Acad Year
24/25

PROJECT NO.
C073

DEVELOPMENT OF A SUPPLY CHAIN PERFORMANCE
DASHBOARD USING TABLEAU

DEVELOPMENT OF A SUPPLY CHAIN PERFORMANCE DASHBOARD USING TABLEAU

Kezia Putri Ersan

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Year (e.g. 2024/2025)

DEVELOPMENT OF A SUPPLY CHAIN PERFORMANCE DASHBOARD USING TABLEAU

SUBMITTED

BY

KEZIA PUTRI ERSAN

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

A final year project report
presented to
Nanyang Technological University
in partial fulfilment of the
requirements for the
Degree of Bachelor of Engineering (Mechanical Engineering)
Nanyang Technological University

ABSTRACT

In the face of increasing complexity and unpredictability in global supply chains, the demand for data-driven tools that enable real-time visibility and informed decision-making has intensified. This project addresses this need through the development of an integrated supply chain performance dashboard that consolidates key metrics across inventory management, supplier reliability, and demand forecasting into a cohesive visual analytics platform using Tableau. The solution is built upon an enhanced dataset generated through a combination of real-world data and synthetic data simulation, incorporating features such as estimated prices and simulated inventory levels. To support the forecasting component, historical monthly sales patterns were synthesized to reflect seasonal and category-based trends, enabling robust time series analysis. Forecasting models including Facebook Prophet and Exponential Smoothing (ETS) were trained and evaluated using performance metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE). Model comparison revealed trade-offs in accuracy and adaptability across product categories, guiding the selection of forecasting logic implemented in the dashboard. The resulting dashboards - Inventory Management, Supplier Performance, and Demand Forecasting - were designed with interactivity, clarity, and operational relevance in mind, providing stakeholders with filterable, real-time insights across stock keeping unit (SKU), category, and supplier levels. This project demonstrates the value of integrating data engineering, statistical modelling, and business intelligence tools in enabling proactive supply chain management, while also highlighting the role of synthetic data generation in addressing limitations of real-world datasets

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Associate Professor Rajesh Piplani, from the School of Mechanical and Aerospace Engineering, Nanyang Technological University, for his invaluable guidance and mentorship throughout the course of this final year project. As someone initially unfamiliar with the domain of supply chain management, I greatly benefited from his deep expertise and constructive feedback, which significantly shaped my understanding of the field and the direction of this work.

I also extend my appreciation to the School of Mechanical and Aerospace Engineering for providing the academic environment and resources necessary to undertake this project. The knowledge and skills acquired throughout the curriculum laid a strong foundation for the successful completion of this work.

Finally, I would like to thank my family and friends for their unwavering support, encouragement, and patience during this academic journey. Their belief in me has been a constant source of motivation and strength.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Background

In today's dynamic business landscape, supply chain management has emerged as a critical function for organizational success [1], [2]. In the era of digital transformation, supply chain management (SCM) has become increasingly data-driven as businesses seek to enhance operational efficiency and responsiveness. Modern supply chains are intricate networks that require collaborative management and global optimization [3]. These networks encompass a variety of coordinated organizations that collaborate in diverse activities. Effective SCM is a key factor in achieving competitive advantage, particularly in today's interconnected global markets. However, traditional supply chain monitoring methods, often involving manual tracking and spreadsheets, are inadequate for managing the complexities of modern supply chains. These conventional methods are prone to errors, lack agility, and offer limited visibility into real-time operations.

To address these challenges, businesses are turning to advanced data analytics and real-time monitoring solutions. Big Data Analytics (BDA) has revolutionized SCM by providing actionable insights, improving visibility across the supply chain network, reducing operational costs, and enhancing decision-making [4],[5]. Nevertheless, the massive volume of supply chain data presents difficulties in data organization and analysis, creating the need for robust visualization tools. Visual Analytics (VA) platforms like Tableau are instrumental in transforming raw data into intuitive dashboards that present real-time insights into key performance indicators (KPIs) [6]. By visualizing supply chain performance, businesses can quickly identify bottlenecks, inefficiencies, and opportunities for improvement. By consolidating data from various sources, these dashboards enable decision-makers to swiftly identify bottlenecks, monitor supplier performance, predict demand fluctuations, and optimize inventory levels.

This project leverages Tableau to design a supply chain dashboard that integrates data from multiple sources. The dashboard will visualize crucial KPIs, focusing on inventory management, supplier performance, and demand forecasting, providing actionable insights for businesses seeking to enhance their supply chain operations.

1.2 Objectives

The main objective of this project is to develop a comprehensive and interactive supply chain performance dashboard using Tableau that supports data-driven decision-making. The dashboard is designed to provide real-time visibility into inventory levels, turnover rates, and stockout risks, enabling proactive inventory management. In addition, it aims to evaluate supplier performance by visualizing critical metrics such as on-time delivery rate, defect rate, and quantity accuracy. Another key objective is to generate demand forecasts using statistical and machine learning models, allowing businesses to anticipate future sales trends and plan inventory accordingly. By integrating these three functional areas into a unified platform, the dashboard seeks to empower supply chain managers with actionable insights for operational improvement.

1.3 Project Scope & Deliverables

The scope of this project encompasses the design and implementation of an end-to-end supply chain dashboard that enables relevant stakeholders to monitor key performance indicators, identify inefficiencies, and make informed, d\ata-backed decisions. The project begins with data collection and preparation, using publicly available data from a United States-based beverage retailer as a representative case study to simulate real-world supply chain operations. This foundational dataset is further enhanced through synthetic data generation, incorporating price estimation, lead time variability, inventory behaviour, and order-level performance attributes.

The core deliverables include three Tableau dashboards - Inventory Management, Supplier Performance, and Demand Forecasting - each tailored to address specific decision-making needs. In support of the forecasting function, the project applies time series models such as Facebook Prophet and Exponential Smoothing (ETS) to produce category-level demand predictions, which are evaluated using Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE). The final dashboards present these insights through intuitive, interactive visualizations that facilitate quick and effective decision-making. Although the project is centred on the beverage retail sector, the dashboard framework is designed to be adaptable across other industries, such as manufacturing, logistics, e-commerce, and etc. While the current implementation does not involve live data integration, the methodology is extendable to real-time business environments with appropriate technical enhancements.

Chapter 2: Literature Review

2.1 Existing Supply Chain Performance Dashboards

A comprehensive review of existing literature reveals a growing trend towards the adoption of supply chain performance dashboards [7]. Hence, The use of dashboards in SCM has become increasingly prevalent as organizations seek to enhance operational transparency, responsiveness, and efficiency. A supply chain dashboard serves as a visual interface that consolidates KPIs and real-time metrics, enabling decision-makers to monitor processes across procurement, inventory, logistics, and customer fulfilment functions. This section reviews several well-established SCM dashboard solutions currently available in the market, highlighting their capabilities, visual approaches, and limitations.

One of the most recognized platforms in this domain is *SAP Integrated Business Planning (SAP IBP)*[9], which provides a comprehensive set of analytics tools for sales and operations planning (S&OP), demand forecasting, inventory optimization, and supply planning [8]. SAP IBP dashboards integrate real-time data from enterprise resource planning (ERP) systems and offer extensive customization and predictive modelling capabilities. The visual interface supports alerts, scenario planning, and KPI tracking, making it well-suited for global enterprises with complex supply networks.

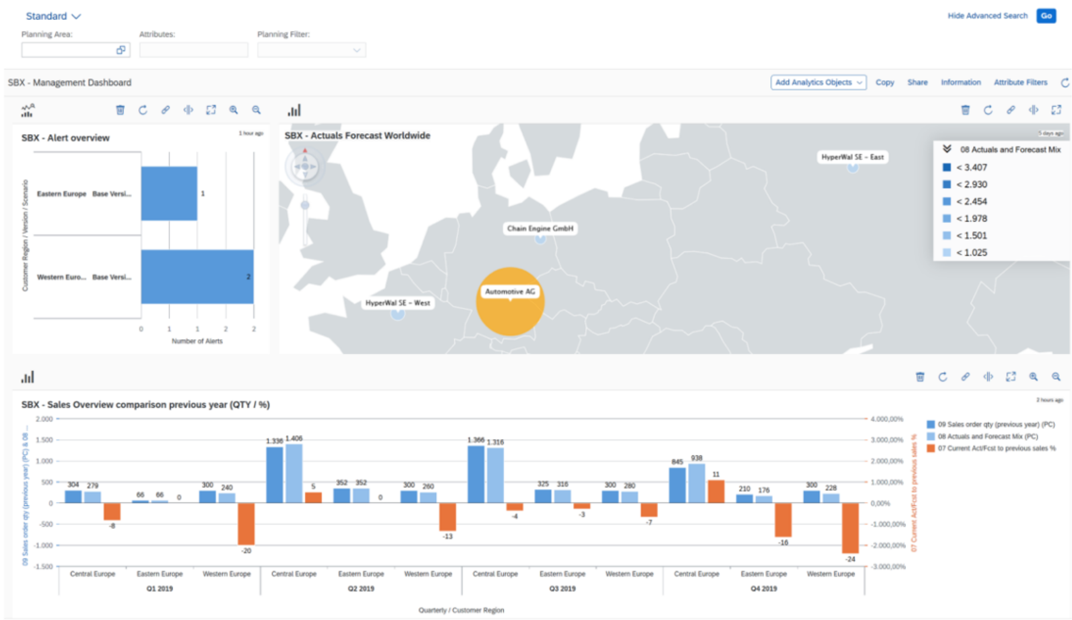


Figure 1. Sample SAP IBP Dashboard

Another widely adopted solution is *Oracle Fusion Cloud Supply Chain & Manufacturing* [10], which includes dashboards for procurement analytics, inventory performance, and supplier evaluation. These dashboards enable users to assess supplier delivery trends, lead time variability, and order accuracy, often through embedded machine learning (ML) insights. Oracle's dashboards are particularly strong in integrating financial and operational data, supporting strategic supply chain planning across departments .

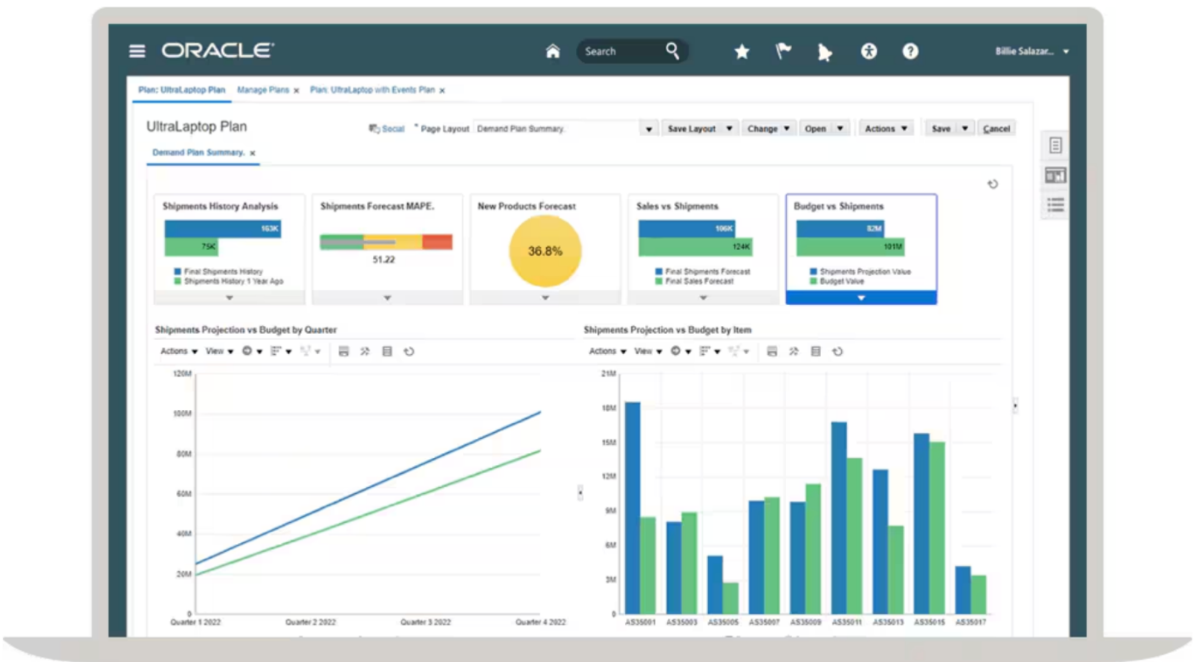


Figure 2. Sample Oracle SCM Cloud Dashboard

Despite their strengths, existing commercial dashboards often suffer from limited flexibility in customization for smaller businesses or specialized supply chain models. Furthermore, many platforms require substantial investment in licensing and implementation, which may not be feasible for medium-sized enterprises. In addition, while machine learning capabilities are increasingly embedded, interpretability and ease of use remain concerns for non-technical users.

2.2 Tools for Customized SCM Dashboards

Tableau, while not a dedicated supply chain software, is frequently adopted by companies for developing custom SCM dashboards. Its flexibility allows users to connect to multiple data sources and design bespoke visualizations tailored to specific business functions. Case studies from companies such as Coca-Cola and Lenovo have demonstrated the use of Tableau dashboards to monitor shipment delays, inventory turnover, and supplier responsiveness in real time [11], [12]. Tableau's strength lies in its ability to provide rapid visual feedback, intuitive drill-downs, and integration with machine learning pipelines.

Microsoft Power BI also offers significant capabilities in supply chain dashboard development. Leveraging the Microsoft ecosystem, Power BI enables integration with Excel, SQL Server, and Azure-based services. Organizations use it to visualize transportation KPIs, warehouse performance, and procurement data. Power BI dashboards are particularly effective for operational reporting and periodic review meetings due to their interactivity and report-building features [13].

The review of existing dashboard solutions reveals the growing importance of visual analytics in SCM and underscores the need for adaptable, interpretable, and cost-effective tools that can be tailored to specific industries or organizations. The dashboard developed in this project aims to address this need by providing a focused, lightweight, and customizable solution centred on three key SCM domains: inventory management, supplier performance, and demand forecasting.

2.3 Key Supply Chain Performance Indicators/Metrics

The selection of KPIs is crucial for the success of any dashboard implementation, necessitating a meticulous alignment with overarching business objectives and the strategic imperatives of the supply chain [14]. Companies leverage these metrics to monitor inventory levels, evaluate supplier performance, and predict future demand. The selection of appropriate KPIs is crucial as it provides actionable insights for decision-making and operational improvements.

2.3.1 Inventory Management Metrics and KPIs

Inventory management is a critical aspect of supply chain management, directly influencing service levels, operational costs, and overall business performance. Organizations often track inventory using quantitative KPIs to maintain balance between inventory availability and holding costs. Some metrics in this area include:

Inventory Turnover Ratio (ITR): The Inventory Turnover Ratio measures how often a company sells and replaces its inventory over a given period. A higher ITR typically indicates efficient inventory management and product demand, while a lower ratio may suggest overstocking or slow-moving inventory.

$$ITR = \frac{Cost\ of\ Goods\ Sold\ (COGS)}{Average\ Inventory} \tag{1}$$

Studies have shown that industries with perishable or seasonal products, such as food and beverages, tend to target higher turnover rates compared to durable goods industries [15].

Days of Inventory on Hand (DOH): DOH is a complementary metric that measures how many days a company can operate using the inventory available. It provides insights into inventory liquidity and supply chain responsiveness.

$$DOH = \frac{Average\ Inventory}{Cost\ of\ Goods\ Sold\ (COGS)} * 365 \tag{2}$$

A lower DOH indicates efficient inventory management, while a higher DOH may imply excess stock or slow-moving inventory [16].

