PGPDSE FT CAPSTONE PROJECT – REPORT

AI-DRIVEN STOCKOUT RISK PREDICTION FOR SMARTER RETAIL INVENTORY MANAGEMENT

Mentor: Mr. Aishwarya Sarda

Date: 26 May 2025

Submitted by:

Cheran K

Het Patel

Medlynn M

Nishant B M

Siddesh S Menon

TABLE OF CONTENTS

1. INTRODUCTION

Stockouts remain a critical challenge in retail supply chains, often leading to lost sales, reduced customer satisfaction, and inefficient inventory management. With the growing availability of data and advancements in machine learning, businesses can now move beyond reactive inventory practices toward predictive, data-driven strategies.

This project focuses on developing an AI-driven stockout risk prediction system using historical retail data. By analyzing inventory levels, sales trends, promotions, and external factors, the goal is to accurately identify potential stockouts before they occur. The report outlines the end-to-end process - from data exploration and preprocessing to model development, evaluation, and deployment - aimed at supporting smarter inventory decisions and minimizing product unavailability.

2. INDUSTRY REVIEW

2.1. Current practices, Background Research

- **Overview of the Retail Industry**

  Retail businesses operate on thin margins where product availability plays a crucial role in customer satisfaction and sales. Ensuring the right stock at the right time is key to minimizing lost revenue from stockouts.

- **Current Practices** in Managing Stockouts Most retailers still rely on reactive systems or simple rules based on past sales to manage inventory levels. Inventory thresholds are often set manually and do not consider dynamic factors such as promotions, weather, or competitor behavior.

- **Challenges in Stockout Prevention**

  - Inaccurate Forecasting leads to unexpected stockouts during high-demand periods.
  - Static Thresholds ignore variability across regions, seasons, and store locations.
  - Lack of Real-Time Analysis reduces responsiveness to sudden demand shifts.
  - Evolving Industry Trends to reduce stockouts, companies are turning to:
    - Predictive analytics and machine learning models for risk assessment.
    - Automated restocking based on real-time insights.
    - Incorporating external variables like weather, holidays, and competitor pricing into forecasting systems.

2.2. Literature Survey - Publications, Application, past and undergoing research

- **Traditional Approaches to Stock Management**

  Initial research focused on sales forecasting using statistical models like ARIMA and exponential smoothing. However, these models often fall short when applied to stockout prevention due to their inability to capture real-world complexity.

- **Stockout Risk Prediction Using Machine Learning**
  Recent studies approach stockout prediction as a classification problem:

- Kumar et al. (2021) used logistic regression and decision trees to predict whether inventory would fall below critical levels.
- Zhang & Zhao (2020) used ensemble models (XGBoost, Random Forest) to classify stockout risk with promising accuracy.

- **Features Used in Prediction Models**

  Common input features in literature include:
  - Inventory levels
  - Sales and order history
  - Promotion and holiday indicators
  - Competitor pricing
  - Weather and seasonality

- **External Influences on Stockouts**

  Research like **Ahmed et al. (2018)** and **Banerjee et al. (2022)** shows that integrating external factors significantly enhances the performance of stockout prediction models.

- **Gaps in Existing Research**

  - Many models focus on single-store or single-variable forecasting.
  - Few studies combine diverse features like weather, regional differences, and seasonal demand in one integrated model.
  - There is limited attention to real-time, explainable stockout risk classification, which is essential for practical decision-making—this is the focus of our project.

3. DATASET AND DOMAIN

3.1. Introduction

This project focuses on the retail supply chain domain, specifically addressing the challenge of stockout prediction. The dataset was sourced from Kaggle and contains historical retail sales, inventory, and promotion data.
Dataset Link: Retail Store Inventory Forecasting Dataset

3.2. Data Dictionary

- **Date**: Daily records from 01-01-2022 to 01-01-2024
- **Store ID & Product ID**: Unique identifiers for stores and products
- **Category**: Product categories like Electronics, Clothing, Groceries, etc
- **Region**: Geographic region of the store
- **Inventory Level**: Stock available at the beginning of the day
- **Units Sold**: Units sold during the day
- **Units Ordered**: Number of units ordered by the store (restocking quantity)
- **Demand Forecast**: Predicted demand based on past trends
- **Price**: Selling price of the product on that date

- **Discount**: Discount applied to the product (in percentage)
- **Weather Condition**: Daily weather impacting sales
- **Holiday/Promotion**: Indicators for holidays or promotions
- **Competitor Pricing**: Price of a similar product from a competitor on the same date
- **Seasonality**: Season corresponding to the date (e.g., Winter, Summer)

Variable Name	Data Type	Unique Values	Missing Values	Sample / Range
Date	object	731	0	2022-01-01 to 2024-01-01
Store ID	object	5	0	S001, S002, S003...
Product ID	object	20	0	P0001, P0002, P0003...
Category	object	5	0	Groceries, Toys, Electronics...
Region	object	4	0	North, South, East, West
Inventory Level	int64	451	0	50 to 500
Units Sold	int64	498	0	0 to 499
Units Ordered	int64	181	0	20 to 200
Demand Forecast	float64	31608	0	-9.99 to 518.55
Price	float64	8999	0	10.0 to 100.0
Discount	object	5	0	0, 5, 10, 15, 20
Weather Condition	object	4	0	Rainy, Sunny, Cloudy, Snowy
Holiday/Promotion	object	2	0	Yes, No
Competitor Pricing	float64	320	0	Variable
Seasonality	object	4	0	Winter, Spring, Summer, Autumn

Table: Variable description

3.3. Variable categorization (count of numeric and categorical)

- Numeric Features - 6 features
- Categoric Features – 8 features

3.4. Pre Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

- There are no missing/null values in the dataset
- No redundant columns

3.5. Project Justification

3.5.1. Project Statement

In the retail industry, stockouts pose a serious problem since they result in decreased sales, unhappy customers, and higher operating expenses. Accurately forecasting demand changes is a challenge for traditional inventory management. The goal of this project is to create a predictive framework based on machine learning that uses past sales, inventory data, and outside variables like promotions and weather to categorize products as either High Risk or Low Risk of stockouts. The approach will increase supply chain efficiency, reduce missed sales, and improve inventory management by integrating real-time monitoring and improving replenishment tactics.

3.5.2. Complexity Involved

- **Data Challenges:**
  - **Variety and Integration of Data**: The dataset contains information from both internal and external sources, including Weather Condition, Seasonality, Price, Discount, Demand Forecast, Inventory Level, Units Ordered, Units Sold, Holiday/Promotion, etc. Advanced data

preprocessing is necessary to integrate these heterogeneous variables for meaningful prediction.

  - **Granularity**: Managing variations across product categories and timeframes requires thoughtful handling.
- **Feature Engineering:**
  - Designing meaningful features from sales and inventory data is critical but time-consuming. For example, handling seasonality, trend analysis, and lag effects can be intricate.
- **Model Selection and Optimization**:
  - Choosing the right classification algorithm is challenging. Balancing accuracy and interpretability are essential, particularly for commercial use.
  - Hyperparameter tuning and addressing class imbalances (e.g., more Low Risk than High Risk stockouts) require rigorous experimentation.
- **Real-Time Implementation**:
  - Incorporating real-time data updates and predictions in a dynamic retail environment demands robust infrastructure.
  - Ensuring scalability and system reliability during peak seasons adds complexity.
- **Stakeholder Adoption**:
  - Convincing stakeholders to trust and adopt the predictive framework requires clear insights and demonstrable benefits.
  - Tailoring the user interface for intuitive interaction with the system may be necessary.

