s41060-021-00314-9.

[15] Y. Zhang and T. Huang. "A Gradient Boosting Approach for Forecasting Promotional Demand in E-commerce". In: *Journal of Retail and Consumer Services* 63 (2021). DOI: 10.1016/j.jretconser.2021.102721.

[16] L. Zhu and H. Wang. "Sales Forecasting for Promotional Products: A Machine Learning Approach". In: *IEEE Access* 9 (2021), pp. 124567–124578. DOI: 10.1109/ACCESS.2021.3108723.

[17] T. Nguyen, K. Tran, and H. Le. "A Comparative Study of Forecasting Algorithms for Retail Sales". In: *Asian Journal of Computer Science and Technology* 11.4 (2021), pp. 41–47.

[18] J. Liu and Y. Chen. "Forecasting Promotional Sales in Retail Using Hybrid Time Series Models". In: *Journal of Business Research* 145 (2022), pp. 342–350. DOI: 10.1016/j.jbusres.2022.03.065.

[19] P. Bansal and R. Gupta. "A Comparative Study of Advanced Forecasting Models for Retail Demand". In: *International Journal of Forecasting and Analytics* 12.3 (2023), pp. 97–104.

[20] V. Raj and R. Kumar. "Machine Learning Based Sales Forecasting and Product Recommendation in Retail". In: *International Journal of Scientific Research in Computer Science* 9.3 (2022), pp. 205–210.

[21] H. Lin and W. Chen. "Promotion-Aware Retail Demand Forecasting Using Feature-Enriched Gradient Boosting". In: *IEEE Transactions on Industrial Informatics* 18.6 (2022), pp. 4183–4192. DOI: 10.1109/TII.2021.3119276.

[22] A. Verma and S. Gupta. "Effect of Promotions on Retail Forecasting Accuracy: A Data-Driven Perspective". In: *Journal of Retail Marketing Trends* 5.2 (2022), pp. 56–65.

[23] Anup Garg, S. Rawat, and A. Tiwari. "On Continuous Integration for Machine Learning Models Using MLOps". In: *International Journal of Engineering Research and Technology* 10.9 (2021), pp. 1155–1159.

[24] J. Jayakrishnan. "Artificial Intelligence in Retailing: Applications and Opportunities". In: *Artificial Intelligence and Digital Systems in Retailing*. Springer, 2021, pp. 23–38. DOI: 10.1007/978-3-030-68290-6_2.

[25] Yasuhiko Takefuji. "Beyond XGBoost and SHAP: Limitations in Interpretability of Ensemble Methods". In: *Journal of Artificial Intelligence and Data Science* 4.1 (2025), pp. 12–22.

[26] Gianluca Bontempi and Souhaib Ben Taieb. "A Strategy for Multiple-Step Ahead Time Series Forecasting". In: *Studies in Computational Intelligence* 330 (2011), pp. 143–162. DOI: 10.1007/978-3-642-21257-1_8.

[27] GeeksforGeeks. *Difference between Random Forest vs XGBoost*. https://www.geeksforgeeks.org/difference-between-random-forest-vs-xgboost/. Accessed May 17, 2025. 2025.

[28] Jason Brownlee. *Mastering Time Series Forecasting: From ARIMA to LSTM*. https://machinelearningmastery.com/mastering-time-series-forecasting-from-arima-to-lstm/. Accessed May 17, 2025. 2025.

[29] Rob Percival. *DeepAR Retail Forecasting (SageMaker Workshop)*. https://github.com/robperc/sagemaker-workshop/blob/master/deepar-retail-forecasting/deepar_retail_forecasting.ipynb. Accessed May 17, 2025. 2025.

[30] PyTorch Forecasting. *AR Tutorial*. https://pytorch-forecasting.readthedocs.io/en/v1.2.0/tutorials/ar.html. Accessed May 17, 2025. 2025.