Stockout Rate: The Stockout Rate measures the percentage of order requests that cannot be fulfilled due to insufficient inventory. High stockout rates can lead to lost sales, reduced customer satisfaction, and reputational damage.

$$Stockout\ Rate = \frac{Number\ of\ Stockouts}{Total\ Order\ Requests} * 100\% \tag{3}$$

Literature suggests that businesses in industries with fluctuating demand patterns, such as retail and e-commerce, often experience higher stockout rates during promotional or seasonal events [17]

Carrying Cost of Inventory: Carrying costs refer to the expenses associated with holding and maintaining inventory. This includes warehousing, insurance, depreciation, and obsolescence. Efficient inventory management reduces carrying costs, thereby improving profitability [18].

$$Carrying\ Cost\ of\ Inventory = \frac{Inventory\ Holding\ Sum}{Total\ Inventory\ Value} * 100 \tag{4}$$

Service Level: Service Level measures a company's ability to meet customer demand without facing stockouts. It is a critical indicator in ensuring customer satisfaction and brand loyalty. A typical benchmark in retail is a service level of 95% or above [19],[20].

$$Service\ Level = \frac{Numbers\ of\ Orders\ Fulfilled\ On\ Time}{Total\ Number\ of\ Orders} * 100 \tag{5}$$

A comprehensive review by Akyuz and Erkan emphasizes the importance of these metrics in evaluating supply chain performance and highlights the need for organizations to adopt a balanced set of KPIs tailored to their specific contexts [21].

2.3.2 Supplier Performance Evaluation Metrics

Supplier performance evaluation metrics are crucial components of supply chain management, offering essential insights into the effectiveness and reliability of suppliers [22]. Evaluating suppliers using well-defined KPIs enables companies to identify reliable partners, mitigate risks, and strengthen relationships. Key metrics in this domain include:

On-Time Delivery (OTD) Rate: OTD Rate measures the percentage of orders delivered by suppliers within the agreed timeline. It is a widely used indicator of supplier reliability. Late deliveries often disrupt production schedules and increase operational costs.

$$OTD\ Rate = \frac{Number\ of\ On-Time\ Deliveries}{Total\ Number\ of\ Deliveries} * 100\ \% \qquad (6)$$

Research highlights that organizations with high OTD rates often experience fewer production delays and improved customer satisfaction [21].

Defect Rate: The Defect Rate evaluates the quality of products delivered by suppliers by measuring the percentage of defective items. A low defect rate is indicative of strong quality control processes and supplier reliability.

$$Defect\ Rate = \frac{Number\ of\ Defective\ Items}{Total\ Items\ Delivered} * 100\ \% \qquad (7)$$

Companies in highly regulated industries, such as pharmaceuticals and aerospace, typically enforce stringent defect rate targets [23].

Quantity Accuracy: Quantity Accuracy tracks the precision of suppliers in delivering the correct order quantities. Inaccurate deliveries can result in excess inventory, operational inefficiencies, or production halts.

$$Quantity\ Accuracy = \frac{Number\ of\ Accurate\ Deliveries}{Total\ Deliveries} * 100\ \% \qquad (8)$$

Supplier Lead Time Variability: Lead time variability measures inconsistencies in the time suppliers take to deliver products, which can be computed by using the standard deviation of lead times. Reducing lead time variability enhances planning accuracy and minimizes inventory carrying costs [24].

$$Standard\ Deviation\ (\sigma) = \sqrt{\frac{\sum((x_i - \mu)^2)}{N}} \qquad (9)$$

- x_i: Individual Lead Time

- μ: Average (mean) Lead Time
- N : Total Number of Lead Times

2.4 Time Series Forecasting Models in Supply Chain Management

Within the realm of supply chain management, time series forecasting models are indispensable tools for predicting future trends and patterns based on historical data [25]. Accurate demand forecasting is fundamental to effective SCM, influencing decisions related to inventory management, production planning, and resource allocation.

Autoregressive Integrated Moving Average (ARIMA)

A versatile model that captures the temporal dependencies in time series data, widely used for forecasting demand, inventory levels, and other critical supply chain metrics [26]. They are particularly useful for datasets where trends and seasonality need to be explicitly modelled. The general ARIMA(p,d,q) model can be mathematically expressed as:

$$\phi_p(B)(1-B)^d y_t = \theta_q(B)\,\epsilon_t \tag{10}$$

- p: order of the autoregressive part
- q: order of the moving-average process
- d: order of the differencing
- y_t: actual value at time t
- B: backshift operator; that is, $B\,y_t = y_{t-1}$
- $1-\ \phi_1 B - \ldots - \phi_p B^p$: autoregressive operator
- $1-\ \phi_1 B - \ldots - \phi_p B^q$: moving-average component
- ϵ_t: independent disturbance, also called the random error

ARIMA is particularly effective for stationary time series and can model a wide variety of temporal patterns. However, it requires careful parameter tuning and is limited in handling complex seasonality without extensions such as Seasonal ARIMA (SARIMA).

Exponential Smoothing (ETS)

Exponential smoothing techniques offer a suite of forecasting methods adept at modelling time series data by assigning exponentially decreasing weights to past observations, where more recent data points are given higher significance [27]. This statistical method models time series data by considering trends and seasonality, making it suitable for data with consistent patterns over time. ETS models are valued for their simplicity and effectiveness in short-term forecasting. [28]. The ETS framework, developed by Hyndman et al., includes three components: Error, Trend, and Seasonality, which can each be additive (A), multiplicative (M), or none (N) [29].

$$Level: \quad l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \tag{11}$$

$$Trend: \quad b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \tag{12}$$

$$Seasonality: \quad s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{13}$$

$$Forecast: \quad \hat{y}_t + h = l_t + hb_t + st + h - m(k+1) \tag{14}$$

- l_t: level
- b_t: trend
- s_t: seasonal component
- α, β, γ : smoothing parameters
- m: seasonality period
- h : forecast horizon

ETS models are effective for capturing level, trend, and seasonal patterns and are suitable for both additive and multiplicative seasonality. Their interpretability and automatic model selection make them practical for SCM forecasting, although they may struggle with irregular or highly non-linear trends.

Facebook Prophet

Facebook Prophet is an additive time series forecasting model designed for handling business time series data that exhibit strong seasonal effects, holiday influences, and trend changepoints [30]. The model decomposes a time series into the following components:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{15}$$

- $g(t)$: piecewise linear or logistic growth trend
- $s(t)$: periodic seasonality modelled with Fourier series
- $h(t)$: effects of holidays or special events
- ϵ_t: error term

Prophet models seasonality using Fourier terms:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos \frac{2n\pi t}{P} + b_n \sin \frac{2n\pi t}{P} \right) \tag{16}$$

Where P is the period (e.g., 365.25 for yearly seasonality) and N is the number of Fourier terms.

Prophet's strength lies in its robustness to missing data, ability to capture multiple seasonalities, and intuitive parameter tuning. It is particularly suited for automated business applications where interpretability, scalability, and responsiveness to irregular trends are important. However, it may underperform in cases with limited historical data or highly non-linear dynamics outside of its decomposition structure.

A comparative study by Makridakis et al. evaluates the performance of these models, noting that while no single method consistently outperforms others across all scenarios, the choice of model should be guided by the specific characteristics of the data and the forecasting context [31]. In the context of supply chain demand forecasting, each model offers unique advantages. ARIMA is ideal for short-term forecasting of stationary series but requires manual tuning. ETS models provide a more automated approach with strong performance on structured seasonal data. Prophet stands out in scenarios involving complex seasonality, holidays, and changepoints, and is especially useful when interpretability and automation are prioritized.

Selecting the appropriate model depends on the characteristics of the underlying data, required forecast horizon, interpretability needs, and the operational context of the supply chain. This project evaluates both ETS and Prophet in terms of forecasting accuracy using metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE), discussed in detail in Chapter 3.

Chapter 3: Methodology

3.1 Data Sources

The primary data source used for this project was obtained from the U.S. Government's open data portal, titled the *Warehouse and Retail Sales Dataset* [32]. This dataset contains transactional records from a US-based beverage retailer, providing valuable insights into the sales, inventory, and supplier performance of the company.

3.1.1 Original Dataset Overview

The raw dataset comprised the following columns:

Column Name	Description	API Field Name	Data Type
# YEAR	Calendar Year	calendar_year	Number
# MONTH	Month	cal_month_num	Number
Tт SUPPLIER	Supplier Name	supplier	Text
Tт ITEM CODE	Item code	item_code	Text
Tт ITEM DESCRIPTION	Item Description	item_description	Text
Tт ITEM TYPE	Item Type	item_type	Text
# RETAIL SALES	Cases of product sold from DLC dispensaries	rtl_sales	Number
# RETAIL TRANSFERS	Cases of product transferred to DLC dispensaries	rtl_transfers	Number
# WAREHOUSE SALES	Cases of product sold to MC licensees	whs_sales	Number

Figure 3. Raw Dataset Overview

While the dataset offered a comprehensive overview of sales performance, it presented several limitations for this project use-case which we have discovered through data exploration which will be discussed further in 3.2, including but not limited to:

- Missing Data: Significant gaps were found in the data, including missing months and incomplete sales records.
- Lack of Critical Fields: The original dataset did not provide essential information such as product prices, inventory levels, and supplier performance metrics. These had to be estimated or generated using assumptions and secondary data.
- Duplicate Records: Multiple instances of the same sales transaction were detected, particularly for specific SKUs and months.

To address these challenges, extensive data cleaning, transformation, and synthetic data generation were applied. After preprocessing, multiple datasets were derived to serve specific analysis and visualization purposes within Tableau.

3.1.2 Final Processed Datasets

Table 1. Processed Datasets Used in Tableau

Dataset Name	Purpose	Size	Columns
main.csv	Main dataset containing cleaned sales data for 2024	150,454 rows	`Month`, `SKU`, `Description`, `Category`, `Supplier`, `Sales`
price_info.csv	Provides product pricing information for each SKU	26,313 rows	`SKU`, `Category`, `Average Category Cost`, `Unit Price`
inventory_info.csv	Contains detailed inventory data for each SKU and month	315,756 rows (26,313 SKUs × 12 months)	`SKU`, `Category`, `Month`, `Monthly Sales`, `Beginning Inventory`, `Ending Inventory`, `Stockout Flag`, `Average Monthly Inventory`
orders_info.csv	Represents supplier order data for performance analysis, each row representing every order placed to supplier	170,000 rows	`SKU`, `Category`, `Supplier`, `Order Placed Date`, `Availability`, `Expected Delivery Date`, `Order Quantity`, `Actual Delivery Date`, `Quantity Received`, `Quantity Accuracy`, `Delivery Status`, `Defect Flag`
leadtime_info.csv	Contains expected lead time data for each SKU	26,313 rows	`SKU, Category, Lead Time (days)`

3.2 Data Exploration & Preprocessing

This section will cover the data exploration process to understand its structure, identify data quality issues, and perform data cleaning to ensure accurate and reliable analysis.

The initial stage of the project involved loading the dataset and conducting an exploratory data analysis (EDA) to gain insights into the structure, quality, and characteristics of the data. Several Python libraries including *Pandas* for data manipulation and analysis, *NumPy* for numerical operations and *Matplotlib* and *Seaborn* for data visualization were used for data exploration.

```
YEAR                  int64
MONTH                 int64
SUPPLIER             object
ITEM CODE            object
ITEM DESCRIPTION     object
ITEM TYPE            object
RETAIL SALES        float64
RETAIL TRANSFERS    float64
WAREHOUSE SALES     float64
```

Figure 4. Data Types Summary of Raw Dataset

	YEAR	MONTH	RETAIL SALES	RETAIL TRANSFERS	WAREHOUSE SALES
count	138638.0	138638.000000	138638.000000	138638.000000	138638.000000
mean	2024.0	6.025823	6.925888	6.906926	25.558478
std	0.0	3.165113	30.781417	30.142878	249.045532
min	2024.0	1.000000	-1.170000	-23.000000	-7800.000000
25%	2024.0	3.000000	0.000000	0.000000	0.000000
50%	2024.0	6.000000	0.240000	0.000000	1.000000
75%	2024.0	9.000000	2.890000	3.000000	6.000000
max	2024.0	11.000000	1752.450000	1470.170000	17810.870000

Figure 5. Summary Statistics of Raw Dataset's Numerical Features

Through the initial exploration, several data quality issues were identified, few selected issues are discussed in further detail in sections 3.2.1 – 3.2.3.

3.2.1 Handling Missing Data

Missing values were identified in multiple columns, particularly within the Supplier field as 96 rows were found with missing supplier information. While missing sales data or inventory data could significantly impact analysis, the missing supplier data was minimal compared to the total dataset. Hence, the affected rows were simply dropped using *dropna()*. Alternative approaches like imputation using the most frequent supplier name could have been applied if the missing data percentage was higher.

3.2.2 Handling Inconsistent Item Codes & Descriptions

A significant issue identified was the inconsistency between item codes and item descriptions. Ideally, every item should have a unique identifier and description. However, the dataset revealed multiple occurrences of :

- One Item Code with Multiple Descriptions: The same item code was mapped to different descriptions.
- One Item Description with Multiple Item Codes: Identical item descriptions were assigned different item codes.

The table below illustrates the issue:

Table 2. Inconsistent Item Codes and Description Mapping

Item Code	Item Description	Issue Type
1001	SAM SMITH ORGANIC PEAR CIDER - 18.7OZ	One Item Code with Multiple Descriptions
1001	S SMITH ORGANIC PEAR CIDER - 18.7OZ	One Item Code with Multiple Descriptions
201543	A RAFANELLI DRY CREEK VALLEY CABERNET - 750ML	One Item Description with Multiple Codes
214452	A RAFANELLI DRY CREEK VALLEY CABERNET - 750ML	One Item Description with Multiple Codes
289817	A RAFANELLI DRY CREEK VALLEY CABERNET - 750ML	One Item Description with Multiple Codes

To address this inconsistency, the *FuzzyWuzzy* Python library was used to detect and consolidate similar item descriptions based on string similarity. FuzzyWuzzy computes similarity scores using the *Levenshtein Distance*, which measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into another [33]. A similarity threshold of 95% was selected to ensure that only highly similar entries were matched, minimizing the risk of incorrect merges. The complete fuzzy matching logic are provided in Appendix A.1.

Following the cleaning process, the dataset was reduced to 33,754 unique item entries, ensuring a one-to-one mapping between item codes and descriptions and eliminating redundant or conflicting entries.

3.2.3 Handling Negative Sales Value

Another data anomaly was the presence of negative values in the sales data. Negative values were detected in 3 columns: Retail Sales, Retail Transfers, and Warehouse Sales. This can also be observed from Figure 5, as part of the summary statistics. Negative sales values are typically implausible in a retail context and, if left unaddressed, can significantly distort downstream metrics such as Inventory Turnover Ratio (ITR) and Days of Inventory on Hand (DOH), ultimately leading to misleading insights.

After the negative values were identified through the summary statistics generated (see Figure 5), conditional filtering techniques were performed to isolate all the sales entries with values less than zero. These entries were flagged for further review.

A domain-based logical assessment was conducted to determine the most appropriate handling strategy. Based on standard industry practices and the contextual understanding of retail operations, negative sales were interpreted as either data entry errors or system processing issues, rather than legitimate reverse transactions or returns. As such, all negative values were replaced with zeroes to preserve dataset consistency while minimizing the introduction of artificial trends.