3.5.3. Project Outcomes

- **Commercial Value**:
  - **Improved Efficiency**: Enhanced inventory management reduces stockout-related losses and operational costs.
  - **Revenue Growth**: By preventing stockouts, retailers capture lost sales, increasing profitability.
  - **Customer Satisfaction**: A reliable inventory system fosters customer loyalty and brand reputation.
- **Academic Value**:
  - **Algorithm Innovation**: The work could lead to advancements in predictive modelling tailored to retail applications.
  - **Data Science Research**: Publication opportunities showcasing how external variables impact inventory decisions.
- **Social Value**:
  - **Reduced Waste**: Efficient inventory management lowers excess stock, aligning with sustainability goals.
  - **Community Impact**: Smooth operations help smaller retailers compete in crowded markets.

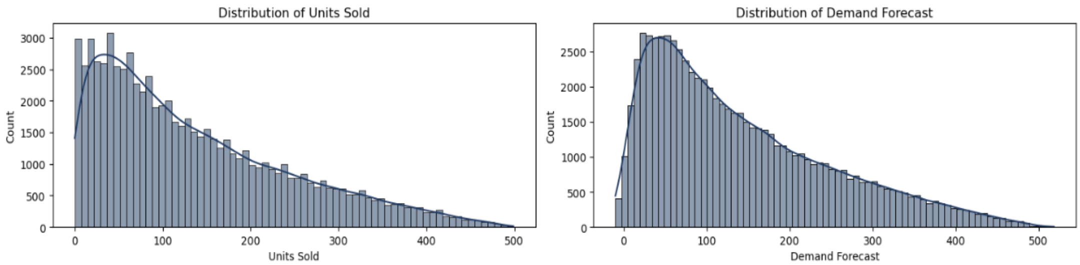
4. DATA EXPLORATION (EDA)

4.1. Introduction

Exploratory Data Analysis (EDA) was conducted to understand the structure, quality, and distribution of the dataset. This step involved identifying data types, detecting missing values, examining class balance, and uncovering relationships between features. EDA provided critical insights that guided feature selection, preprocessing strategies, and model development.

4.2. Distribution of variables

4.2.1. Numeric Features – Univariate Analysis

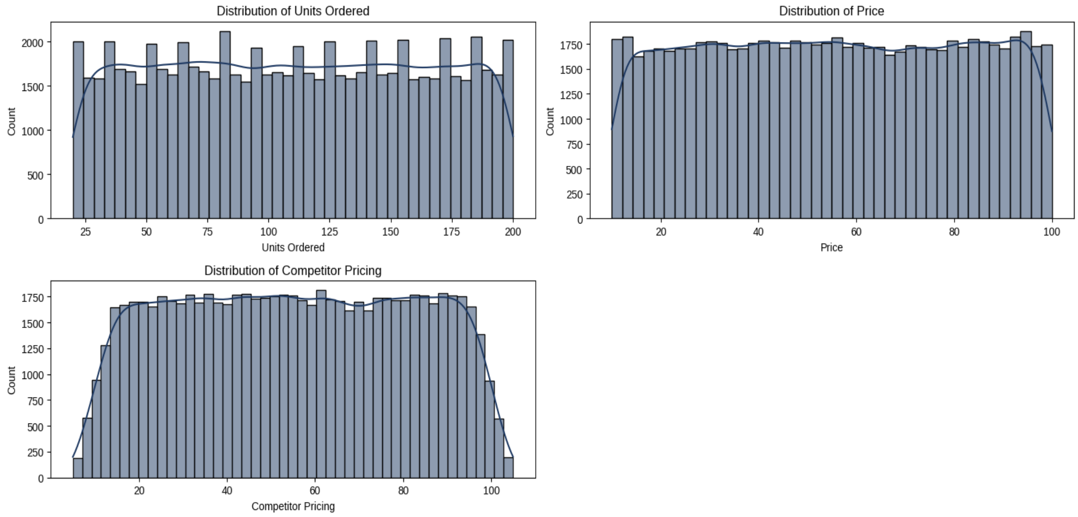


Units Sold:

- The distribution is **right skewed** with a skewness of **0.905**, with most sales occurring below 100 units.
- 673 records have demand forecast negative observation; demand forecast cannot be negative; these values will be replaced with 0.
- Few high-selling products create a long tail, indicating **demand variability**.

Demand Forecast:

- The demand forecast is **right skewed** with a skewness of **0.895**, with most products having forecasted demand under 100 units.
- This indicates that low-to-moderate demand products dominate the dataset.



Units Ordered:

- The distribution is **uniform**, showing that units are ordered in a consistent pattern across all quantities.
- This suggests **static restocking rules**, potentially disconnected from actual demand

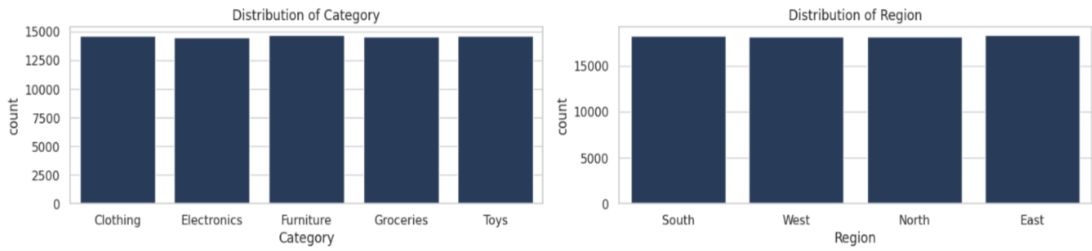
Price:

- Product pricing appears **evenly distributed** between 10 and 100, implying broad pricing strategies.
- There's **no visible skew**, meaning all price points are equally represented.

Competitor Pricing:

- Like product price, competitor pricing is also **uniformly distributed**.
- This reflects a **competitive and balanced market** across pricing tiers.

4.2.2. Categoric Features – Univariate Analysis



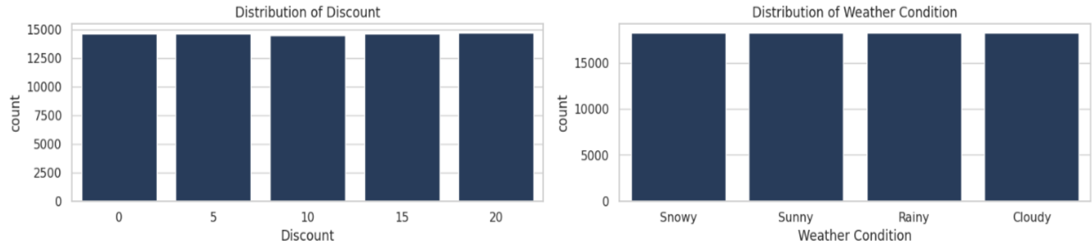
Category:

- All product categories (Clothing, Electronics, Furniture, Groceries, Toys) are **evenly distributed**.
- Demand trends and supply chain limitations probably differ for different product kinds.
- This feature can help the model learn category-specific stockout tendencies.

Region:

- All regions (South, West, North, East) have almost **equal representation**.

- This ensures no regional bias in the data - good for model generalization.