Appendix

Gantt Chart

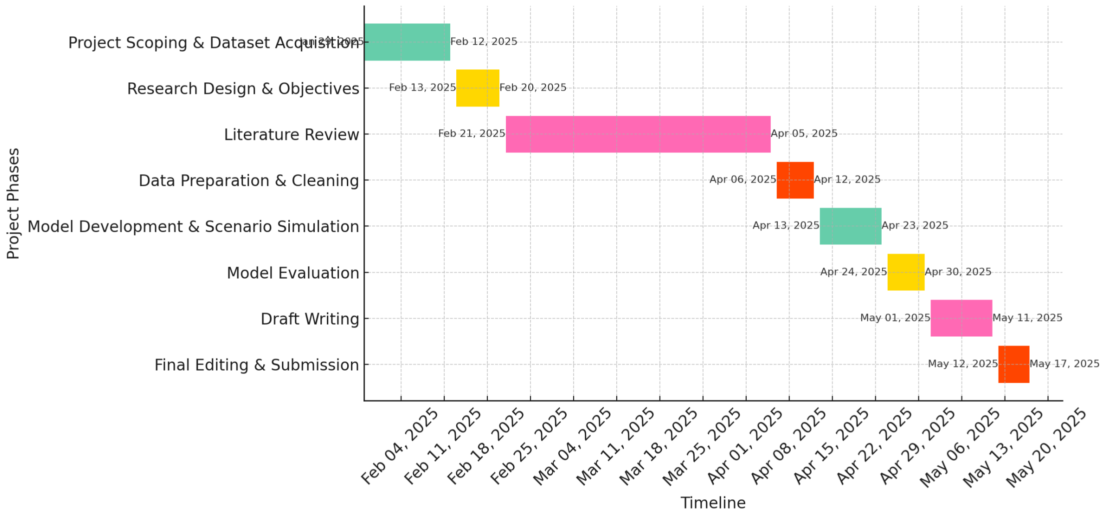

Figure 1: Gantt Chart

Python Scripts

Data Processing

```
# Data Processing.py
# Importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import missingno as msno
import numpy as np
import seaborn as sns

# Loading the dataset
data = pd.read_csv(r"C:\Users\ss\OneDrive\Desktop\Msc Project\Datasets
    \Warehouse_and_Retail_Sales.csv")

# Creating a datetime field
data['DATE'] = pd.to_datetime(data['YEAR'].astype(str) + '-' + data['
    MONTH'].astype(str))
data.sort_values('DATE', inplace=True)
```

```python
# Reordering columns
data = data[['DATE', 'YEAR', 'MONTH', 'SUPPLIER', 'ITEM CODE', 'ITEM
    DESCRIPTION',
             'ITEM TYPE', 'RETAIL SALES', 'RETAIL TRANSFERS', '
                WAREHOUSE SALES']]

# Descriptive stats
print(data.info())
print(data.describe())
print("Unique items:", data['ITEM CODE'].nunique())

# Handling null values
data['SUPPLIER'] = data['SUPPLIER'].fillna('UNKNOWN')
data.dropna(subset=['ITEM TYPE'], inplace=True)

# Feature engineering: total sales
data['TOTAL SALES'] = data['RETAIL SALES'] + data['WAREHOUSE SALES']
data = data[data['TOTAL SALES'] >= 0]

# Heatmap for numeric correlation
corr = data.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm')

# Time-series plot
data_sales = data.groupby('DATE')[['TOTAL SALES']].sum().reset_index()
data_sales.plot(x='DATE', y='TOTAL SALES', figsize=(12,6))

# Pie chart by item type
item_type_counts = data['ITEM TYPE'].value_counts()
labels = item_type_counts.index
sizes = item_type_counts.values
plt.pie(sizes, labels=labels, startangle=140, autopct='%1.1f%%')

# Boxplot to detect skewness
sns.boxplot(x=data['TOTAL SALES'])

# Seasonal decomposition
from statsmodels.tsa.seasonal import seasonal_decompose
monthly_sales = data.groupby(pd.Grouper(key='DATE', freq='M'))['TOTAL
    SALES'].sum()
decomposition = seasonal_decompose(monthly_sales, model='additive',
    period=12)
decomposition.plot()

# Outlier detection for promotional data
Q1 = data['TOTAL SALES'].quantile(0.25)
Q3 = data['TOTAL SALES'].quantile(0.75)
IQR = Q3 - Q1
lower_bound, upper_bound = Q1 - 1.5 * IQR, Q3 + 1.5 * IQR
Promo_data = data[(data['TOTAL SALES'] < lower_bound) | (data['TOTAL
    SALES'] > upper_bound)]

# Normal sales = data - Promo_data
Normal_sales_data = data[(data['TOTAL SALES'] >= lower_bound) & (data[
    'TOTAL SALES'] <= upper_bound)]

# Monthly sales visualization (normal vs promo)
```

```
normal_monthly = Normal_sales_data.groupby(Normal_sales_data['DATE'].
    dt.to_period('M'))['TOTAL SALES'].sum().reset_index()
promo_monthly = Promo_data.groupby(Promo_data['DATE'].dt.to_period('M'
    ))['TOTAL SALES'].sum().reset_index()
# Convert to timestamp
normal_monthly['DATE'] = normal_monthly['DATE'].dt.to_timestamp()
promo_monthly['DATE'] = promo_monthly['DATE'].dt.to_timestamp()

plt.plot(normal_monthly['DATE'], normal_monthly['TOTAL SALES'], label=
    'Normal')
plt.plot(promo_monthly['DATE'], promo_monthly['TOTAL SALES'], label='
    Promo')
plt.legend()

# Promo sensitivity analysis and visualization
promo_items = Promo_data['ITEM CODE'].unique()
normal_items = Normal_sales_data['ITEM CODE'].unique()
common_items = list(set(promo_items).intersection(set(normal_items)))

# Volatility analysis
supplier_monthly = Normal_sales_data.groupby(['SUPPLIER', 'DATE'])['
    TOTAL SALES'].sum().reset_index()
supplier_volatility = supplier_monthly.groupby('SUPPLIER')['TOTAL
    SALES'].agg(['mean', 'std']).reset_index()
supplier_volatility['SVI'] = supplier_volatility['std'] /
    supplier_volatility['mean']
```