In addition to addressing negative values, an outlier detection step was applied to identify abnormally high sales values that could bias category-level forecasts or distort visualizations. The Z-Score Method was used to measure the extremity of each sales value with respect to the overall distribution [34]. The full Z-score formula and justification are provided in Appendix A.2.

Following these corrections, the dataset was re-evaluated using descriptive statistics and visualizations to confirm the removal of negative values and the normalization of sales distributions. This ensured the integrity of subsequent KPIs and forecasting models.

3.3 Feature Engineering

Feature engineering involves creating meaningful variables from raw data to enhance the effectiveness of downstream analysis, visualisation, and modelling. Feature engineering techniques can be as simple as converting data types and more complicated such as the creation of interaction variables [35]. Key engineered datasets in this project include *price_info, inventory_info, orders_info* and *leadtime_info.*

3.3.1 Feature Engineering for Price Information (*price_info*)

The original dataset did not contain explicit information on item prices, which posed a challenge for calculating key supply chain metrics that depend on unit-level sales data, such as inventory turnover and unit-level demand forecasting. To address this, a synthetic price assignment process was implemented based on domain knowledge and market-informed category estimates. The resulting dataset includes average category cost, unit price, and raw simulated price for each SKU. Detailed price simulation logic and steps are provided in Appendix B.1.

The final *price_info* dataset preview is illustrated by the figure below:

	SKU	Category	Average Category Cost	Unit Price	Raw Price (Simulated)
0	10346	BEER	8	8.0	7.937270
1	100009	WINE	15	20.0	20.225357
2	100012	WINE	15	18.0	18.115997
3	100023	WINE	15	16.0	16.049329
4	100024	WINE	15	10.0	9.828009

Figure 6. Preview of price_info Dataset

This dataset was later joined with the main dataset to compute unit-level metrics and enable enhanced visualizations within Tableau. The feature engineering process ensured consistency, realism, and flexibility in unit price simulation across thousands of SKUs.

However, there are some assumptions and limitations to this approach. It was assumed that sales figures were in thousands due to their scale and expected unit price range. This assumption may introduce proportional scaling bias if the original figures reflected a different unit. However, the approach provides a reasonable approximation for prototyping and visualization purposes.

3.3.2 Feature Engineering for Inventory Estimation (*inventory_info*)

The original dataset lacked inventory data, which is essential for analyzing stock availability, detecting stockout conditions, and computing inventory management metrics such as stockout rate and inventory turnover ratio. Therefore, a synthetic inventory estimation method was developed to simulate beginning inventory, ending inventory, and stockout conditions for each stock keeping unit (SKU) over time. Simulation details and logic validation are provided in Appendix B.2.

The final *inventory_info* dataset preview is illustrated by the figure below:

	SKU	Category	Month	Monthly Sales	Beginning Inventory	Ending Inventory	Stockout Flag	Average Monthly Inventory
0	100007	WINE	1	0.0	NaN	NaN	False	NaN
1	100007	WINE	2	0.0	0.00	0.00	True	0.00
2	100007	WINE	3	0.0	0.00	0.00	True	0.00
3	100007	WINE	4	1000.0	1111.44	111.44	False	611.44
4	100007	WINE	5	1000.0	1272.90	272.90	False	772.90

Figure 7. Preview of inventory_info Dataset

Some assumptions that we have made are that this simulation assumes consistent replenishment logic and does not account for dynamic reorder policies, lead times, or multi-echelon inventory systems. The stockout probability (30%) is static and applied uniformly, which may not reflect actual supply constraints or seasonality effects in real operations.

3.3.3 Feature Engineering for Lead Time Data (*leadtime_info*)

Lead time refers to the time elapsed between order placement and delivery and is a key metric in evaluating supplier performance, particularly in retail supply chains where delayed deliveries can result in stockouts and lost sales. However, the original dataset did not contain lead time data. To facilitate on-time delivery analysis and simulate order fulfilment behaviour, a synthetic lead time dataset was generated based on category-level expectations. Lead time generation methodology is described in Appendix B.3.

The final l*eadtime_info* dataset preview is illustrated by the figure below:

	SKU	Category	Lead Time (days)
0	10346	BEER	12
1	100009	WINE	17
2	100012	WINE	14
3	100023	WINE	14
4	100024	WINE	21

Figure 8. Preview of leadtime_info Dataset

Some assumptions made in this section was that the assigned lead time ranges are based on typical industry benchmarks and generalized supply chain behaviour. They do not reflect vendor-specific agreements, geographic factors, or dynamic conditions such as peak season or transportation disruptions. Nonetheless, the generated values serve as a reasonable proxy for modelling supplier responsiveness and delivery reliability.

3.3.4 Feature Engineering for Supplier Orders Data (*orders_info*)

To simulate a realistic supplier order workflow in the absence of transaction-level procurement data, a synthetic dataset titled *orders_info* was generated with approximately 170,000 records. Each record represents an individual supplier order with attributes supporting supplier performance evaluation. Full generation logic, distribution parameters, and field derivations are included in Appendix B.4.

The final *orders_info* dataset preview is illustrated by the figure below:

	SKU	Category	Supplier	Order Placed Date	Availability	Expected Delivery Date	Order Quantity	Actual Delivery Date	Quantity Received	Quantity Accuracy	Delivery Status	Defect Flag
0	101532	WINE	THE SPANISH WINE IMPORTERS LLC	2024-03-21 02:25:33.755627	True	2024-04-08 02:25:33.755627	935	2024-04-09 02:25:33.755627	932	0	late	0
1	43233	WINE	WILSON DANIELS LTD	2024-10-28 02:25:33.755627	True	2024-11-17 02:25:33.755627	65	2024-11-15 02:25:33.755627	65	1	early	0
2	68271	BEER	VIN DIVINO LTD	2024-04-25 02:25:33.755627	True	2024-05-02 02:25:33.755627	431	2024-05-07 02:25:33.755627	431	1	late	0
3	42024	LIQUOR	BOSTON BEER CORPORATION	2024-09-20 02:25:33.755627	True	2024-10-06 02:25:33.755627	16	2024-10-08 02:25:33.755627	16	1	late	0
4	82403	WINE	CONSTANTINE WINES INC	2024-05-17 02:25:33.755627	True	2024-06-07 02:25:33.755627	981	2024-06-06 02:25:33.755627	978	0	early	0

Figure 9. Preview of orders_info Dataset

The generation of the *orders_info* dataset relied on a series of simplifying assumptions aimed at balancing realism with computational efficiency. All supplier orders were simulated as independent events, with no modelling of batch ordering behaviour, minimum order quantities, or supplier-specific contract terms. In practice, suppliers often operate under predefined order cycles or quantity thresholds, which may affect delivery patterns and inventory replenishment logic.

3.4 Forecasting Model Selection and Evaluation Process

Demand forecasting is a critical component in supply chain planning. Accurate forecasts enable better inventory control, supplier coordination, and improved service levels. Demand forecasting encompasses a diverse array of methodologies, each characterized by distinct assumptions, computational complexities, and predictive capabilities [43]. In this project, we implemented and evaluated multiple time series forecasting models to predict future demand based on historical sales data. This section outlines the steps taken to generate a historically complete dataset for training and testing forecasting models, as well as the rationale behind model selection.

3.4.1 Generating Extended Dataset

To enable time series forecasting with proper consideration for seasonality and historical patterns, it was necessary to extend the dataset beyond the single available year of 2024. Since the original dataset lacked historical sales records, synthetic data was generated for the years 2022 and 2023 using main.csv as a foundational reference. The objective was to produce plausible monthly sales values for each SKU while embedding seasonal fluctuations and demand irregularities.

Using *main.csv* as a foundational reference, the generation process applied category-informed seasonal multipliers, controlled randomness, and probability-based logic to determine whether a SKU-month entry should be created. Baseline sales values were adjusted based on expected seasonal behaviour and randomized scaling to reflect market variability while maintaining coherence with 2024 patterns. The final extended dataset comprises two data frames, one for 2022 and one for 2023, which were concatenated with the cleaned 2024 data to form a unified historical dataset, *main_extended.csv*. This file served as the basis for all time-series forecasts conducted in Python and Tableau. Full details of the generation logic and steps taken are provided in Appendix B.5.

	Month	SKU	Description	Category	Supplier	Sales
0	2022-01-01	102	TWO BOTTLE WINE TOTE	STR_SUPPLIES	Default	480.0
1	2022-01-01	104	FOUR BOTTLE WINE TOTE	STR_SUPPLIES	Default	8420.0
2	2022-01-01	105	WINE GIFT TOTE SINGLE BOTTLE	STR_SUPPLIES	Default	45372.0
3	2022-01-01	106	SIX BOTTLE WINE TOTE (NO LOGO)	STR_SUPPLIES	Default	3505.0
4	2022-01-01	109	WINE TOTE WOOD HANDLE 2 POCKET 4 BTL	STR_SUPPLIES	Default	588.0

Figure 10. Preview of main_extended Dataset

3.4.2 Selected Models for Testing and Comparison

To determine the most suitable forecasting model for demand prediction, two well-established univariate time series models were selected: Facebook Prophet and Exponential Smoothing (ETS). These models were chosen for their widespread adoption in industry and academic research, their ability to model seasonality and trend, and their ease of integration within Python-based forecasting pipelines. The following table compares the two forecasting models:

Table 3. Table Comparison of Prophet and ETS

Criteria	Prophet	Exponential Smoothing (ETS)
Developed by	Facebook (Meta)	Statistical community / Forecasting literature
Model Type	Additive time series model	Statistical exponential smoothing model
Handles Trend	Yes (linear/non-linear changepoints allowed)	Yes (additive/multiplicative)
Handles Seasonality	Yes (automatically detects & models it)	Yes (explicit seasonal components)
Handles Holidays/Events	Yes (user-defined custom events)	No
Robust to Outliers & Missing	Yes	Less robust (sensitive to anomalies)
Ease of Use	High (API is simple to use)	Moderate

Forecast Accuracy (in project)	Better (MAPE = 7.10%, SMAPE = 7.50%)	Moderate (MAPE = 8.96%, SMAPE = 9.49%)
Interpretability	Moderate (black-box components)	High (transparent smoothing components)
Suitability for Business Use	Very suitable for business-level applications	Suitable for regular, stable patterns

The following steps were taken to develop the two models:

Data Aggregation: Monthly sales were aggregated per SKU to create a univariate time series suitable for forecasting. This step reduced noise and enabled clearer seasonal and trend component extraction.

Train-Test Split: The combined data from 2022 and 2023 served as the training period, while the actual sales for 2024 were withheld and used as ground truth for out-of-sample validation.

Forecast Generation: Both Prophet and ETS models were independently trained on the aggregated monthly sales from 2022–2023. Each model was used to forecast monthly sales for the year 2024. The forecasts were then compared to the actual 2024 values to assess predictive accuracy. Evaluation metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Symmetric MAPE (SMAPE) were used, as described in Section 3.4.3.

A visualization showing the comparison of actual vs forecasted sales by both models is shown below:

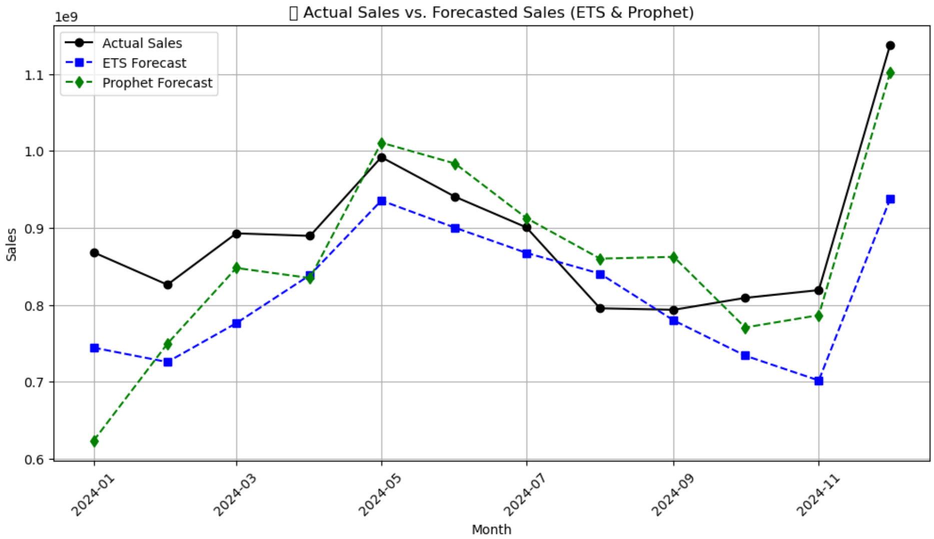


Figure 11. Actual Sales vs. Forecasted Sales (ETS & Prophet)

3.4.3 Forecast Accuracy Metrics

To assess the performance of the forecasting models and guide model selection, three standard evaluation metrics were used: Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE). These metrics collectively evaluate both the magnitude and directional bias of forecast errors.

Mean Absolute Percentage Error (MAPE)

MAPE quantifies the average percentage deviation between forecasted and actual values. It is widely used for intuitive interpretation and unit-free nature. The formula is:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{17}$$

Where $A_t$is the actual value and F_t is the forecasted value at time t. Lower values indicate higher forecasting accuracy.

Root Mean Squared Error (RMSE)

RMSE measures the standard deviation of residuals (prediction errors), reflecting how concentrated the data is around the line of best fit. It penalizes large errors more heavily, making it sensitive to outliers:

$$MAPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_t - F_t)^2} \tag{18}$$

This metric is useful when the cost of large deviations is high.

Symmetric Mean Absolute Percentage Error (SMAPE)

SMAPE is a variant of MAPE that scales the absolute error by the average of actual and forecasted values, reducing bias when actual values approach zero:

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \tag{19}$$

It provides a more balanced view of over- and under-estimation performance

Table 4. Forecast Accuracy Metrics Prophet vs. ETS

Metric	Prophet	ETS
MAPE	7.10%	8.96%
RMSE	84.6M	95.5M
SMAPE	7.50%	9.49%

3.5 Tools, Technologies, Libraries Used

This project utilized a robust and well-integrated technology stack, combining Python-based data processing with advanced visualization and forecasting tools. The entire data pipeline - from raw ingestion and cleaning to synthetic data generation, forecasting, and dashboard development - was implemented using the tools outlined below. The selection of technologies was driven by the need for reproducibility, ease of integration, and support for time series analysis and dashboarding.

Programming Language and Development Environment

Python served as the primary programming language due to its versatility and extensive ecosystem of data science libraries. All code was developed and tested in Jupyter Notebook, which enabled modular, interactive scripting and iterative experimentation. The notebook environment also facilitated transparency and version control during data transformation and model evaluation.

Data Processing and Feature Engineering

Data loading, manipulation, and feature engineering were primarily carried out using *Pandas*, which offers a flexible DataFrame structure ideal for structured data operations. *NumPy* was used for numerical operations, including array-based sampling and synthetic data simulation. *Datetime* functions were extensively applied to generate order dates, lead times, and monthly sales aggregations. Together, these tools enabled end-to-end transformation of the dataset into multiple analysis-ready formats.

Data Cleaning and Enhancement

To resolve inconsistencies in item codes and descriptions, *FuzzyWuzzy* was employed for fuzzy string matching using *Levenshtein* distance. This facilitated the consolidation of near-duplicate entries. *SciPy* was used to identify and handle outliers using the Z-score method, which was essential for ensuring accurate forecasting and KPI computation.