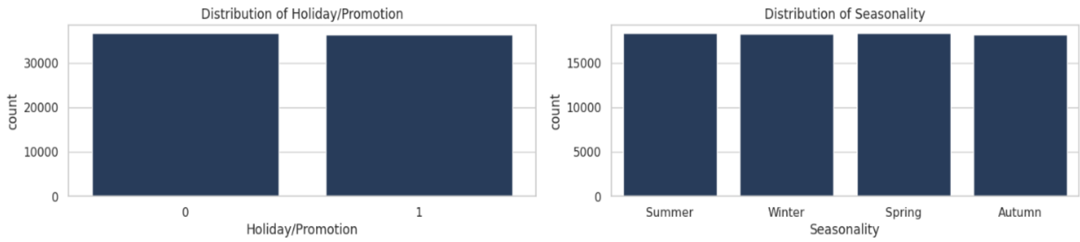


Discount:

- Discounts are **evenly distributed** across all ranges.
- Suggests systematic discount strategies likely applied across products.

Weather Condition:

- All four Weather types appear **almost equally**, with no dominant condition.
- Weather variability is well-captured, which could be useful for demand shifts.



Holiday/Promotion:

- The data is roughly split 50-50 between promotional and non-promotional periods.
- This balance is helpful for evaluating how promotions impact stockout risk.

Seasonality:

- The column shows inconsistencies, with different seasons assigned to the same date and region, indicating data quality issues.
- To resolve this, a new column will be engineered based on accurate date logic, and the old column is retained only for univariate analysis reference.

Feature engineering is performed on the available columns to enhance model interpretability and predictive power.

Features added to modelling dataset:

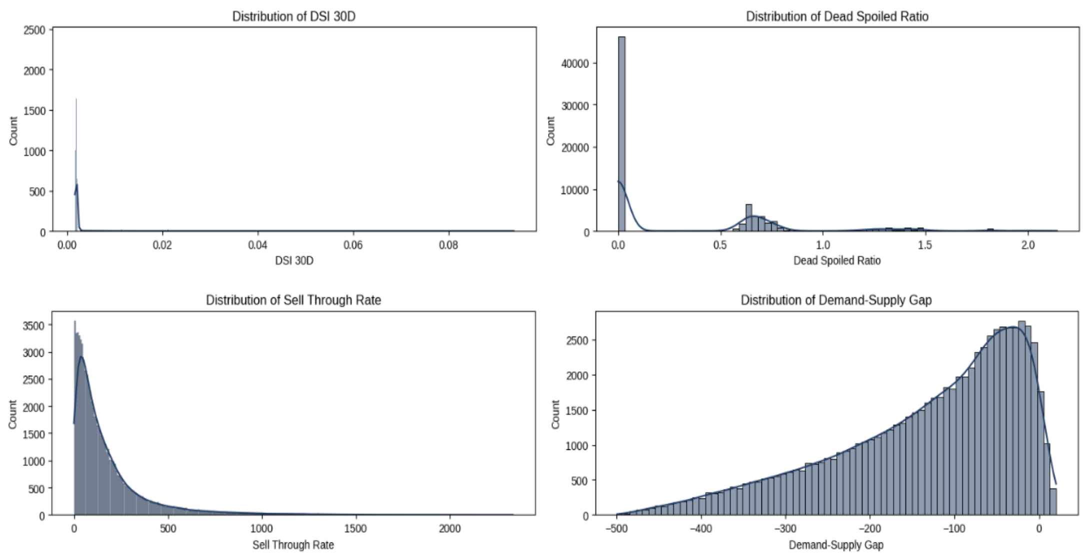
- **Month**: Extracted from date
- **Seasonality**: Replaced **old Seasonality** column and is derived based on date logic
- **DSI 30D**: Days Sales of Inventory (30-Day Rolling)

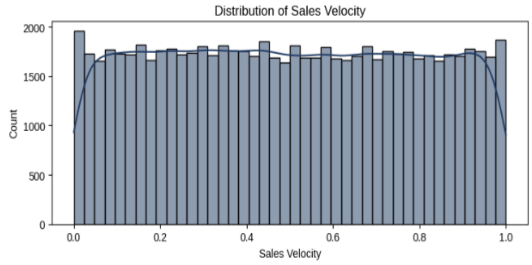
- The 30-day rolling DSI was derived by dividing the average inventory over 30 days by the total cost of goods sold (Units Sold x Price) in the same period, then multiplying by 30. It reflects how many days, on average, inventory remains unsold, calculated per category and date.
- **Dead Spoiled Ratio**: The Dead Spoiled Ratio quantifies the percentage of days a product had inventory but failed to sell despite forecasted demand. It is calculated as the ratio of unsellable days to total available stock days per product, indicating potential dead or spoiled stock.
- **Is Weekend**: A binary feature indicating whether the transaction occurred on a weekend, capturing potential patterns in customer behaviour and sales fluctuations.
- **Sell Through Rate**: Sell Through Rate measures the percentage of ordered units that were sold, offering insights into inventory efficiency and product demand.
- **Demand-Supply Gap**: Calculated as the difference between demand forecast and current inventory level, this feature reflects potential understock or overstock situations.
- **Sales Velocity**: Indicates how quickly inventory is turning over, computed as units sold divided by inventory level.

Features eliminated from modelling dataset:

- **Date**: It does not directly contribute predictive value in its raw form.
- **Store ID**: It is a categorical identifier without inherent predictive value and could introduce high cardinality noise.
- **Product ID**: It is removed to avoid overfitting to specific items, as it functions solely as a unique identifier rather than a predictive feature.
- **Inventory Level**: Target variable leakage as Inventory Level is used in the classification of Stockout Risk
- **Units Sold**: Including them can lead to overly optimistic model performance and reduce the generalizability of the results.
- **Competitor Pricing**: Due to multicollinearity, which will be addressed in a separate section.

4.2.3. Numeric Features – Univariate Analysis of Feature Engineered Columns





DSI 30D:

- The distribution of DSI 30D is heavily right skewed with extreme outliers, indicating that a small portion of items have exceptionally high days of inventory.
- Transformation is required to normalize this variable for modelling.

Dead Spoiled Ratio:

- This feature shows a sharp peak near zero with a long right tail, suggesting that most products have minimal spoilage while a few are highly perishable.
- Power transformation is considered to reduce skewness.

Sell Through Rate:

- The Sell Through Rate is highly skewed with most values clustered at lower ranges.
- This suggests that many products sell slowly, requiring normalization to stabilize variance.

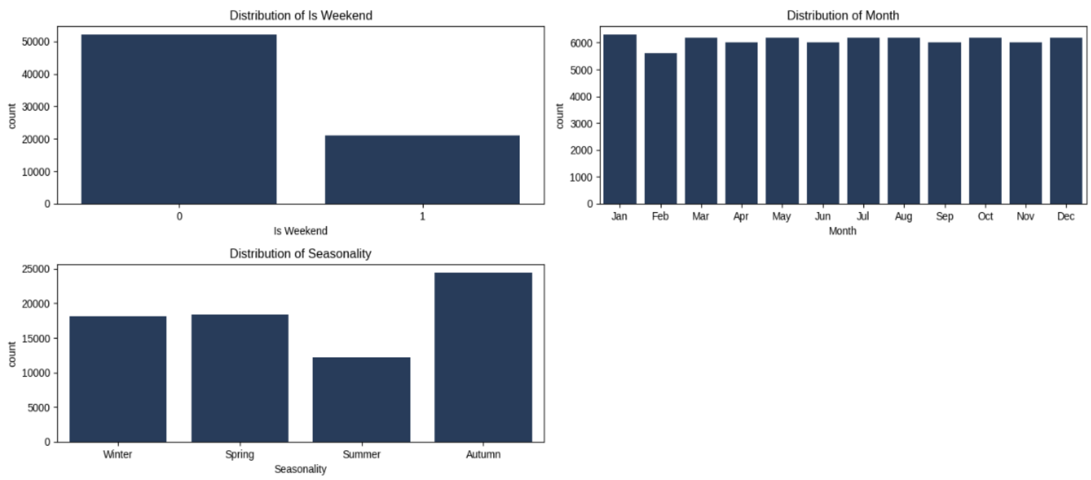
Demand-Supply Gap:

- The Demand-Supply Gap distribution is left skewed, indicating frequent undersupply relative to demand.
- This supports its relevance in predicting stockout risk.