Feature Engineering - Normal Sales

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

Normal_sales_data=pd.read_csv(r"C:\Users\ss\OneDrive\Desktop\Msc
    Project\Datasets\Normal_sales_data.csv")
Normal_sales_data['DATE'] = pd.to_datetime(Normal_sales_data['DATE'])

Normal_sales_data['YEAR'] = Normal_sales_data['DATE'].dt.year
Normal_sales_data['MONTH'] = Normal_sales_data['DATE'].dt.month
Normal_sales_data['QUARTER'] = Normal_sales_data['DATE'].dt.quarter
Normal_sales_data['day_of_week'] = Normal_sales_data['DATE'].dt.
    weekday
Normal_sales_data['week_of_year'] = Normal_sales_data['DATE'].dt.
    isocalendar().week

Normal_sales_data['SALES_LAG_1'] = Normal_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(1)
Normal_sales_data['SALES_LAG_3'] = Normal_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(3)
Normal_sales_data['SALES_LAG_6'] = Normal_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(6)

Normal_sales_data['SALES_ROLLING_MEAN_3'] = Normal_sales_data.groupby(
    'ITEM DESCRIPTION')['TOTAL SALES'].transform(lambda x: x.rolling(
    window=3).mean())
```

```
Normal_sales_data = pd.get_dummies(Normal_sales_data, columns=['ITEM
    TYPE'], prefix='TYPE')

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Normal_sales_data['SUPPLIER_ENCODED'] = le.fit_transform(
    Normal_sales_data['SUPPLIER'])

median_sales = Normal_sales_data.groupby('ITEM DESCRIPTION')['TOTAL
    SALES'].median().reset_index()
median_sales.rename(columns={'TOTAL SALES': 'MEDIAN_PRODUCT_SALES'},
    inplace=True)
Normal_sales_data = Normal_sales_data.merge(median_sales, on='ITEM
    DESCRIPTION', how='left')

normal_data_unstack = Normal_sales_data.groupby(['YEAR', 'MONTH']).
    size().unstack(fill_value=0)

Normal_sales_data.to_csv('Normal_sales_data_with_features_final.csv',
    index=False)
```

Feature Engineering - Promotional Sales

```
import numpy as np
import pandas as pd

Promo_sales_data=pd.read_csv(r"C:\Users\ss\OneDrive\Desktop\Msc
    Project\Datasets\Promo_sales_data.csv")
Promo_sales_data['DATE'] = pd.to_datetime(Promo_sales_data['DATE'])

Promo_sales_data['YEAR'] = Promo_sales_data['DATE'].dt.year
Promo_sales_data['MONTH'] = Promo_sales_data['DATE'].dt.month
Promo_sales_data['QUARTER'] = Promo_sales_data['DATE'].dt.quarter

Promo_sales_data['SALES_LAG_1'] = Promo_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(1)
Promo_sales_data['SALES_LAG_3'] = Promo_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(3)
Promo_sales_data['SALES_LAG_6'] = Promo_sales_data.groupby('ITEM
    DESCRIPTION')['TOTAL SALES'].shift(6)