Exploratory Analysis and Visualization

Initial data exploration and feature distribution analysis were conducted using *Matplotlib* and *Seaborn*, which supported the generation of heatmaps, boxplots, KDE plots, and comparative visualizations. *Statsmodels* was used to generate Autocorrelation Function (ACF) plots, helping to identify trend and seasonality components in sales data before model selection.

Forecasting Models and Evaluation

Two forecasting models were selected for comparison: *Facebook Prophet*, which offers additive modeling of trends and seasonality with strong handling of irregularities, and ETS (Exponential Smoothing), implemented using the Exponential Smoothing class from *Statsmodels*. These models were trained on SKU-level monthly sales data from 2022–2023 and used to forecast demand for 2024–2025. Forecast accuracy was assessed using Scikit-learn, which was used to compute performance metrics such as MAPE, RMSE and SMAPE.

Dashboarding and Business Intelligence

The final visual interface was developed in *Tableau*, chosen for its powerful interactive visualization capabilities. Tableau was used to create three interconnected dashboards covering inventory management, supplier performance, and demand forecasting. Calculated fields and parameters were configured to allow dynamic filtering, drill-downs, and scenario-based analysis.

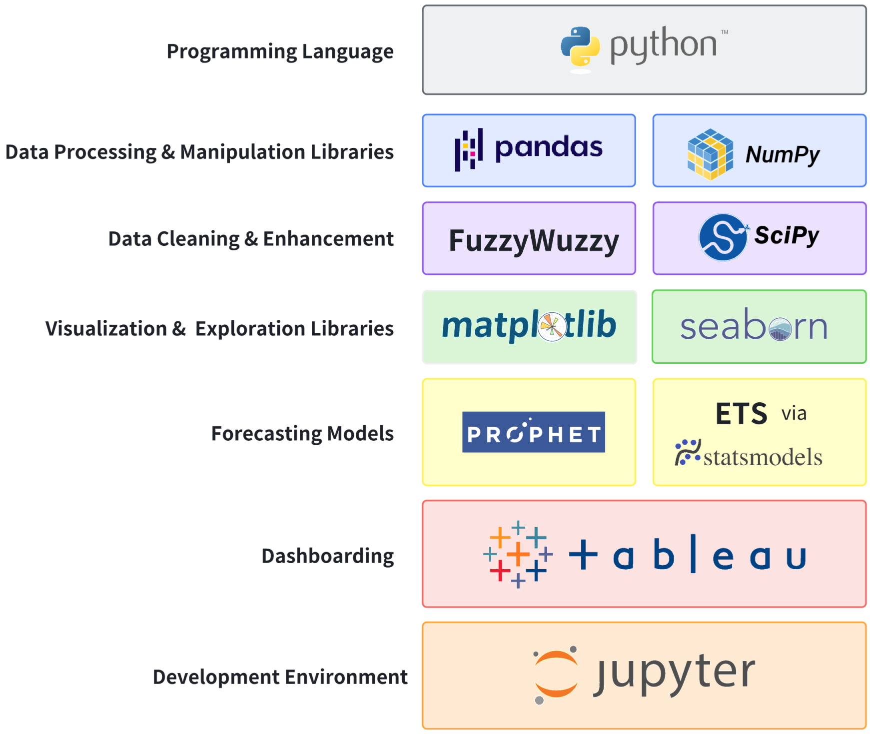


Figure 12. Technology Stack Diagram

Chapter 4: Dashboard Design & Implementation in Tableau

4.1 Dashboard Overview

Three core dashboards were developed in Tableau to address key functions of the supply chain: inventory management, supplier performance, and demand forecasting. Each dashboard was designed to be modular, user-friendly, and tailored to the specific needs of supply chain analysts and decision-makers. The dashboards draw from the cleaned and engineered datasets outlined in Chapter 3, and each leverages interactive components to facilitate real-time insights, drill-down analysis, and proactive decision-making. The dashboards were built with a consistent design language and a focus on clarity, responsiveness, and performance optimization across a range of use cases.

4.2 Inventory Management Dashboard

The Inventory Management Dashboard provides a high-level overview of stock health and inventory movement across the organization. It is designed to help supply chain managers track key inventory metrics and quickly identify issues such as overstocking, low turnover, and frequent stockouts.

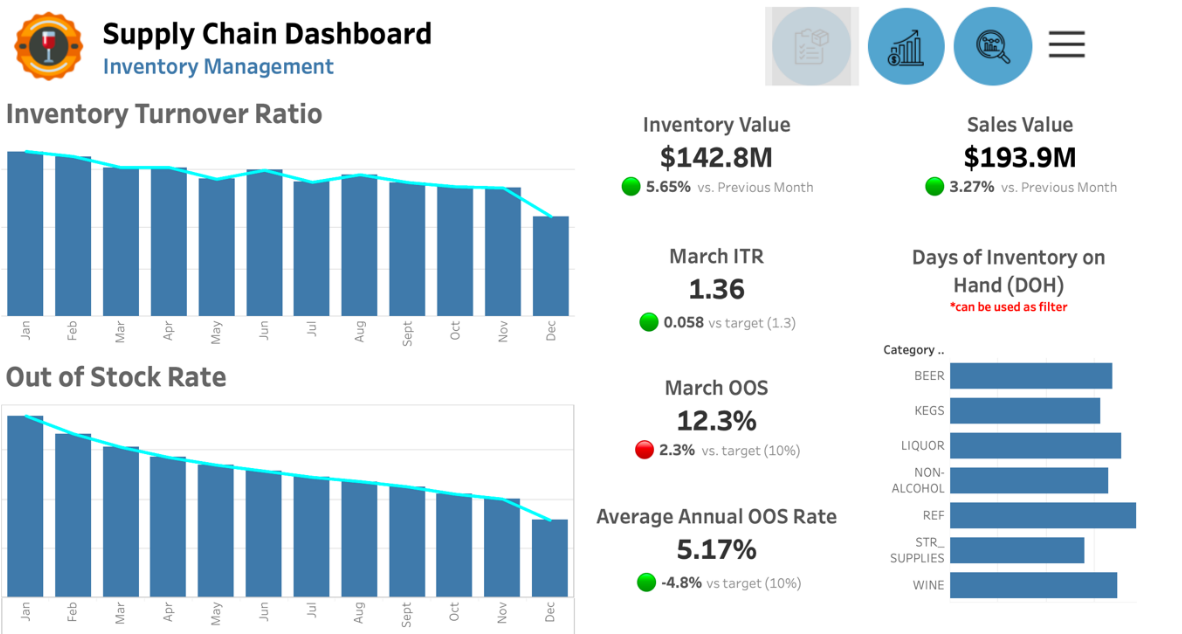


Figure 13. Inventory Management Dashboard

4.2.1 Key Metrics

This dashboard features the following key performance indicators (KPIs):

Inventory Turnover Ratio (ITR)

Displayed using a monthly bar chart with an overlaid trend line to reveal changes in inventory movement over time. This format emphasizes fluctuations across the year and allows users to quickly identify months with high or low turnover.

Out of Stock (OOS) Rate

Shown as a time-series bar chart, the OOS rate illustrates the proportion of SKUs that experienced stockouts each month. The bar chart format facilitates clear comparison across months and highlights improvements or deteriorations in stock availability.

Days of Inventory on Hand (DOH)

Represented using a horizontal bar chart categorized by product type. This layout was chosen to make category-level comparisons intuitive and scannable. The chart also serves as an interactive filter, allowing users to isolate specific categories and dynamically adjust the entire dashboard view.

Sales and Inventory Value

Positioned prominently as KPI summary cards, these values provide a quick snapshot of financial performance. They include delta indicators to show percentage changes from previous months, with color-coded signals (i.e. green for improvement, red for decline) to aid rapid interpretation.

Additional views of monthly ITR and OOS are positioned as summary cards and benchmarked against internal targets (e.g., an ITR target of 1.3 and OOS threshold of 10%). These metrics are accompanied by visual indicators that highlight whether performance exceeds or falls short of expectations.

While elements such as dynamic filters, tooltips, and conditional formatting enhance the dashboard's interactivity, these features are discussed in greater detail in Section 4.3, which outlines the overall interactivity framework and shared design logic across all dashboards.

The Inventory Management Dashboard ultimately empowers users with actionable insights into stock movement and availability, contributing to a more responsive and efficient supply chain operation.

4.2.2 Use Case Scenario

To demonstrate the practical application of the Inventory Management Dashboard, this section presents how different supply chain roles can leverage the dashboard for operational decision-making. The dashboard supports a variety of tasks, including stock monitoring, replenishment planning, and identifying inefficiencies. Table 4. summarizes potential use cases for key roles, followed by a detailed scenario that illustrates a real-world application in a distribution center setting.

Table 5. Roles and Use Cases for the Inventory Management Dashboard

Role	Use Case
Inventory Manager	Monitor DOH, detect low stock SKUs, initiate replenishment.
Warehouse Supervisor	Track high-turnover items and coordinate shelf space/logistics.
Procurement Planner	Identify overstocked SKUs and pause/schedule future orders.
Supply Chain Analyst	Analyze trends in stockout rates and inventory turnover for reporting.
Category Manager	Review inventory levels by category and adjust safety stock thresholds.

To provide further clarity, the following use case example focuses on the role of an inventory manager overseeing a regional distribution centre. This example outlines how the dashboard's interactivity and real-time metrics enable operational decisions that enhance inventory performance.

User Role: Inventory Manager at a Regional Distribution Centre

Objective**:** Minimize stockouts while optimizing stock holding through monthly performance reviews.

Scenario Overview*:* At the beginning of each month, the inventory manager conducts a monthly inventory performance review using the dashboard. Upon selecting March from the interactive filter, the Out-of-Stock (OOS) Rate displays a value exceeding the internal threshold of 10% and highlighted in red. This alerts the manager to potential product availability risks. The Inventory Turnover Ratio (ITR) for March is 1.36, slightly above the target of 1.3, indicating moderate stock movement, but uneven availability across categories remains a concern.

To localize the issue, the manager refers to the DOH by Category bar chart. Categories such as "Kegs" and "Wine" have notably lower DOH, signalling fast depletion, whereas "STR_SUPPLIES" shows high DOH, indicating excess stock.

Action Taken: Using the dashboard's interactive filter, the manager selects "Kegs" to isolate SKU-level performance for this category. Upon review, several SKUs are approaching stockout, likely due to increased demand linked to recent events. The manager initiates a priority replenishment order and updates the safety stock configuration for Kegs moving forward. Simultaneously, the overstock condition for "STR_SUPPLIES" prompts an internal review, leading to a temporary halt in procurement and a proposal for bundled promotions.

Additionally, the Inventory Value metric indicates a 5.65% increase from the previous month, suggesting rising inventory costs. This further justifies corrective actions for overstocked categories.

Benefits & Outcome:

- Stockouts were prevented through early detection using monthly performance metrics.
- Excess inventory was flagged and addressed through adjusted procurement strategies.
- Data from the dashboard enabled proactive, role-specific decision-making aligned with operational goals.

4.3 Supplier Performance Dashboard

The Supplier Performance Dashboard was developed to monitor, benchmark, and compare supplier reliability based on multiple service quality metrics. It supports procurement teams in identifying top-performing suppliers, flagging underperformers, and analyzing category-specific lead time patterns that may impact delivery consistency.

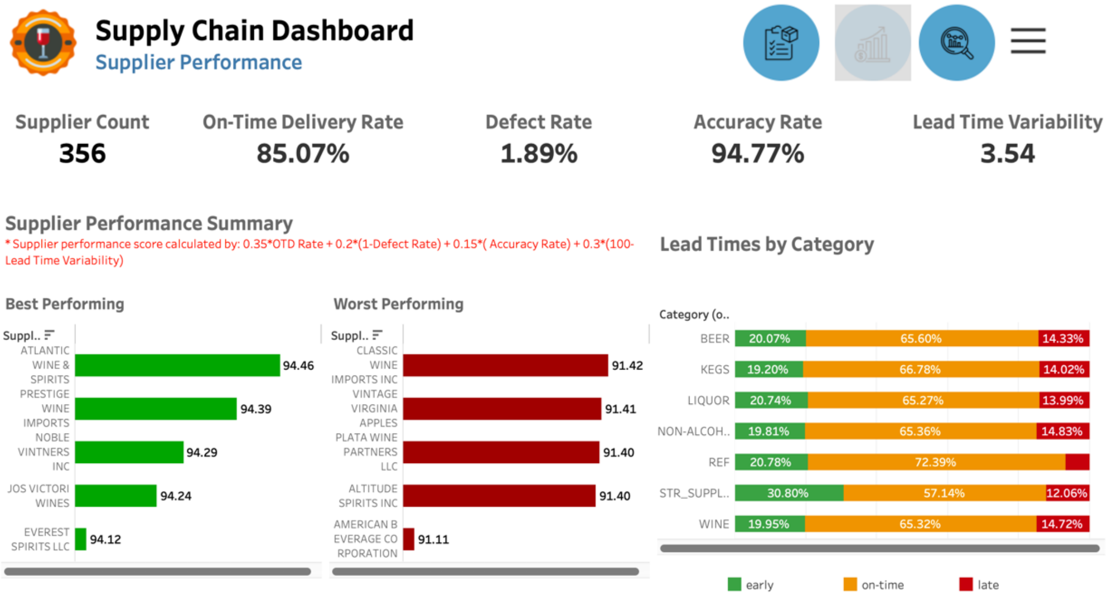


Figure 14. Supplier Performance Dashboard

4.3.1 Key Metrics

This dashboard features the following key metrics:

Supplier Count

A high-level count of unique suppliers in the dataset, presented as a summary card to provide immediate scope context.

On-Time Delivery (OTD) Rate

Visualized as a headline KPI card, this metric measures the percentage of orders delivered on or before the expected delivery date. The clean, prominent display emphasizes its critical role in supplier evaluation.

Defect Rate

Represented using a horizontal bar chart categorized by product type. This layout was chosen to make category-level comparisons intuitive and scannable. The chart also serves as an interactive filter, allowing users to isolate specific categories and dynamically adjust the entire dashboard view.

Accuracy Rate

Displayed as part of the supplier KPI summary cards, the accuracy rate reflects the proportion of orders where the quantity delivered matched the quantity ordered. It is calculated as the percentage of accurate orders over total orders and provides a key indicator of fulfillment precision.

Lead Time Variability

This metric quantifies the consistency of a supplier's delivery lead time by measuring the standard deviation of actual delivery days versus expected lead times. High variability indicates unpredictable delivery performance, which increases the risk of stockouts or overstocking. The dashboard highlights suppliers with significant fluctuations, enabling procurement managers to engage suppliers for process improvement or reassess sourcing strategies.

Supplier Performance Summary (Top 5 and Bottom 5)

Two horizontal bar charts display the five best-performing and five worst-performing suppliers, based on a weighted composite score. The visualization uses color-coded bars (green for best, red for worst) to enhance clarity and draw attention to extremes.

This composite index captures both service quality and consistency dimensions, allowing a more nuanced comparison beyond individual metrics.

Lead Times by Category

A stacked bar chart shows the proportion of early, on-time, and late deliveries per category. This visualization aids in identifying which product categories are more prone to lead time issues, enabling category-specific supplier engagement strategies. The use of green, orange, and red segments improves interpretability and facilitates root cause analysis.