Sales Velocity:

- Sales Velocity appears uniformly distributed across its range from 0 to 1.
- Scaling is not required for the feature.

4.2.4. Categoric Features – Univariate Analysis of Feature Engineered Columns



Is Weekend:

- Majority of the data points fall on weekdays, with fewer entries on weekends.
- This may influence consumer demand patterns and should be captured by the model.

Month:

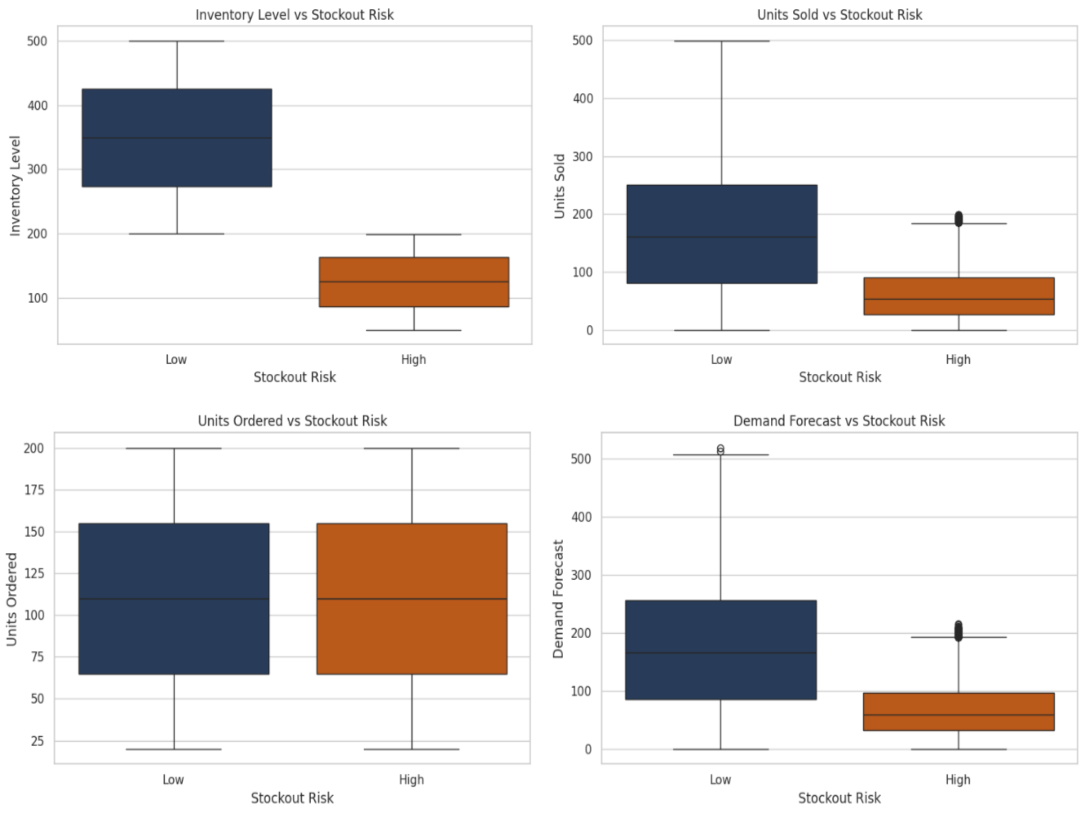
- Data is uniformly distributed across all 12 months, indicating no missing months or strong seasonal biases.
- This ensures temporal balance and supports seasonality analysis.

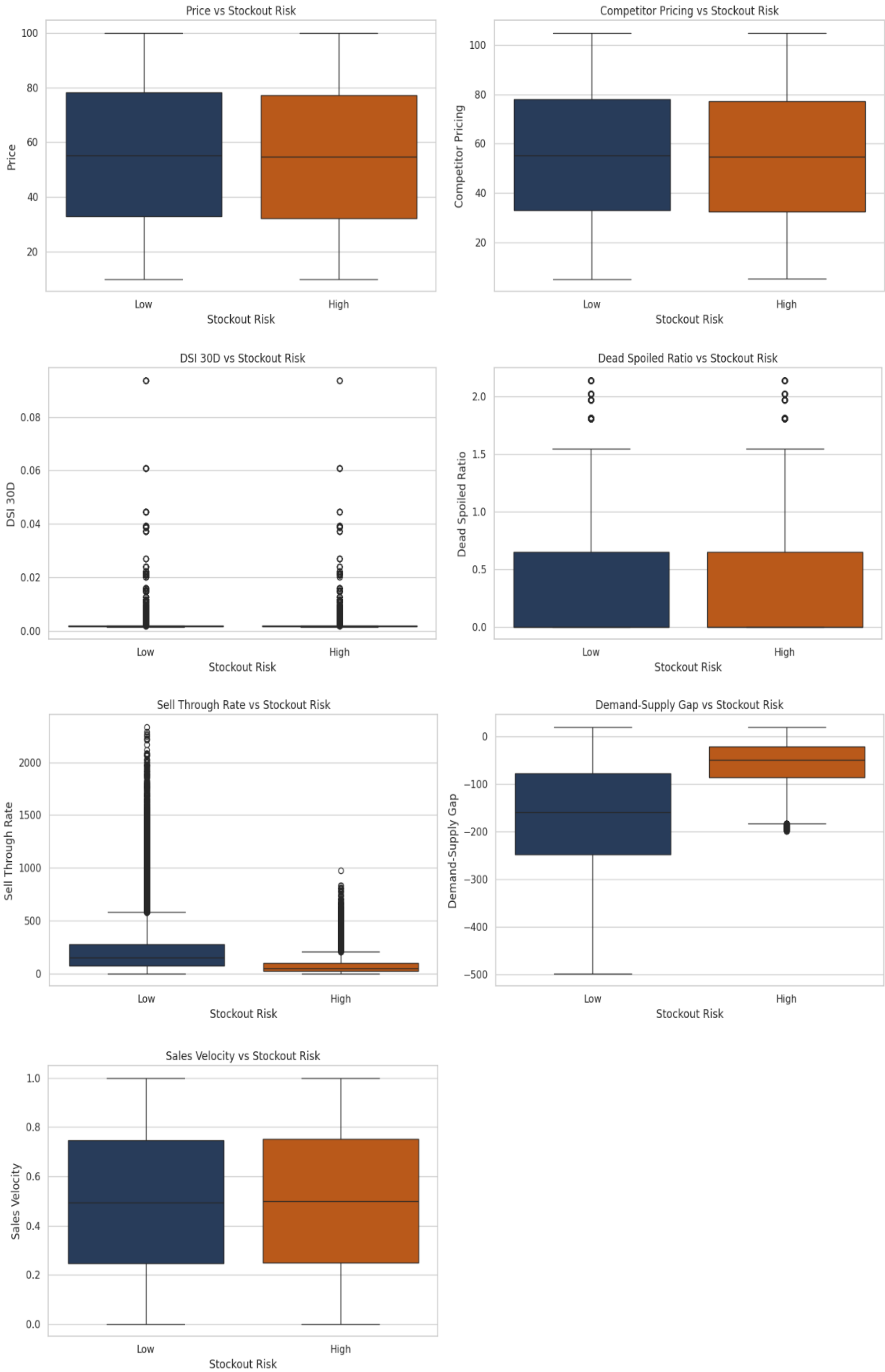
Seasonality:

- Autumn has the highest count while Summer is underrepresented.
- Seasonality differences might correlate with changes in demand, influencing stockout risk.