Promo_sales_data['SALES_ROLLING_MEAN_3'] = Promo_sales_data.groupby('
    ITEM DESCRIPTION')['TOTAL SALES'].transform(lambda x: x.rolling(
    window=3).mean())

Promo_sales_data = pd.get_dummies(Promo_sales_data, columns=['ITEM
    TYPE'], prefix='TYPE')

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Promo_sales_data['SUPPLIER_ENCODED'] = le.fit_transform(
    Promo_sales_data['SUPPLIER'])

median_sales = Promo_sales_data.groupby('ITEM DESCRIPTION')['TOTAL
    SALES'].median().reset_index()
median_sales.rename(columns={'TOTAL SALES': 'MEDIAN_PRODUCT_SALES'},
    inplace=True)
```

```
Promo_sales_data = Promo_sales_data.merge(median_sales, on='ITEM
    DESCRIPTION', how='left')

normal_data_unstack = Promo_sales_data.groupby(['YEAR', 'MONTH']).size
    ().unstack(fill_value=0)

# Promo_sales_data.to_csv('Promo_sales_data_with_features.csv', index=
    False)
```

Modelling

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, TimeSeriesSplit,
     GridSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error,
    r2_score
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from xgboost import XGBRegressor, plot_importance
from sklearn.multioutput import MultiOutputRegressor
import matplotlib.pyplot as plt

# Load datasets
normal_data = pd.read_csv("Normal_sales_data_with_features_final.csv")
promo_data = pd.read_csv("Promo_sales_data_with_features.csv")

# Encode categorical features
for col in normal_data.select_dtypes(include=['object']).columns:
    normal_data[col] = LabelEncoder().fit_transform(normal_data[col].
        astype(str))
for col in promo_data.select_dtypes(include=['object']).columns:
    promo_data[col] = LabelEncoder().fit_transform(promo_data[col].
        astype(str))

# Time series split and scaling
X = normal_data.drop(['DATE', 'TOTAL SALES'], axis=1)
y = normal_data['TOTAL SALES']
tscv = TimeSeriesSplit(n_splits=5)
for train_idx, test_idx in tscv.split(X):
    X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
    y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
X_train_scaled = MinMaxScaler().fit_transform(X_train)
X_test_scaled = MinMaxScaler().fit(X_train).transform(X_test)

# Train model for normal sales
xgb_model_normal = XGBRegressor(n_estimators=300, learning_rate=0.1,
    max_depth=7)
xgb_model_normal.fit(X_train_scaled, y_train)
y_pred = xgb_model_normal.predict(X_test_scaled)

print("Normal Sales XGBoost Performance:")
print(f"MAE: {mean_absolute_error(y_test, y_pred)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred))}")
```

```
print(f"R  : {r2_score(y_test, y_pred)}")

# Grid search (optional)
grid_search = GridSearchCV(XGBRegressor(), {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'n_estimators': [100, 200, 300]
}, cv=3, scoring='neg_mean_squared_error')
grid_search.fit(X_train_scaled, y_train)

# Promo modelling
X_promo = promo_data.drop(['DATE', 'TOTAL SALES'], axis=1)
y_promo = promo_data['TOTAL SALES']
for train_idx, test_idx in tscv.split(X_promo):
    X_train_promo, X_test_promo = X_promo.iloc[train_idx], X_promo.
        iloc[test_idx]
    y_train_promo, y_test_promo = y_promo.iloc[train_idx], y_promo.
        iloc[test_idx]
X_train_promo_scaled = MinMaxScaler().fit_transform(X_train_promo)
X_test_promo_scaled = MinMaxScaler().fit(X_train_promo).transform(
    X_test_promo)

xgb_model_promo = XGBRegressor(n_estimators=100, learning_rate=0.1,
    max_depth=3)
xgb_model_promo.fit(X_train_promo_scaled, y_train_promo)
y_pred_promo = xgb_model_promo.predict(X_test_promo_scaled)

print("\nPromo Sales XGBoost Performance:")
print(f"MAE: {mean_absolute_error(y_test_promo, y_pred_promo)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test_promo, y_pred_promo))