4.3.2 Use Case Scenario

The Supplier Performance Dashboard enables procurement teams and vendor managers to evaluate supplier reliability and quality across key performance indicators. By visualizing delivery accuracy, defect rates, and lead time variability, this dashboard supports strategic sourcing decisions and ensures that supplier-related risks are detected early. The table below outlines how different stakeholders can potentially utilize the dashboard in monthly supplier performance reviews.

Table 6. Roles and Use Cases for the Supplier Performance Dashboard

Role	Use Case
Procurement Analyst	Track supplier OTD rates, flag underperformers for follow-up.
Vendor Manager	Evaluate quantity accuracy and defect trends to initiate supplier audits
Operations Manager	Identify overstocked SKUs and pause/schedule future orders.
Supply Chain Risk Analyst	Analyze trends in stockout rates and inventory turnover for reporting.

To provide a more detailed illustration, the following scenario focuses on a procurement analyst responsible for tracking monthly supplier performance and driving improvements in order fulfilment efficiency.

User Role: Procurement Analyst in the Central Purchasing Division

Objective: Identify suppliers with declining delivery performance and initiate corrective actions.

Scenario Overview: During the monthly supplier performance review for March, the procurement analyst uses the dashboard to filter by category. The On-Time Delivery (OTD) Rate for Supplier B drops significantly below the internal benchmark of 90%, and is visually flagged in red. At the same time, Lead Time Variability for the same supplier is elevated, suggesting unstable delivery scheduling. The dashboard also shows a spike in the Defect Rate, adding further concern regarding quality assurance.

Action Taken: The analyst compiles the performance data and schedules a review meeting with the supplier's account manager. During the meeting, they discuss potential causes, ranging from upstream disruptions to process inefficiencies, and set a corrective action plan with monthly performance targets. Internally, the analyst recommends shifting partial volume to Supplier C, whose performance metrics have been consistently within target thresholds. These changes are documented in the supplier scorecard system, and the dashboard is updated with notes to monitor changes in OTD and defect rate trends in subsequent months.

Benefits & Outcome:

- Underperforming supplier was flagged using monthly KPI thresholds.
- Risk was mitigated by redistributing orders to higher-performing suppliers.
- The dashboard served as a central tool for supplier negotiation and long-term performance tracking.

4.4 Demand Forecasting Dashboard

The Demand Forecasting Dashboard is designed to assist supply chain analysts and planners in evaluating the accuracy of demand forecasts and identifying potential stock imbalances in advance. By combining historical sales data with machine learning-based predictions, this dashboard provides forward-looking insights that support data-driven inventory and procurement decisions.

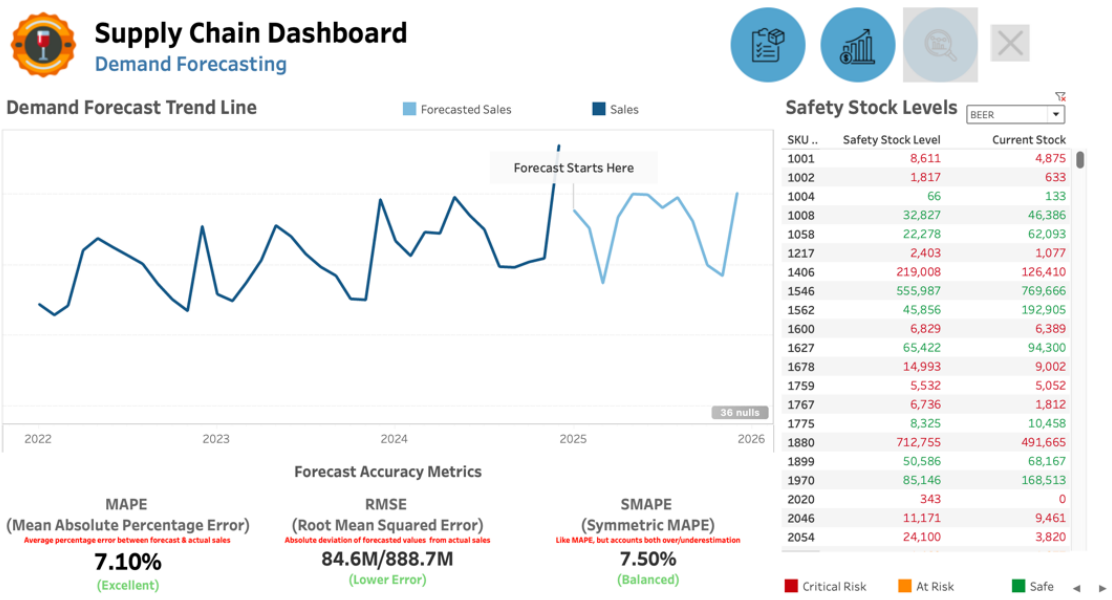


Figure 15. Demand Forecasting Dashboard

4.4.1 Key Metrics

The dashboard features the following key metrics:

Demand Forecast Trend Line

A line chart comparing actual sales and forecasted sales over time, enabling users to visually assess the accuracy of forecasts and detect demand surges or drops. The forecast start date is clearly annotated to distinguish historical data from projected values.

Forecast Accuracy Metrics

Three industry-standard metrics are displayed to evaluate forecast model performance:

- MAPE: Measures average absolute deviation as a percentage of actual values.
- RMSE**:** Shows the square root of average squared differences between predicted and actual values.
- SMAPE: A variation of MAPE that balances over- and under-forecasting penalties.

Safety Stock Levels Tables

A sortable table showing SKU-level safety stock thresholds compared against current stock levels. SKUs are conditionally formatted based on risk status:

- Critical Risk (red): Current stock falls significantly below safety stock.
- At Risk **(**amber): Marginal shortfalls relative to buffer levels.
- Safe **(**green): Sufficient inventory relative to forecasted demand.

This dashboard enables planners to validate forecasting models, identify SKUs with high volatility, and flag items requiring immediate restocking or supply adjustments. Its visual clarity and color-coded risk alerts help prioritize actions, especially in fast-moving or seasonal product categories.

4.4.2 Use Case Scenario

The Demand Forecasting Dashboard empowers supply chain planners and analysts to assess the accuracy of demand predictions and make informed decisions about production, procurement, and inventory planning. By comparing historical sales with forecasted values and evaluating performance metrics such as MAPE and SMAPE, the dashboard provides actionable insights into how well demand is being predicted and where adjustments may be needed. The following table summarizes typical use cases by key roles.

Table 7. Roles and Use Cases for Demand Forecasting Dashboard

Role	Use Case
Demand Planner	Monitor forecast accuracy and adjust forecasting models or parameters.
Supply Chain Analyst	Identify high-variance SKUs and validate forecasts before execution.
Procurement Manager	Align order quantities with projected demand to avoid over/under-buying.
Production Scheduler	Coordinate manufacturing plans with forecasted demand trends.

To give a more detailed illustration, the following scenario focuses on a demand planner in a fast-moving consumer goods (FMCG) company who is responsible for reviewing forecasting performance and refining upcoming procurement quantities based on data insights from the dashboard.

User Role: Demand Planner for the FMCG Division

Objective: Evaluate forecast accuracy, identify at-risk SKUs, and adjust future procurement decisions based on performance insights.

Scenario Overview: At the start of April, the demand planner launches the dashboard to review March's forecast performance. The forecast trend line chart shows how actual sales compared to predicted values across multiple months, with the forecast start point clearly annotated. The Mean Absolute Percentage Error (MAPE) for March, and the Symmetric MAPE (SMAPE) is both within the acceptable range. The RMSE, displayed in both unit and monetary terms confirms a reasonably low deviation between predicted and actual sales.

To further investigate risk areas, the planner uses the category filter to narrow the view to "BEER". The Safety Stock Levels table immediately highlights several SKUs in red (critical risk) and orange (at risk), indicating that current stock is falling below safety thresholds. For example:

- SKU #1001 has a safety stock of 8,611 units but only 4,875 in current stock
- SKU #2046 is also at risk, with 9,461 units in stock against a safety level of 11,171

These insights suggest that although the overall forecast error is acceptable, specific SKUs have been under-forecasted or replenishment has lagged behind demand.

Action Taken: The planner flags these SKUs for immediate procurement review and checks historical sales data to confirm that recent promotions or external events may have triggered unexpected spikes. After discussion with the marketing and category teams, the planner manually adjusts next month's forecast for affected SKUs and initiates safety stock recalibration to reduce future stockouts.

Additionally, the planner documents this as a forecast feedback loop for model training – fine - tuning Prophet model seasonality settings to better capture event-based demand spikes in future forecasts.

Benefits & Outcome:

- SKUs at risk of stockout were identified early using safety stock comparisons
- Forecast accuracy was refined through manual override and parameter tuning
- Future inventory planning is improved by integrating promotional signals and model feedback.

4.5 Data Source and Integration in Tableau

All three dashboards were developed in Tableau using a series of pre-processed and feature-engineered datasets created in Python, as described in Chapter 3. These datasets were structured to support seamless integration with Tableau, ensuring optimal performance, scalability, and maintainability. To enable efficient data exploration and dashboard responsiveness, key datasets were joined and transformed during the data preparation stage, minimizing the need for complex real-time computation within Tableau.

This section outlines the data flow architecture implemented in Tableau, followed by a detailed description of the calculated fields and logic used to derive dashboard-level insights.

4.5.1 Data Flow and Integration

Figure 16. Tableau Data Source Connection

The core of the data model was based on the following structured joins:

Table 8. Tableau Data Source Relationships

Join Relationship	Key Used	Cardinality	Purpose
`main` ↔ `inventory_info`	SKU	Many-to-Many	Match sales and inventory records across SKUs and months
`main` ↔ `leadtime_info`	SKU	Many-to-One	Associate each SKU with its category-level lead time data
`leadtime_info` ↔ `orders_info`	SKU	One-to-Many	Link each product category to supplier order records
`main` ↔ `price_info`	SKU	Many-to-One	Enrich SKU sales data with estimated unit prices
`main_extended ∪ forecast_2025 (Union)`	SKU, Month	— *(Union)*	Combine historical (2022–2024) and forecasted (2025) data
`Union` ↔ `price_info`	SKU	Many-to-Many	Associated unioned sales/forecast records with pricing information

All joins were constructed intentionally based on referential logic determined during data preprocessing. By performing most transformations and data cleaning in Python (as detailed in Chapter 3), only minimal logic needed to be applied within Tableau. This reduced dashboard latency and improved responsiveness. The resulting schema allowed each dashboard to retrieve precisely the information needed for real-time analysis while maintaining model consistency across views.

4.5.2 Calculated Fields and Logic

To enable advanced interactivity and derive analytical insights from raw datasets, a series of calculated fields, logic conditions, and parameter-driven computations were implemented in Tableau. These fields played a critical role in transforming static data into responsive dashboards aligned with supply chain KPIs and decision-making needs.

Parameter-Driven Monthly Calculations

A central parameter, *Select Month*, was created to allow users to choose the month of interest interactively. This parameter dynamically filters and drives multiple calculated fields, including:

- *Current Month (CM)* and *Previous Month (PM)* values for inventory, sales, ITR, and other KPIs.
- Month-specific metrics enabled flexible month-to-month comparisons and trend analyses within a consistent dashboard view.

Inventory and Sales Metrics

Key inventory-related fields included:

- *Inventory Turnover Ratio (ITR)* and *Days of Inventory on Hand (DOH),* both computed using monthly sales and average inventory.
- *Stockout Rate* and *Total Sales Events*, which collectively capture SKU-level availability risk.
- *Percentage Change Fields* such as *% Inventory Change* and *% Sales Change* to enable performance comparisons between the selected and previous months.

Supplier Performance Metrics

Supplier reliability was assessed through a combination of:

- *OTD Rate*, *Defect Rate*, *Accuracy Rate*, and *Lead Time Variability (σ_LT)* - calculated using order delivery records and defect flags.
- A composite *Supplier Performance Score*, calculated using the following weighted formula:

$$Score = 100 \times \left(0.35a + 0.2(1 - b) + 0.15c + \frac{0.3(100 - d)}{100}\right. \quad (20)$$

where a = OTD Rate, b = Defect Rate, c= Accuracy Rate, and d = Lead Time Variability

- Supplier Ranking Logic using *RANK_UNIQUE()* to extract Top 5 and Bottom 5 suppliers, with label fields (*Supplier Label, Supplier Ranking Top/Bottom 5*) to support dynamic bar chart titles and colour formatting.

Forecasting and Inventory Risk Logic

Fields related to forecasting and safety stock included:

- Forecasted Safety Stock Level, calculated using demand variability and lead time:

$$Safety\ Stock = Z \times \sigma_D \times \sqrt{Lead\ Time} \quad (21)$$

  Where Z = 1.65 for a 95% service level.
- Stock Status, a logic-based classification indicating "Critical Risk," "At Risk," or "Safe" based on current stock compared to safety stock thresholds.
- Demand Variability (σ_D), computed using standard deviation of monthly sales.

Conditional Formatting and Color Logic

Several fields were used exclusively to control formatting and UX:

- *ITR Target Diff* and its accompanying emoji-coded result (*ITR Target Diff Color*) to signal over/under performance.
- *Color-based field outputs* for performance bands in supplier scores and stock status.
- Logical filters such *as Stockout Flag, Stockout Events*, and *OOS Difference* enabled targeted highlighting of underperforming SKUs and suppliers.

In total, these calculated fields were essential to linking raw data with key dashboard elements, enabling responsive visualization, trend tracking, and risk identification across the dashboards. Further logic related to filters, highlights, and user interactivity is described in Section 4.3.

4.6 Dashboard Interactivity and Features

The dashboards developed in this project are designed to be both visually intuitive and functionally interactive, allowing users to extract insights dynamically without requiring prior technical knowledge of data analytics tools. Interactivity features were incorporated across all three dashboards—Inventory Management, Supplier Performance, and Demand Forecasting—to enhance user experience, support real-time exploration, and enable flexible analysis of supply chain performance metrics.

4.6.1 Parameters and Filter

Interactive filters were implemented to allow users to dynamically explore performance by category , or time period. Drop-down menus, category selectors, and clickable bar charts enable fast filtering without needing to leave the dashboard context.

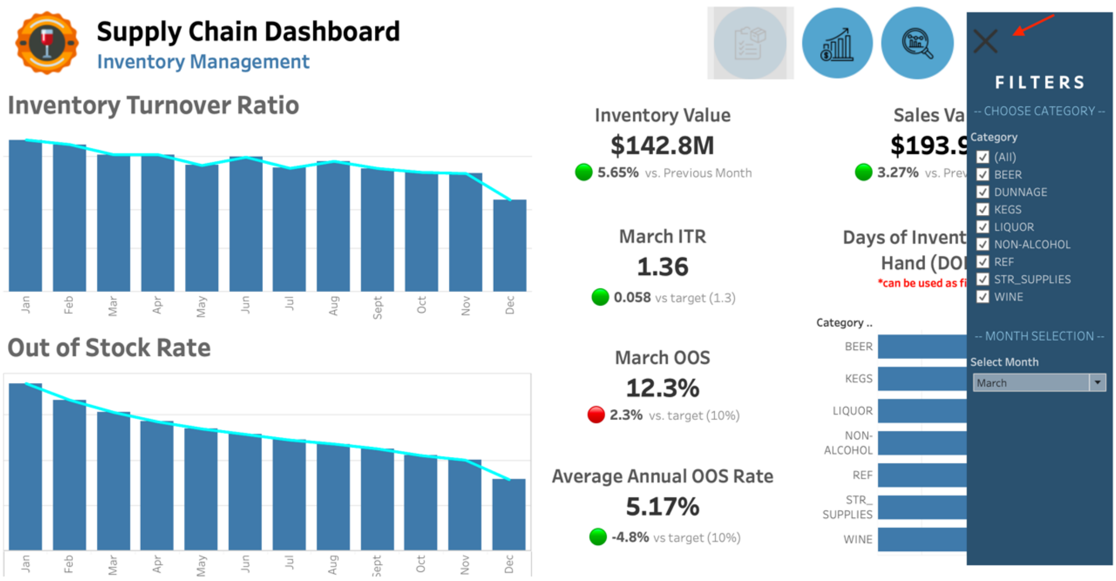


Figure 17. Interactive Filters Integrated in Dashboard

4.6.2 Icon-Based Dashboard Navigation

Each dashboard includes a row of icon-based navigation buttons positioned at the top of the interface. These icons represent the main views - Inventory, Supplier, and Forecasting - and serve as clickable shortcuts to enable smooth transitions across different analytical perspectives. This enhances the user experience by reducing friction in navigating the overall dashboard system.