4.3. Relationship between variables

4.3.1. Numeric Features vs Stockout Risk – Bivariate Analysis





Inventory Level:

- There is a clear **inverse relationship** between inventory level and stockout risk, suggesting that maintaining higher inventory significantly reduces the risk of stockouts.

Units Sold:

- Products with **low stockout risk** tend to have **significantly higher units sold**, indicating better product availability drives more sales.
- Conversely, **high stockout risk** is linked to **lower sales volumes**, emphasizing the negative impact of frequent stockouts on revenue.

Units Ordered:

- Products with high stockout risk still receive comparable order volumes, indicating demand exists but supply may be constrained.
- This suggests stockouts are likely due to inventory issues rather than lack of customer interest.

Demand Forecast:

- Demand forecasted is significantly lower when stockout risk is high, suggesting inventory shortages may hinder meeting customer needs.
- Lower demand forecasts in high-risk scenarios may reflect conservative planning due to supply limitations.

Price:

- There is **no significant** price **difference** between high and low stockout risk categories, suggesting price is not a major factor driving stockouts.
- Stockout risk appears to be more influenced by inventory management than pricing strategy.

Competitor Pricing:

- Competitor pricing appears to have **no significant difference** between low and high stockout risk categories, with both showing similar distributions and median values.
- This suggests that **stockout risk is not strongly influenced by competitor pricing**, indicating other internal factors may play a more critical role.

DSI 30D:

- The boxplot shows that Days Sales of Inventory (DSI) over 30 days is generally **higher for low stockout risk** and lower for high stockout risk.
- This suggests that efficient inventory management helps maintain healthier stock levels, reducing volatility.

Dead Spoiled Ratio:

- The boxplot indicates that the Dead Spoiled Ratio is **higher when stockout risk is low**, suggesting better-stocked inventory may lead to more spoilage.
- Additionally, high stockout risk corresponds to significantly lower units sold, reinforcing the impact of inventory constraints on sales performance.

Sell Through Rate:

- The boxplot shows that Sell Through Rate is generally **higher for low stockout risk** and lower for high stockout risk.

- This suggests that maintaining sufficient inventory improves sell-through efficiency, reducing lost sales opportunities.

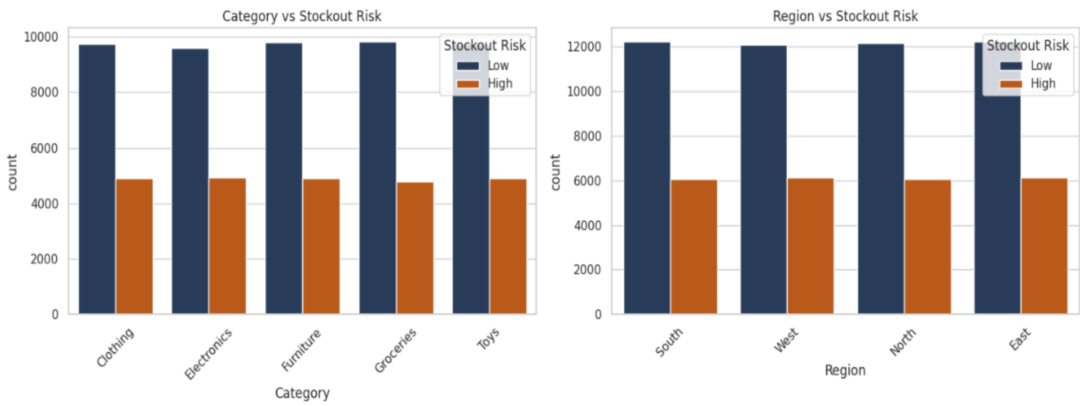
Demand Supply Gap:

- The boxplot illustrates the demand-supply gap across stockout risk levels, showing a **more negative gap when stockout risk is low**.
- This suggests that lower stockout risk is associated with oversupply, while high stockout risk indicates a constrained inventory unable to meet demand.

Sales Velocity:

- Sales Velocity is **almost the same** for both Low and High Stockout Risk groups.
- Sales Velocity may not be very useful for predicting stockout risk.

4.3.2. Categoric Features vs Stockout Risk – Bivariate Analysis

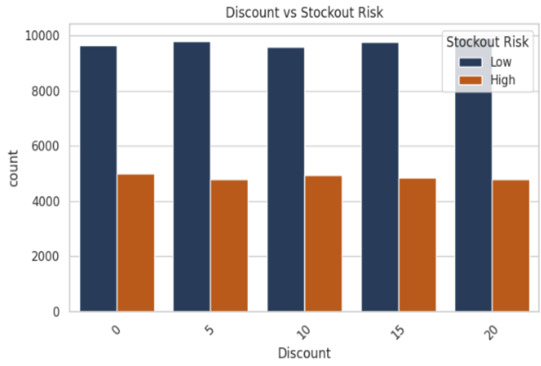


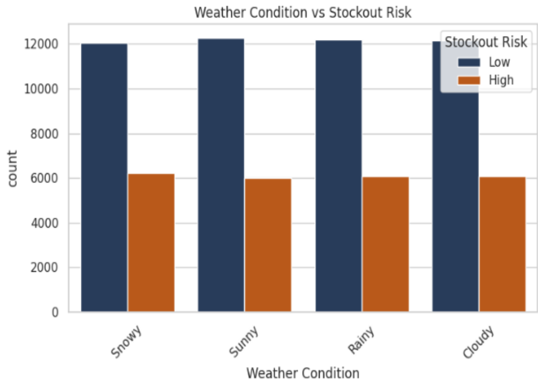
Category:

- Most categories have a **higher count of Low Stockout Risk** compared to High.
- The distribution is consistent across categories, so product category alone may not strongly influence stockout risk.

Region:

- All regions show a similar pattern, with a **greater number of Low Stockout Risk** than High.
- This suggests that region doesn't significantly impact stockout risk in the data.



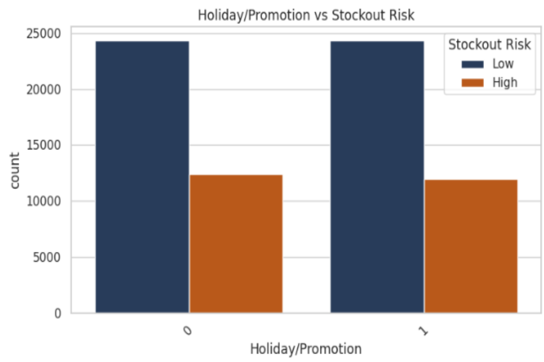


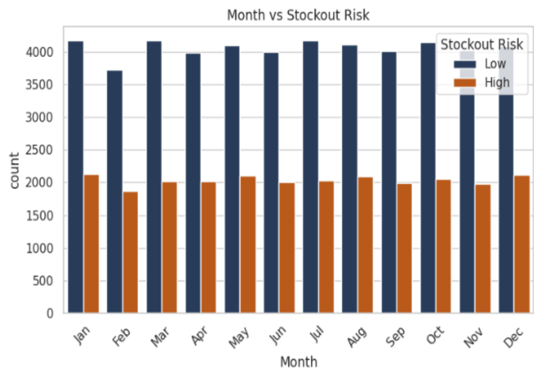
Discount:

- All the discount value shows **similar pattern**, so this suggest that the discount doesn't significantly shows any impact on stockout risk.