    }")
print(f"R  : {r2_score(y_test_promo, y_pred_promo)}")

# Scatter plot function
def scatter_actual_vs_predicted(y_test, y_pred, title):
    plt.figure(figsize=(6, 6))
    plt.scatter(y_test, y_pred, alpha=0.6, edgecolor='k')
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()
        ], 'r--')
    plt.title(title)
    plt.grid(True)
    plt.tight_layout()
    plt.show()

scatter_actual_vs_predicted(y_test, y_pred, "Actual vs Predicted -
    Normal Sales")
scatter_actual_vs_predicted(y_test_promo, y_pred_promo, "Actual vs
    Predicted - Promo Sales")

# Multistep targets
for df in [normal_data, promo_data]:
    df.sort_values(by=['ITEM CODE', 'DATE'], inplace=True)
    df['SALES_t+1'] = df.groupby('ITEM CODE')['TOTAL SALES'].shift(-1)
    df['SALES_t+2'] = df.groupby('ITEM CODE')['TOTAL SALES'].shift(-2)
    df['SALES_t+3'] = df.groupby('ITEM CODE')['TOTAL SALES'].shift(-3)

# Multi-step regression
def multi_step_forecast(df):
```

```
    df = df.dropna(subset=['SALES_t+1', 'SALES_t+2', 'SALES_t+3'])
    X = df.drop(['DATE', 'TOTAL SALES', 'SALES_t+1', 'SALES_t+2', '
        SALES_t+3'], axis=1)
    y = df[['SALES_t+1', 'SALES_t+2', 'SALES_t+3']]
    split_idx = int(len(X) * 0.8)
    model = MultiOutputRegressor(XGBRegressor(n_estimators=100))
    model.fit(X.iloc[:split_idx], y.iloc[:split_idx])
    return model.predict(X.iloc[split_idx:]), y.iloc[split_idx:]

y_pred_norm, y_test_norm = multi_step_forecast(normal_data)
y_pred_promo, y_test_promo = multi_step_forecast(promo_data)

# Forecast comparison
forecast_comparison = pd.DataFrame(y_pred_norm, columns=['Month1', '
    Month2', 'Month3'])
forecast_comparison[['Promo_M1', 'Promo_M2', 'Promo_M3']] =
    y_pred_promo
for i in range(1, 4):
    forecast_comparison[f'Diff_M{i}'] = forecast_comparison[f'Promo_M{
        i}'] - forecast_comparison[f'Month{i}']
    forecast_comparison[f'Pct_Diff_M{i}'] = (forecast_comparison[f'
        Diff_M{i}'] / forecast_comparison[f'Month{i}']) * 100

forecast_comparison['Avg_Pct_Diff'] = forecast_comparison[[f'
    Pct_Diff_M{i}' for i in range(1, 4)]].mean(axis=1)
forecast_comparison['Promo_Sensitivity'] = pd.cut(forecast_comparison[
    'Avg_Pct_Diff'],
    bins=[-np.inf, 10, 30, np.inf], labels=['Low', 'Moderate', 'High'
        ])

# Visualization
forecast_comparison['Promo_Sensitivity'].value_counts().plot(
    kind='pie', autopct='%1.1f%%', startangle=90, figsize=(6, 6),
        colors=['lightgreen', 'gold', 'salmon']
)
plt.title('Distribution of Products by Promotion Sensitivity')
plt.ylabel('')
plt.show()

# Simulated Recommendations
forecast_comparison['Promo_Uplift_%'] = np.random.randint(5, 45, size=
    len(forecast_comparison))
forecast_comparison['Sensitivity'] = pd.cut(
    forecast_comparison['Promo_Uplift_%'],
    bins=[-np.inf, 9, 29, np.inf],
    labels=['Low', 'Moderate', 'High']
)

def action_strategy(row):
    if row['Sensitivity'] == 'High':
        return 'Promote monthly or during key events'
    elif row['Sensitivity'] == 'Moderate':
        return 'Use selective/seasonal promotions'
    return 'Keep stocked, no frequent promotion'

forecast_comparison['Action'] = forecast_comparison.apply(
    action_strategy, axis=1)
```

```python
# Top products chart
top_sales = forecast_comparison.sort_values(by='Month1', ascending=
    False).head(5)
top_sales[['Month1', 'Month2', 'Month3']].plot(kind='bar', stacked=
    True, figsize=(10, 6))
plt.title('Quarterly Sales Forecast for Top 5 Products')
plt.ylabel('Forecasted Sales Volume')
plt.xlabel('Product Index')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```