Figure 18. Icons to Navigate Across Dashboards

4.6.3 Drill-Down and Cross-Filtering Features

Charts and visual components are designed to act as both data visualizations and interactive filters. For example, in the Inventory Management dashboard, clicking a category within the Days of Inventory on Hand (DOH) chart automatically filters the Inventory Turnover Ratio and Out-of-Stock Rate charts to reflect the selected category (see Figure 17). Similarly, selecting a supplier bar in the Supplier Performance dashboard filters all corresponding KPIs and trend charts (Figure 18). These drill-down and cross-filtering capabilities allow users to uncover granular insights within specific segments.

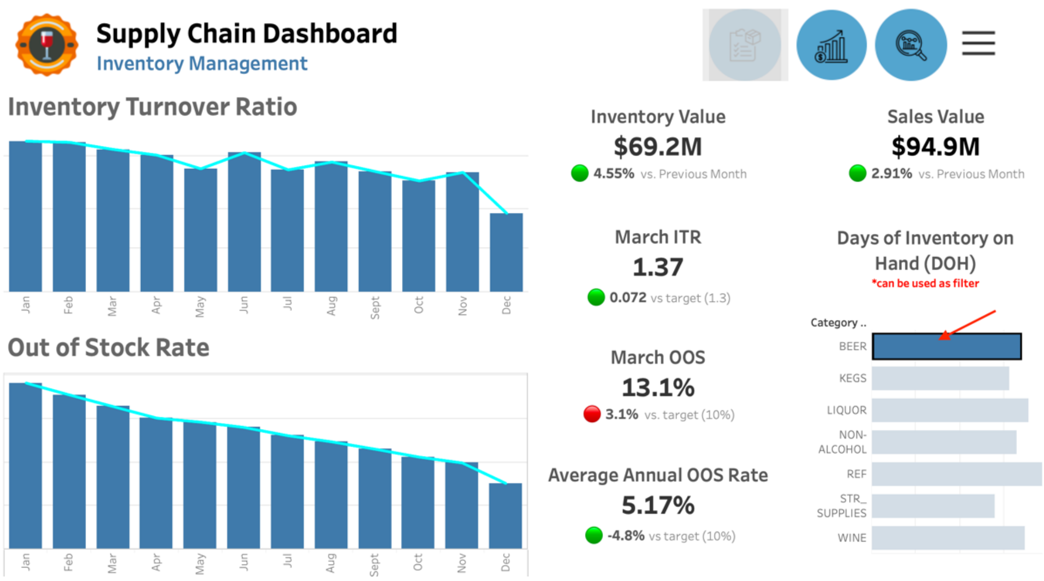


Figure 19. DOH Bar Chart as Interactive Filters

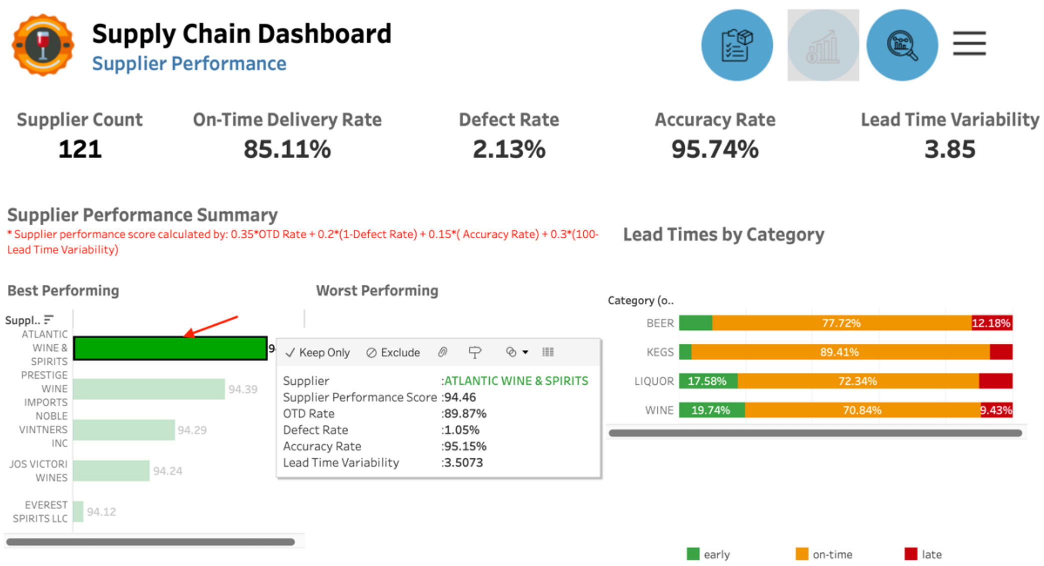


Figure 20. Supplier Rank Bar Chart as Interactive Filters

4.6.4 Conditional Formatting and Visual Cues

To support fast interpretation of KPI status, conditional logic and color cues were applied throughout:

- Stock status indicators are colour coded as:
  - 🟥 Critical Risk: Stock below safety threshold
  - 🟧 At Risk: Stock slightly above threshold
  - 🟩 Safe: Stock exceeds safety buffer
- *Supplier Performance Bars* are color-coded to distinguish top 5 (green) and bottom 5 (red) suppliers.
- *ITR Target Differences* are presented with visual emojis (🟢 / 🔴) to immediately indicate whether performance exceeds or falls short of the benchmark.

4.6.5 Tooltips and Hover Effects

Custom tooltips were implemented across all chart elements to provide context-aware data on hover. For example, hovering over supplier bars reveals key performance metrics, such as On-Time Delivery Rate, Defect Rate, and Quantity Accuracy, for that supplier (Figure 24). KPI cards also feature hover-enabled tooltips indicating month-over-month performance and alignment with target benchmarks.

These tooltips were designed to be concise, informative, and context-specific, allowing users to extract additional insights without crowding the visual space. Hover effects are further used to highlight data points and facilitate user focus. The following figures illustrates all the tooltips included in the dashboards.

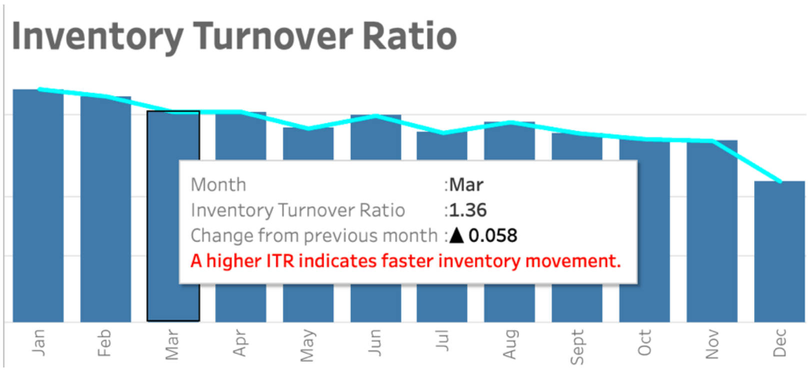


Figure 21. ITR Bar Chart Tooltip

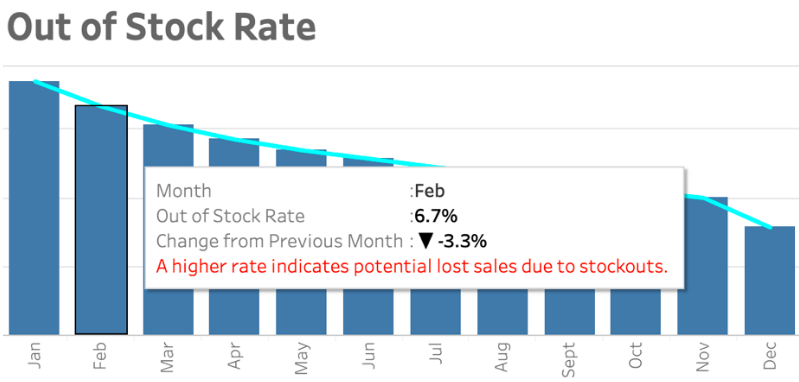


Figure 22. OOS Rate Bar Chart Tooltip

Figure 23. DOH Bar Chart Tooltip

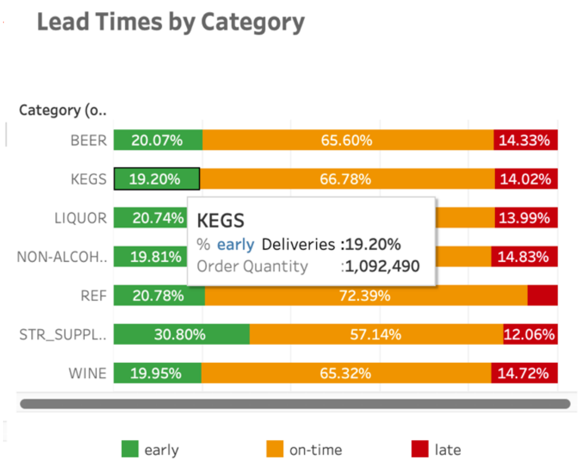


Figure 24. Lead Times Bar Chart Tooltip

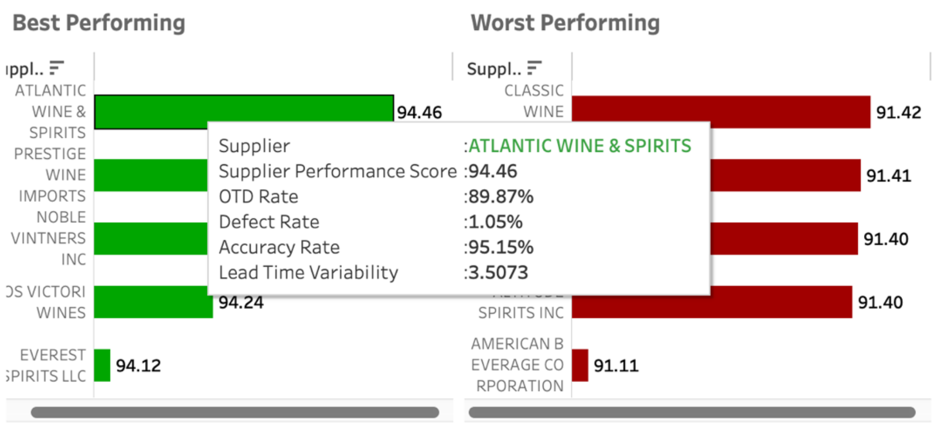


Figure 25. Supplier Performance Rank Chart Tooltip

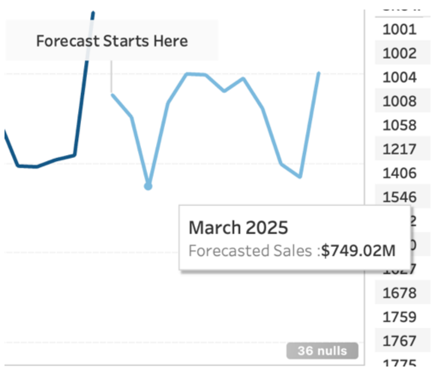


Figure 26. Forecast Line Tooltip

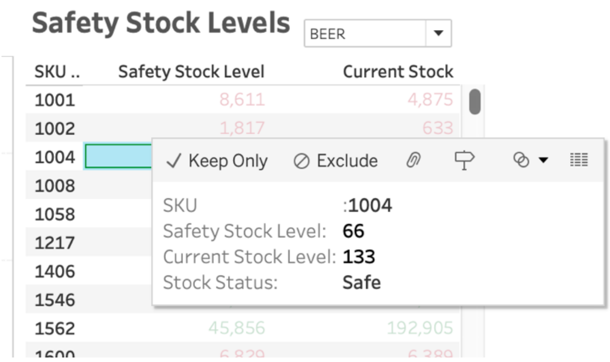


Figure 27. Safety Stock Levels Tooltip

Chapter 5: Challenges & Limitations

This chapter outlines the key challenges encountered during the development of the supply chain performance dashboard, as well as the limitations associated with the data and modelling approach used in the project.

5.1 Data Limitations

A significant limitation of this project was the reliance on synthetically generated data. While every effort was made to simulate realistic supply chain conditions - such as category-specific lead times, price variations, order volumes, and stock levels - the data may not fully capture the nuances and irregularities of actual industry operations.

Additionally, certain assumptions had to be made to fill gaps that typically exist in real-world data:

- Availability of complete supplier and order information: In practice, data can be fragmented across systems, contain missing entries, or lack consistent formatting.
- Simplified demand patterns: Seasonal multipliers and noise were applied to simulate demand fluctuations; however, real-world demand is often influenced by external factors like promotions, market trends, or macroeconomic events.
- Absence of real-world constraints: Operational constraints such as procurement delays, production bottlenecks, and supplier negotiations were not modelled.

As a result, while the dashboard effectively demonstrates analytical capabilities, its outcomes are illustrative rather than prescriptive.

5.2 Model Limitations

The forecasting component of the project utilized time series models—specifically Prophet and Exponential Smoothing (ETS)—trained on synthetic historical data. Although these models provide interpretable and reasonably accurate forecasts within the synthetic environment, several limitations apply:

- Short training window: The models were trained on only two years of historical data (2022–2023), which limits their ability to capture long-term trends or rare events.
- Assumption of stationarity: ETS models assume relatively stable trends and seasonality, which may not generalize well if applied to volatile or irregular real-world data.
- Lack of exogenous variables: The forecasts were univariate and did not account for external regressors such as marketing campaigns, weather effects, or competitor activity, which could significantly affect demand accuracy in practice.
- No ensemble modeling or hyperparameter optimization: Model selection was based on performance metrics across default or minimally tuned configurations, leaving room for improvement using more advanced model tuning or hybrid/ensemble techniques.

Despite these limitations, the models served the intended purpose of demonstrating forecast integration into a Tableau dashboard and highlighting how forecast accuracy can vary by product or category.

5.3 Dashboard Limitations

While the dashboards were designed with user-centric interactivity and modular design principles, several limitations exist in their current implementation:

- Scalability with large datasets: The dashboards were optimized for the project's synthetic dataset, which was relatively small. Performance issues may arise when applied to large-scale, real-world enterprise data involving thousands of SKUs, suppliers, or transactions.
- Lack of mobile responsiveness: The dashboards were designed for desktop use. When viewed on tablets or mobile devices, elements such as filters and tooltips may not render optimally.
- Manual data refresh: The current Tableau implementation requires manual data refresh unless integrated with automated pipelines or APIs, which were not within the project scope.
- Limited multi-user personalization: The dashboards do not currently support user-specific views, access controls, or role-based filtering, which would be necessary for enterprise deployment.