Weather Condition:

- All weather conditions show a **higher count of Low Stockout Risk** than High.
- This indicates that weather condition may not have a strong effect on stockout risk.



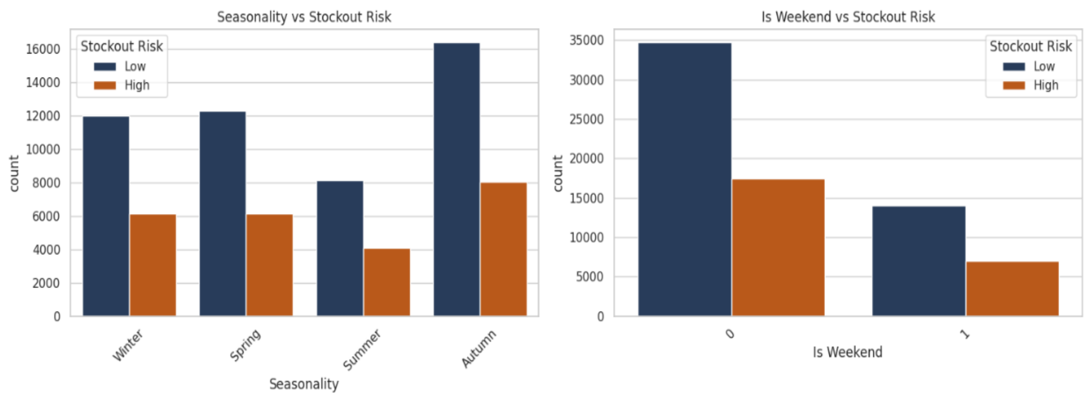


Holiday/Promotion:

- There is **not a dramatic difference** in stockout risk between holiday/promotion (1) and non-holiday periods (0).
- The count of high stockout risk is slightly lower during holiday/promotion periods.
- Both categories show similar patterns in overall distribution.

Month:

- The count of **low stockout risk** remains relatively **consistent** across all months.
- The **high stockout risk** count shows **slight variation** but stays generally stable, with **February showing the lowest** and **December showing the highest** risk counts.
- December has the **highest overall stockout risk**, which might correspond to **year-end holiday demand surges**.



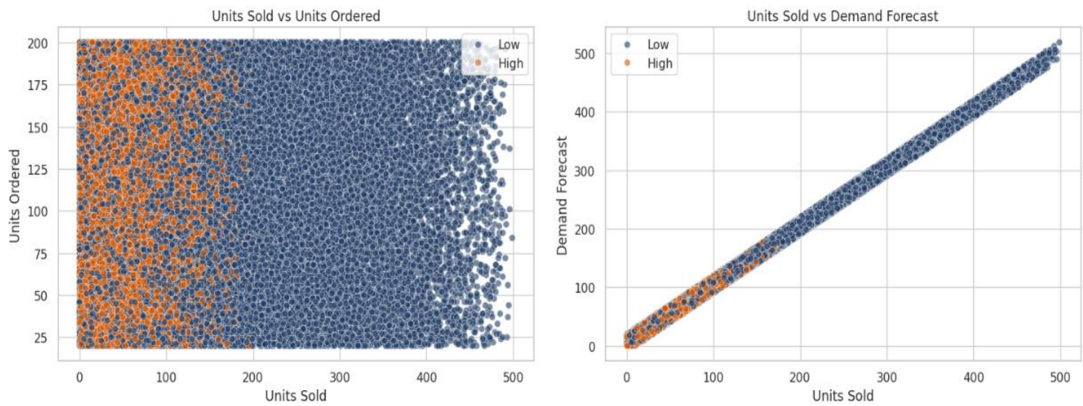
Seasonality:

- **Autumn** has the **highest count of both low and high stockout risk**, with a noticeable spike in overall stock volume.
- **Summer** has the **lowest count** for both risk categories, suggesting reduced activity or demand.
- **Winter and Spring** show **similar levels** of stockout risk, but Spring maintains a slightly higher low-risk count.

Is Weekend:

- **Weekdays (0)** see a significantly **higher number of both low and high stockout cases**.
- **Weekends (1)** have a **much lower count** for both categories, though the **ratio of high to low stockout** seems relatively consistent.

4.3.3. Numeric Features vs Numeric Features – Bivariate Analysis

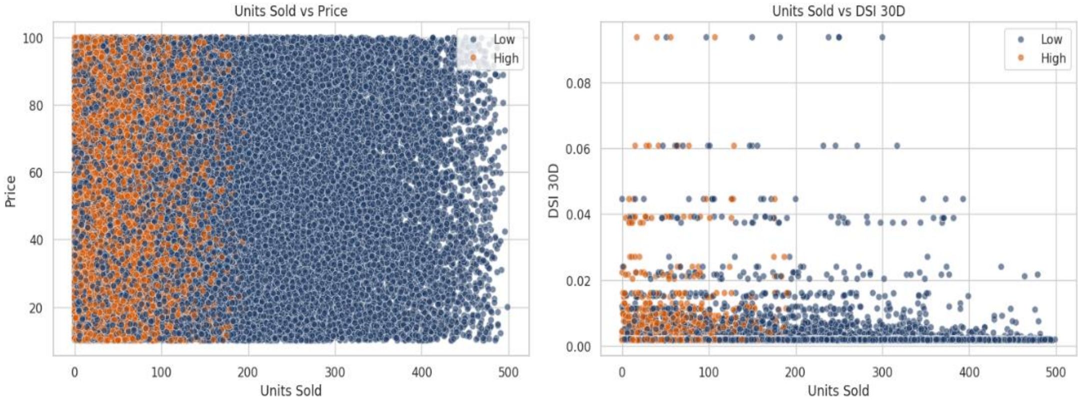


Units Sold vs Units Ordered:

- **High stockout risk** suggests cases where inventory **ordered fewer units than were sold**, leading to shortages.
- **Low stockout risk** demonstrates **better alignment**, meaning inventory orders generally meet demand.
- Inventory management for high-stockout-risk items may need **order optimization**, potentially factoring in lead times and real-time demand fluctuations.

Units Sold vs Demand Forecast:

- **High stockout risk** often occurs when **demand forecasts underestimate actual sales**, resulting in **insufficient ordering**
- **Low stockout risk** aligns well with forecasts, supporting a **reliable inventory replenishment strategy**.
- Fine-tuning demand forecasts—considering factors like promotions, seasonality, and unexpected spikes—could **help mitigate stockouts**.