- No predictive alerts or anomaly detection: While forecast data is visualized, the dashboards do not include automated alerts or anomaly detection mechanisms (e.g., flagging sudden demand spikes or supply delays).

Addressing these limitations would improve the dashboards' robustness, scalability, and enterprise readiness in a real-world deployment.

Chapter 6: Future Work & Improvements

While the current project demonstrates a functional and visually informative supply chain performance dashboard, there are multiple opportunities for extending its capabilities and improving its applicability to real-world scenarios. These opportunities span across data integration, forecasting methodologies, dashboard design, and automation.

6.1 Integration with Real-Time and Live Data Sources

One key area for improvement is the integration of the dashboard with real-time data pipelines. In an enterprise environment, inventory levels, supplier updates, and demand trends are often captured in real time through ERP systems, IoT devices, or sales platforms. Future implementations could leverage APIs or cloud-based connectors to automate data ingestion, enabling real-time visibility and reducing the need for manual refreshes.

6.2 Incorporation of External Variables in Forecasting

The current forecasting models rely solely on historical sales data. In practice, demand is often influenced by external drivers such as promotional campaigns, competitor activity, seasonality beyond fixed multipliers, weather conditions, and macroeconomic trends. Future work could involve building multivariate forecasting models that include external regressors, improving the accuracy and contextual relevance of demand forecasts.

6.3 Enhanced Predictive Analytics and Alerts

The dashboards currently function as descriptive and diagnostic tools. Future iterations could include predictive and prescriptive analytics, such as:

- Anomaly detection (e.g., flagging sudden stockouts or delivery failures)
- Safety stock optimization models
- Root cause analysis for supplier underperformance
- Trigger-based alerts (e.g., when forecast error exceeds a threshold or OTD drops below target)

These enhancements would support more proactive decision-making and operational agility.

6.4 Role-Based Access and Personalization

In an enterprise setting, users may require different levels of access and tailored views. For instance, procurement managers may focus on supplier KPIs, while inventory planners prioritize stock levels and reorder cycles. Future versions of the dashboard could incorporate user authentication and role-based filtering, enabling personalized insights based on user roles or departments.

6.5 Mobile Responsiveness and UI Improvements

To improve accessibility and adoption, the dashboard could be redesigned for mobile responsiveness. This would allow users to access insights on-the-go through tablets or smartphones. Additionally, improvements in UI layout, such as collapsible panels, search-enabled filters, and dark mode support, could further enhance user experience.

6.6 Evaluation of Real Users

A natural extension of this project would be to conduct formal usability testing and stakeholder interviews. This would provide feedback on dashboard layout, interactivity, and usefulness for real decision-making. Insights gathered could inform iterative improvements and validate the dashboard's effectiveness in a real business context.

Chapter 7: Conclusion

This project set out to develop an integrated and interactive supply chain performance dashboard using Tableau, with the objective of enhancing data visibility and supporting informed decision-making across three key domains: inventory management, supplier performance, and demand forecasting. Through a structured methodology that combined data engineering, statistical modelling, and dashboard design, a functional prototype was created that demonstrates the practical value of business intelligence (BI) tools in managing modern supply chain operations.

The project began with a critical evaluation of the raw dataset, identifying gaps and inconsistencies that necessitated extensive data cleaning and feature engineering. To overcome the absence of key attributes such as pricing, lead time, and inventory levels, synthetic data generation techniques were employed, ensuring a realistic and analysis-ready dataset. These enhancements enabled the calculation of important metrics such as Inventory Turnover Ratio (ITR), Days of Inventory on Hand (DOH), On-Time Delivery (OTD) rate, defect rate, and forecasting accuracy.

On the modelling front, two forecasting techniques - Exponential Smoothing (ETS) and Facebook Prophet - were evaluated on their ability to predict SKU-level monthly demand. The analysis revealed a trade-off between accuracy and interpretability, with Prophet exhibiting marginally superior performance in terms of Mean Absolute Percentage Error (MAPE) and Symmetric MAPE (SMAPE), particularly in capturing complex seasonal patterns.

The final dashboards were developed with an emphasis on user interactivity, modular design, and operational relevance. Dynamic filters, cross-filtering logic, and conditional formatting were applied to facilitate intuitive data exploration. The dashboards allow users to assess supplier reliability, monitor stock levels, and evaluate forecast accuracy, thereby enabling a more proactive and responsive supply chain strategy.

While the dashboard fulfils its design intent in a simulated environment, the project also acknowledges certain limitations, the reliance on synthetic data, the absence of real-time updates, and simplified assumptions in modelling supplier behaviour and demand drivers. Nevertheless, the framework developed is highly adaptable and provides a strong foundation for future extensions, including live data integration, anomaly detection, and role-based personalization.

In conclusion, this project highlights the importance of integrating descriptive analytics, forecasting models, and interactive visualization to build effective supply chain monitoring tools. By combining technical rigor with business context, the dashboard developed herein offers a scalable and practical solution to the challenges of supply chain visibility and performance optimization.

APPENDIX

A. Data Pre-Processing Mathematical Formulas and Technical Logic

A.1 Levenshtein Distance

Formally, for strings a and b, the distance $d(i,j)$ between the first i characters of a and the first j characters of b is defined recursively as:

$$d(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} d(i-1,j)+1 \\ d(i,j-1)+1 \\ d(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases}$$

The resulting distance is normalized to compute a similarity score between 0 and 100, where a higher score indicates greater similarity. This score can be expressed as:

$$Fuzzy\ Score = \left(1 - \frac{d(a,b)}{\max(|a|,|b|)}\right) * 100$$

As explained in the text, a similarity threshold of 95% was selected to ensure that only highly similar entries were matched, minimizing the risk of incorrect merges. For example, descriptions like "Organic Apple Juice 1L" and "Organic Apple Juice - 1 Liter" were recognized as matches and consolidated under a unified entry. The matching process was followed by a manual validation step to ensure accuracy, particularly for frequently sold SKUs (stock keeping units).

A.2 Z-Score Outlier Detection

$$Z = \frac{X - \mu}{\sigma}$$

- X: observed sales value
- μ: mean sales value
- σ: standard deviation of sales

Entries with Z-scores exceeding a predefined threshold (typically $|Z|>3$) were flagged for manual review. For values without business justification, replacement was performed using the median sales value of the corresponding product category to maintain a realistic and context-sensitive distribution.

B. Feature Engineering Logic & Detailed Steps

B.1 Price Information (price_info)

1. Category-Level Price Assignment

Using publicly available retail pricing data and industry benchmarks, average unit costs were assigned to each major product category. For example, product groups such as WINE, LIQUOR, and BEER were assigned higher average unit prices compared to NON-ALCOHOLIC beverages or general STORE SUPPLIES. This stratification reflects typical market trends in beverage retail pricing [36], [37].

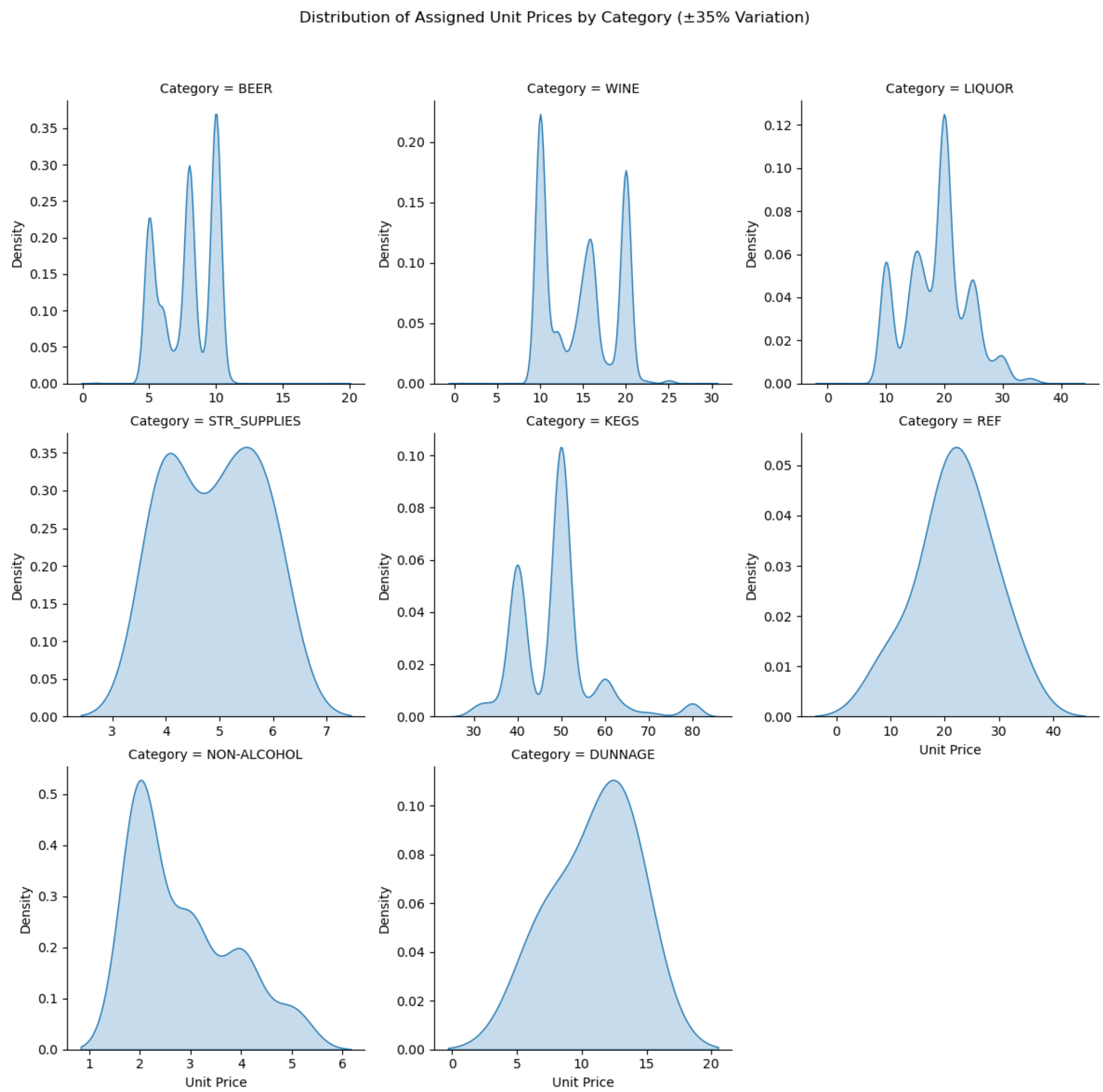
Item Category	Average Unit Cost (USD)
BEER	8
WINE	15
LIQUOR	20
STR_SUPPLIES	5
KEGS	50
REF	25
NON-ALCOHOL	3
DUNNAGE	10

2. Introducing Controlled Price Variation

To simulate realistic variability within each category, a controlled random adjustment was applied using a uniform distribution. The adjusted price for each SKU was calculated as:

$$Unit\ Price = Average\ Category\ Price \times (1 + \epsilon)$$

Where $0 \sim U(-0.35, +0.35)$, representing a $\pm 35\%$ variation. This allowed each SKU to deviate slightly from the category mean while maintaining overall category-level price structure. This step ensured diversity in unit prices while avoiding unrealistic outliers.



3. Rounding for Logical Retail Accuracy

To reflect typical pricing conventions observed in retail settings, each generated unit price was rounded to the nearest 0.5 increment (e.g., \$3.0, \$6.5, \$10.0). Retail prices typically follow simple conventions such as rounding to the nearest 0.5 or 1.0. To reflect this, each generated price was rounded using the function:

$$Rounded\ Price = \frac{round(2\ \times\ Unit\ Price)}{2}$$

This logic rounds values to the nearest 0.5 increment (e.g., \$3.15 → \$3.0, \$5.76 → \$6.0), creating more realistic retail price points. This step was essential in avoiding excessively granular pricing and ensuring consistency across SKUs.

4. Estimating Units Sold

The original *Sales* column captured aggregated sales value per SKU per month, but no quantity information was available. Based on observed values and plausible beverage pricing logic, it was assumed that sales were recorded in thousands of dollars. Hence, units sold were estimated using the following formula:

$$Units\ Sold = \frac{Sales\ \times\ 1000}{Unit\ Price}$$

This allowed conversion of revenue into approximate quantity sold, enabling calculations of operational metrics such as forecast accuracy, safety stock, and reorder quantities.

B.2 Inventory Estimation (inventory_info)

1. Initializing Beginning Inventory

For the first month of each SKU's timeline, beginning inventory was estimated by applying a random buffer to the number of units sold. This buffer simulated safety stock typically held in inventory and was defined by:

$$Beginning\ Inventory = Units\ Sold\ \times\ (1+ \in) \qquad where\ \epsilon \sim U(0.10, 0.30)$$

This variability was introduced using *np.random.uniform(0.10, 0.50),* reflecting a 10% to 50% buffer over expected demand. For subsequent months, beginning inventory was initialized based on the ending inventory of the previous month, ensuring continuity:

$$Beginning\ Inventory_t\ \geq Ending\ Inventory_{t-1}$$

2. Generating Ending Inventory

Ending Inventory for each SKU was calculated as:

$$Ending\ Inventory = Beginning\ Inventory - Units\ Sold$$

If the result was zero, this was treated as a *stockout event*, indicating that all available inventory was consumed.

3. Simulating Stockouts

To realistically model occasional stock unavailability, a 30% probability of stockout was introduced using *np.random.choice().* When a stockout was simulated, the beginning inventory was force-set equal to the units sold, ensuring that the ending inventory would be zero. This logic models real-world disruptions or delayed replenishments.

4. Validation and Logical Adjustments

To ensure logical consistency in inventory flows, the following conditions were validated and corrected through iterative checks:

• Beginning inventory was ensured to be greater than or equal to the previous month's ending inventory.

• If no stockout occurred, beginning inventory was set to at least the number of units sold to avoid negative ending inventory.
• Ending inventory was never allowed to fall below zero.

These rules were enforced using sequential loops and conditional logic in Python to reflect realistic inventory transitions month-over-month

B.3 Lead Time Data (leadtime_info)

1. Defining Category-Based Lead Time Ranges

Realistic lead time intervals were defined for each product category based on domain knowledge, logistics standards, and retail sourcing practices. The lead time ranges were assigned based on common retail sourcing patterns and logistics expectations, informed by industry reports, retail delivery timelines, and supply chain literature [38] – [41]. Categories with higher complexity or longer procurement cycles, such as WINE and LIQUOR, were assigned longer lead time ranges (e.g., 10–20 days), whereas readily available products like NON-ALCOHOLIC beverages or STORE SUPPLIES were assigned shorter ranges (e.g., 3–7 days).

Item Category	Assigned Lead Time Range (days)
BEER	14-21
WINE	7-14
LIQUOR	10-20
STR_SUPPLIES	14-21
KEGS	3-7
REF	5-10
NON-ALCOHOL	5-10
DUNNAGE	3-7

2. Randomized Lead Time Assignment

To reflect natural variability within categories, a custom function estimate_lead_time() was defined to randomly assign an integer lead time to each SKU using NumPy's np.random.randint() function:

$$Lead\ Time_{SKU} \sim U(a, b)$$

Where a and b are the lower and upper bounds of the lead time range for the SKUs category. This ensured that SKUs within the same category had differing but plausible lead times, simulating the variability observed in real-world vendor performance

B.4 Suppliers Orders Date (orders_info)

1. Order Initialization

To generate realistic order records, the simulation began by constructing a pool of unique SKU–Category–Supplier combinations from the cleaned dataset. From this pool, individual order entries were created through uniform random sampling, meaning each possible combination had an equal probability of being selected regardless of product type or supplier. This method avoids bias toward any particular subset of the data and ensures wide coverage across all SKUs and vendors.