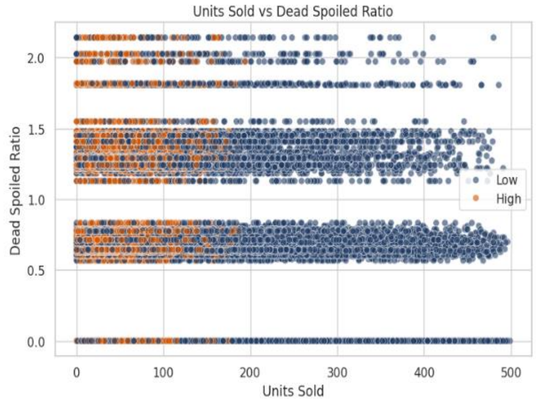


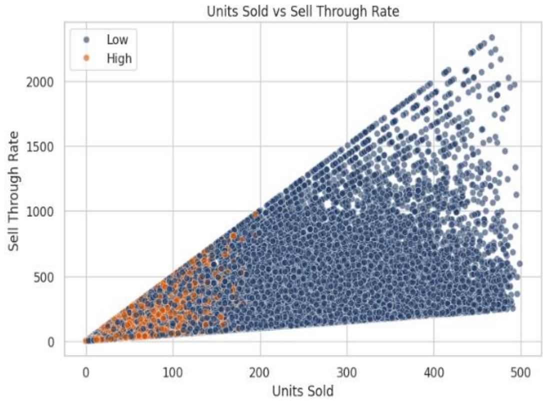
Units Sold vs Price:

- **High stockout risk** clusters toward **higher prices**, indicating that expensive items may have tighter inventory constraints, leading to shortages.
- **Low stockout risk** (blue dots) shows **a wider distribution**, suggesting more flexibility in pricing with better stock availability.
- Price-sensitive demand fluctuations could impact stockout risks, potentially requiring **dynamic pricing** or better inventory replenishment for high-risk items.

Units Sold vs DSI 30D (Days Sales of Inventory):

- **High stockout risk** tends to occur at **low DSI 30D values**, meaning inventory is turning over quickly, leading to **potential shortages**.
- **Low stockout risk** corresponds to **higher DSI 30D values**, suggesting a more balanced inventory flow where stock levels support demand more consistently.
- Adjusting safety stock levels based on rolling DSI trends could **help prevent sudden shortages** while maintaining healthy turnover rates.



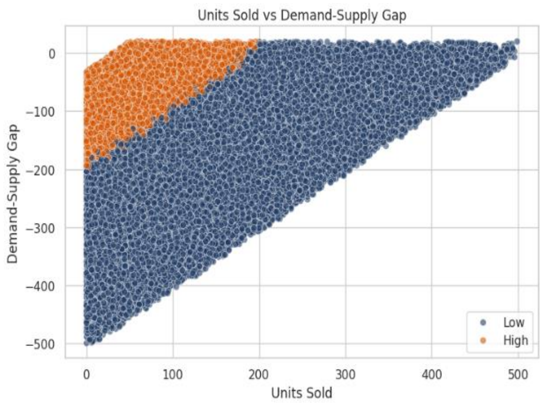


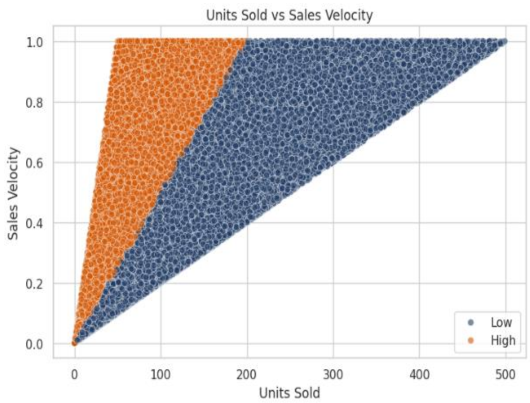
Units Sold vs Dead Spoiled Ratio:

- **High stockout risk** (orange dots) tends to have **lower Dead Spoiled Ratios**, indicating that when items are at risk of stockout, waste is relatively controlled.
- **Low stockout risk** (blue dots) shows more **variability in spoilage**, suggesting excess inventory may lead to higher waste levels.
- **Reducing unnecessary overstock** for low-stockout-risk items while improving replenishment for high-risk items could **optimize inventory usage** and minimize spoilage.

Units Sold vs Sell Through Rate:

- **High stockout risk** aligns with **higher Sell Through Rates**, meaning these items move quickly, often depleting inventory before replenishment.
- **Low stockout risk** has **more spread**, indicating balanced inventory levels that enable steadier sales performance.
- **Fine-tuning stock levels based on demand pace** can help **prevent stockouts** while keeping inventory efficient.



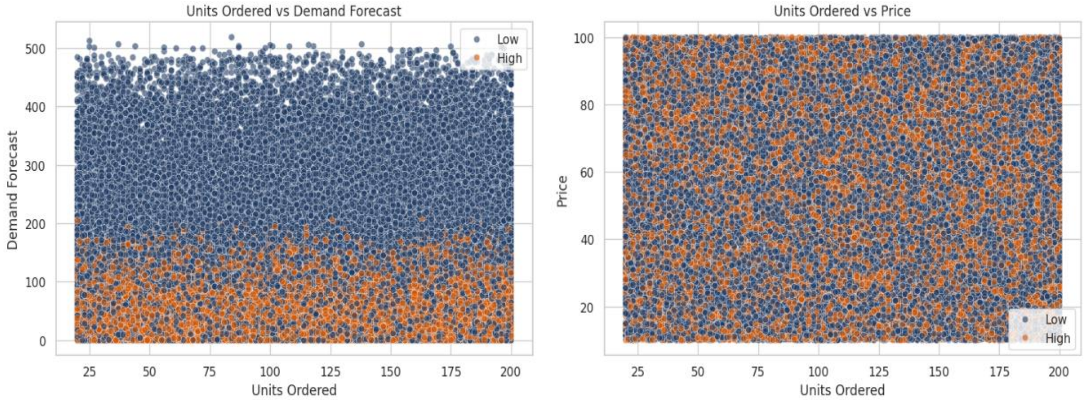


Units Sold vs Demand-Supply Gap:

- **High stockout risk** (orange dots) clusters in areas with **larger demand-supply gaps**, suggesting **inventory shortages** where demand exceeds supply.
- **Low stockout risk** (blue dots) is distributed across **smaller gaps**, indicating better alignment between supply and demand.
- Addressing stockout risk for high-risk items may require **better demand forecasting** and **dynamic replenishment strategies** to minimize supply mismatches.

Units Sold vs Sales Velocity:

- **High stockout risk** is concentrated in regions of **high sales velocity**, meaning these items sell quickly but may not be replenished efficiently
- **Low stockout risk** spans a wider range, suggesting steadier sales flows with **more controlled inventory management**.
- Ensuring **faster replenishment cycles** and **real-time demand tracking** for high-risk items could prevent sudden shortages.

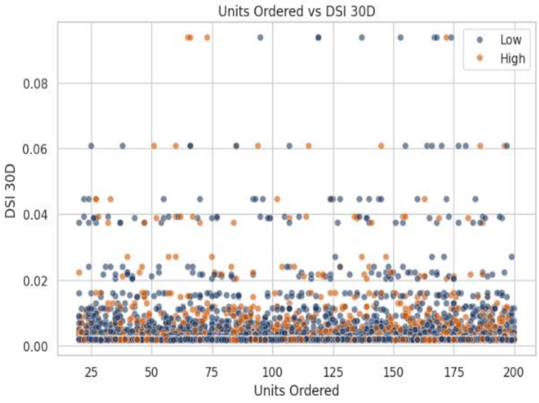


Units Ordered vs Demand Forecast:

- **High stockout risk** (orange dots) tends to cluster in regions where **units ordered are significantly lower than the forecasted demand**, indicating a risk of supply shortages.
- **Low stockout risk** (blue dots) shows better alignment between demand forecasts and ordering, suggesting a more balanced inventory replenishment approach.
- High-stockout-risk products may require **adjustments in reorder strategies** or **more dynamic replenishment models** to bridge demand gaps effectively.