In implementation, this was achieved using *np.random.choice()* applied to the index of the unique combinations list. For each order, an index was selected at random from a uniform distribution over the range $[0, N-1]$, where N is the total number of unique combinations. The sampling was done with replacement, allowing the same combination to appear in multiple synthetic orders, simulating real-world scenarios where products are ordered repeatedly over time.

This logic was used to simulate approximately 170,000 supplier order events, ensuring diversity in the dataset while maintaining a simple and interpretable generation process.

2. Availability Rate Esimatation

A *Beta distribution* with parameters 0was used to generate realistic supplier availability rates. This distribution is right-skewed, favouring high availability values between 80% and 100%, while still allowing some variance:

$$Availability\ Rate \sim Beta(\alpha = 5, \beta = 2)$$

Each order was then randomly assigned a binary *Availability* flag based on whether the generated probability exceeded a threshold, indicating whether the supplier fulfilled the order or not.

3. Order Quantity and Placement Date

The quantity ordered for each record was generated as an integer sampled uniformly between 1 and 1000 units:

$$Quantity\ Ordered \sim U\{1{,}1000\}$$

This range was selected to represent the typical volume of purchases in a retail environment, ranging from low-quantity trial orders to high-volume replenishment batches. The use of a discrete uniform distribution ensures all quantities within the specified range have equal probability, which avoids introducing skew into the simulated procurement volume.

Order placement dates were generated to simulate realistic transaction activity distributed over the past 12 months. A start date was defined as one year prior to the data generation date, and for each order, a random number of days d was drawn from a uniform distribution:

$$Order\ Placed\ Date = Start\ Date + d, \qquad d \sim U(0, 365)$$

This method ensures that orders are spread throughout the year without clustering, which reflects typical order frequency and seasonality in large retail datasets. The Python *datetime* and *timedelta* libraries were used to compute the actual calendar date values.

4. Delivery Date Simulation

The *Expected Delivery Date* was computed by merging the *orders_info* dataset with *leadtime_info* and adding the assigned lead time (in days) to the order placed date. To simulate delivery variability, a random offset $\delta \sim U(-3, +3)$ days was added to derive the *Actual Delivery Date*:

$$Actual\ Delivery\ Date = Expected\ Delivery\ Date + \delta$$

5. Delivery Status Classification

Each order was labelled as *early, on-time*, or *late* based on a comparison between actual and expected delivery dates. This Delivery Status flag was defined as follows:

- On-time: Delivered on the expected date.
- Early: Delivered more than one day before the expected date.
- Late: Delivered after the expected date.

These classifications enabled assessment of on-time delivery rates (OTD) across suppliers.

6. Defect Flag Generation

In real-world supply chains, a small percentage of supplier orders are expected to contain defects, such as damaged packaging, expired goods, or incorrectly shipped items. To incorporate this dimension of quality control into the simulation, a binary defect flag was introduced to each order record.

The defect flag was generated using a *Bernoulli process*, where each order has a fixed probability $p = 0.02$ of being defective. In other words:

$$Defect\ Flag \sim Bernoulli(p = 0.02)$$

This was implemented in Python using the *np.random.choice()* function with a weighted probability vector $[0.98, 0.02]$, assigning a value of 1 to indicate a defect and 0 otherwise. The 2% defect rate was selected based on common benchmarks observed in the beverage retail industry and literature, where well-managed suppliers typically maintain defect rates below 5% [42].

Orders flagged as defective were treated independently of quantity accuracy, allowing for scenarios where an order might be accurate in quantity but still have quality issues. This feature supports the computation of Defect Rate per Supplier, which is a core metric visualized in the Supplier Performance Dashboard.

7. Quantity Accuracy Evaluation

Realistic fulfillment behavior was simulated by allowing quantity deviations between the ordered and received quantities. A tolerance of 95% was set, and the *Quantity Accuracy* flag was defined as:

$$Quantity\ Accuracy\ Flag = \begin{cases} 1, & if\ \dfrac{Quantity\ Received}{Quantity\ Ordered} \geq 0.95 \\ 0, & otherwise \end{cases}$$

The received quantity was generated by applying a multiplier from a *Beta distribution*, $Beta(9,1)$, skewing toward high accuracy with slight under-deliveries.

B.5 Extended Dataset (main_extended)

1. Seasonality Multiplier Design

As retail demand tends to vary systematically throughout the year, a set of monthly seasonality multipliers was first constructed to simulate realistic fluctuation in demand. These multipliers were informed by domain knowledge and common retail trends (e.g., increased spending in November and December, reduced activity in February). Each month was assigned a base multiplier, which was then slightly adjusted across years to reflect natural year-over-year variation.

Month	Descripition	Multiplier
Jan	Post-Holiday Dip	0.85
Feb	Lowest Demand	0.80
Mar	Slight Recovery	0.95
Apr	Average Sales	1.00
May	Small Peak	1.10
Jun	Summer Demand Rises	1.05
Jul	Peak Consumption	1.15
Aug	Summer Demand Continues	1.20
Sep	Stabilization	1.00
Oct	Pre-Holiday Stocking	1.05
Nov	Start of Holiday Spike	1.10
Dec	Holiday & New Year Peak	1.40

These multipliers were embedded into the logic used to generate synthetic sales values, thereby encoding consistent seasonality patterns into the historical data.

2. Synthetic Sales Generation Logic

To simulate monthly sales for each SKU in the years 2022 and 2023, a function named *generate_historical_sales()* was defined. This function used the 2024 dataset as a baseline reference, drawing on cleaned sales values while applying transformations to reflect seasonal and stochastic variation. The overall structure for each synthetic sales value $S_{y,m,i}$ (for year y, month m, and SKU i) was defined as:

$$S_{y,m,i} = S_{2024,m,i} \times M_{y,m} \times \epsilon$$

Where:

- $S_{2024,m,i}$: baseline monthly sales of SKU i in month m from 2024 (or a similar SKU if not available)
- $M_{y,m}$: seasonality multiplier for month m, in year y
- $\epsilon \sim U(0.85,0.15)$: random scaling factor to introduce variability

A 65% probability was applied per SKU-month combination to determine whether a sales record should be generated using:

$$P(Generate\ Sales_{y,m,i}) = 0.65$$

When a sale was generated, a random integer $n \in \{1,2,3\}$ was drawn to simulate the number of times that SKU was sold that month. The individual events were summed to produce a final value for that SKU–month. Each synthetic row was then rounded to the nearest whole number to reflect typical sales reporting granularity.

3. Validation and Quality Checks

To ensure the integrity and realism of the synthetic dataset, several validation checks were implemented:

- Uniqueness: Each SKU-month pair was guaranteed to appear only once to avoid duplication.
- Positive Sales: Any zero or negative sales values were filtered out to ensure logical consistency.
- Scale Calibration: The row counts for 2022 and 2023 were tuned to approximate the scale of 2024, without exact replication, maintaining plausible historical volume.
- Seasonality Realism: Aggregated monthly sales trends were visually inspected to confirm that expected seasonal peaks (e.g., December) and troughs (e.g., February) were present, validating the effect of the applied multipliers

REFERENCES

[1] S. Vilarinho, I. Lopes, and S. Sousa, "Developing dashboards for SMEs to improve performance of productive equipment and processes," Journal of Industrial Information Integration, vol. 12, pp. 13–22, Dec. 2018, doi: https://doi.org/10.1016/j.jii.2018.02.003.

[2] G. C. Souza, "Supply chain analytics," Business Horizons, vol. 57, no. 5, pp. 595–605, Sep. 2014, doi: https://doi.org/10.1016/j.bushor.2014.06.004.

[3] N. Stefanovic, "Proactive Supply Chain Performance Management with Predictive Analytics," The Scientific World Journal, vol. 2014, no. 528917, pp. 1–17, 2014, doi: https://doi.org/10.1155/2014/528917.

[4] N. R. Sanders, "How to Use Big Data to Drive Your Supply Chain," California Management Review, vol. 58, no. 3, pp. 26–48, May 2016.

[5] P. Nair, "Tackling Supply Chain Management through Business Analytics: Opportunities and Challenges," *Advances in intelligent systems and computing*, Jan. 2015, doi: https://doi.org/10.1007/978-3-319-13728-5_64.

[6] G. C. Souza, "Supply chain analytics," Business Horizons, vol. 57, no. 5, pp. 595–605, Sep. 2014, doi: https://doi.org/10.1016/j.bushor.2014.06.004.

[7] K. Govindan, T. C. E. Cheng, N. Mishra, and N. Shukla, "Big data analytics and application for logistics and supply chain management," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 343–349, Jun. 2018, doi: https://doi.org/10.1016/j.tre.2018.03.011.

[8] "SAP Integrated Business Solutions," *NTT DATA Business Solutions Singapore*, Jan. 27, 2025. https://nttdata-solutions.com/sg-en/products/sap-integrated-business-planning/

[9] "SAP IBP real time insights for business," *SAP Community*, Nov. 13, 2019. https://community.sap.com/t5/supply-chain-management-blogs-by-members/sap-ibp-real-time-insights-for-business/ba-p/13440017

[10] Oracle, "Supply Chain Management (SCM) and Manufacturing | Oracle," *www.oracle.com*, 2023. https://www.oracle.com/scm/

[11] "Coca-Cola Bottling Company empowers the enterprise with Tableau Mobile dashboards to drive bottom line," *Tableau Software*. https://www.tableau.com/solutions/customer/eliminating-reporting-bottleneck-ccbc

[12] "Lenovo India increases reporting efficiency by 95% across the enterprise," *Tableau*. https://www.tableau.com/solutions/customer/lenovo-increases-reporting-efficiency-across-enterprise

[13] “Inventory Visibility Power BI dashboard - Supply Chain Management | Dynamics 365,” *Microsoft.com*, Sep. 27, 2023. https://learn.microsoft.com/en-us/dynamics365/supply-chain/inventory/inventory-visibility-dashboard.

[14] M. A. Waller and S. E. Fawcett, “Data Science, Predictive Analytics, and Big Data: a Revolution That Will Transform Supply Chain Design and Management,” *Journal of Business Logistics*, vol. 34, no. 2, pp. 77–84, Jun. 2013.

[15] J. Zhang and M. Govindan, "Supply chain performance measurement using key performance indicators and analytics," *International Journal of Production Economics*, vol. 235, p. 108109, 2021.

[16] P. M. Childerhouse, E. D. Ivanov, and D. A. Waters, "Inventory management and supply chain performance: A systematic literature review," *Journal of Supply Chain Management*, vol. 58, no. 3, pp. 34–49, 2022.

[17] T. Wu, X. Huang, and K. N. Boyer, "Mitigating stockouts and excess inventory with predictive analytics: Evidence from the retail sector," *Journal of Operations Management*, vol. 74, no. 1, pp. 23–39, 2023.

[18] S. Chopra and P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*, 8th ed., Upper Saddle River, NJ, USA: Pearson, 2023.

[19] L. Zhao, A. Ghani, and J. Wang, "Service level optimization in retail supply chains: A data-driven approach," *Decision Support Systems*, vol. 173, p. 113850, 2022.

[20] “Service Rate: Calculation, Importance, and Improvements,” *Nomadia*, Aug. 23, 2023. https://www.nomadia-group.com/en/resources/blog/service-rate-calculation-importance-and-improvements/

[21] F. Akyuz and T. Erkan, "Performance measurement practices in supply chain management: A literature review," *International Journal of Production Research*, vol. 59, no. 5, pp. 1–23, 2021.

[22] “Inderscience Publishers - linking academia, business and industry through research,” *Inderscience.com*, 2025. https://www.inderscience.com/offers.php?id=62328 (accessed Apr. 11, 2025).

[23] M. R. Shamsuzzoha and P. M. Helo, "Supplier evaluation and performance monitoring in supply chains: Trends and best practices," *Operations Management Research*, vol. 15, no. 2, pp. 45–61, 2022.

[24] L. Tang, X. Chen, and Y. Liu, "Evaluating supplier reliability using lead time variability and delivery performance metrics," *Journal of Manufacturing Systems*, vol. 67, pp. 92–105, 2023.

[25] Anitha P and M. M. Patil, “A Review on Data Analytics for Supply Chain Management: A Case study,” *International Journal of Information Engineering and Electronic Business*, vol. 10, no. 5, p. 30, 2018, Available: https://www.mecs-press.org/ijieeb/ijieeb-v10-n5/v10n5-5.html

[26] A. Nguyen, S. Lamouri, R. Pellerin, S. Tamayo, and B. Lekens, “Data analytics in pharmaceutical supply chains: state of the art, opportunities, and challenges,” *International Journal of Production Research*, vol. 60, no. 22, pp. 1–20, Jul. 2021, doi: https://doi.org/10.1080/00207543.2021.1950937.

[27] M. D. Shamout, “Does Supply Chain Analytics Enhance Supply Chain Innovation and Robustness Capability?,” *Organizacija*, vol. 52, no. 2, pp. 95–106, May 2019, doi: https://doi.org/10.2478/orga-2019-0007.

[28] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.

[29] “Forecasting using R Rob J Hyndman 2.1 State space models.” Accessed: Apr. 11, 2025. [Online]. Available: https://robjhyndman.com/eindhoven/2-1-StateSpace.pdf

[30] S. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[31] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 15, no. 3, p. e0232531, 2020.

[32] Data.gov, "Warehouse and Retail Sales Dataset," [Online]. Available: https://catalog.data.gov/dataset/warehouse-and-retail-sales

[33] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[34] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 6th ed., Hoboken, NJ, USA: Wiley, 2014.

[35] Nunung Nurhasanah, None Machfud, Djumali Mangunwidjaja, and Muhammad Romli, “A conceptual framework on the design of intelligent supply chain for natural fibre agroindustry,” *AIP conference proceedings*, vol. 2217, pp. 030050–030050, Jan. 2020, doi: https://doi.org/10.1063/5.0000742.

[36] Statista, "Average price of wine, beer, and spirits in the U.S.," 2023. [Online]. Available: https://www.statista.com

[37] Total Wine & More, "Shop by Category – Beverage Pricing," 2023. [Online]. Available: https://www.totalwine.com

[38] DHL Supply Chain, "A Guide to Supply Chain Lead Times in Retail," *DHL Logistics Insights*, 2022. [Online]. Available: https://www.dhl.com

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

[40] Total Wine & More, "Shipping & Delivery Information," 2023. [Online]. Available: https://www.totalwine.com

[41] U.S. Alcohol and Tobacco Tax and Trade Bureau (TTB), "Alcohol Importation Process Overview," 2023. [Online]. Available: https://www.ttb.gov

[42] M. Christopher, *Logistics & Supply Chain Management*, 5th ed., Pearson Education, 2016.

[43] E. Hofmann and E. Rutschmann, “Big data analytics and demand forecasting in supply chains: a conceptual analysis,” *The International Journal of Logistics Management*, vol. 29, no. 2, pp. 739–766, May 2018, doi: https://doi.org/10.1108/ijlm-04-2017-0088.