Units Ordered vs Price:

- **High stockout risk** appears concentrated at **higher price points**, meaning expensive items might have tighter inventory constraints.
- **Low stockout risk** spans a broader price range, suggesting inventory strategies may be more optimized for different pricing tiers.
- Evaluating **price elasticity and replenishment cycles** could refine inventory management, ensuring higher-priced items don’t experience frequent stockouts.

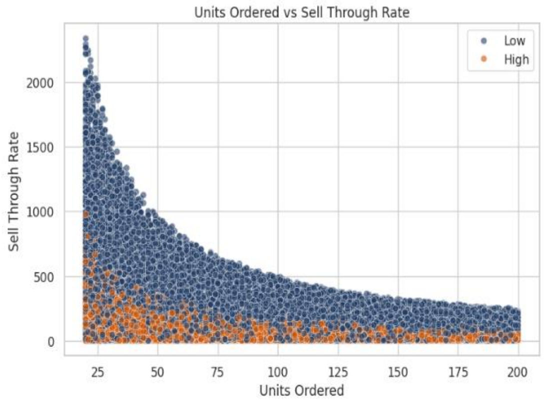


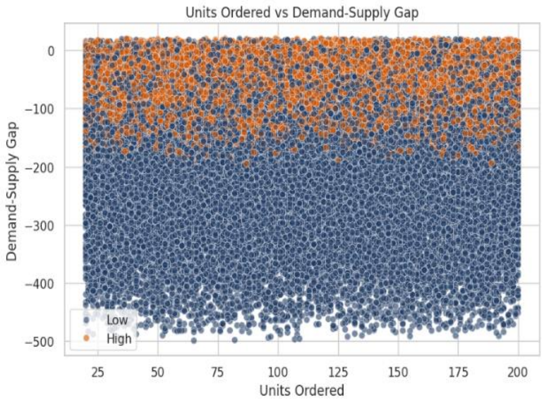
Units Ordered vs DSI 30D:

- **High stockout risk** (orange dots) is mostly **clustered at lower DSI 30D values**, indicating faster inventory depletion and limited stock coverage.
- **Low stockout risk** (blue dots) has a wider spread, suggesting **a more balanced inventory flow** with better stock management.
- Items with high stockout risk may need **adjustments in replenishment cycles**—especially for fast-moving products—to maintain adequate stock levels.

Units Ordered vs Dead Spoiled Ratio:

- **High stockout risk** tends to align with **lower dead spoiled ratios**, meaning that inventory is turning over quickly, reducing excess stock that could lead to waste.
- **Low stockout risk** shows **higher variability**, with some items accumulating spoilage due to slower movement.
- Optimizing **ordering strategies based on turnover patterns** could help minimize spoilage without increasing stockout risks.



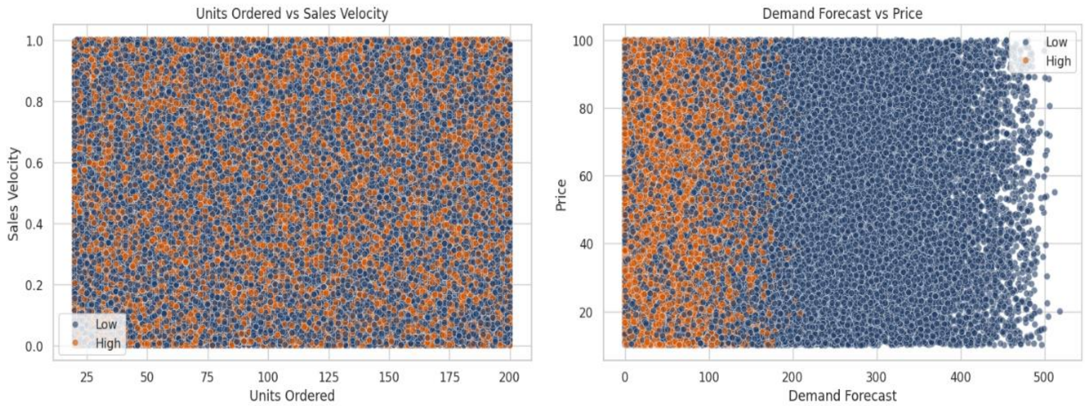


Units Ordered vs Sell Through Rate:

- **High stockout risk** shows more spread, indicating **lower sell-through rates** despite orders being placed.
- **Low stockout risk** (blue dots) clusters toward **higher sell-through rates**, meaning orders align more efficiently with demand.
- High-stockout-risk products may suffer from inventory misalignment, possibly requiring **smarter replenishment timing** or **real-time demand tracking**.

Units Ordered vs Demand Supply Gap:

- **High stockout risk** is associated with **smaller negative demand-supply gaps**, meaning the shortage is less severe but still persistent.
- **Low stockout risk** clusters at **larger negative demand-supply gaps**, showing better supply-demand alignment.
- Improving **reordering strategies** could further close demand-supply mismatches for high-risk items, ensuring smoother fulfilment.

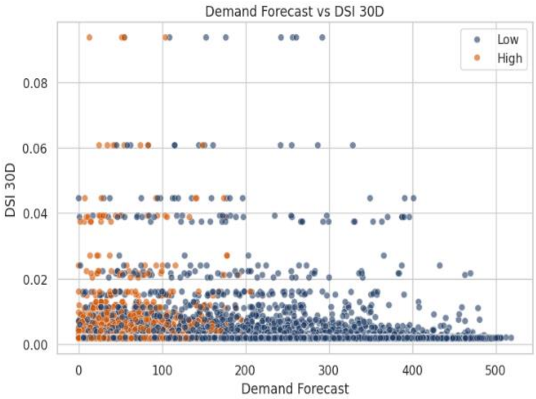


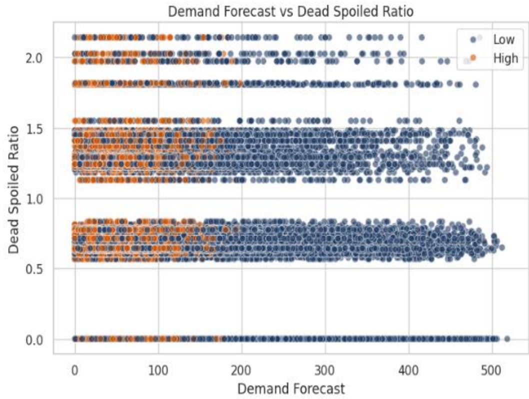
Units Ordered vs Sales Velocity:

- **High stockout risk** appears spread across various ordered quantities, with some clustering at **lower units ordered but higher sales velocity**—indicating items that sell rapidly but may not be replenished efficiently.
- **Low stockout risk** (blue dots) tends to align with **higher units ordered**, showing a more stable inventory turnover.
- Adjusting **reorder frequency and quantity** for high-stockout-risk items could improve availability while maintaining healthy turnover rates.

Units Ordered vs Price:

- **High stockout risk** tends to align with **moderate demand forecasts at higher price points**, suggesting that expensive products may experience **forecasting inconsistencies** leading to shortages.
- **Low stockout risk** spans a wider range of forecast values, suggesting **better predictability for price-sensitive demand**.
- Refining **forecast models for premium-priced items** could mitigate stockout risks by improving reorder decisions.



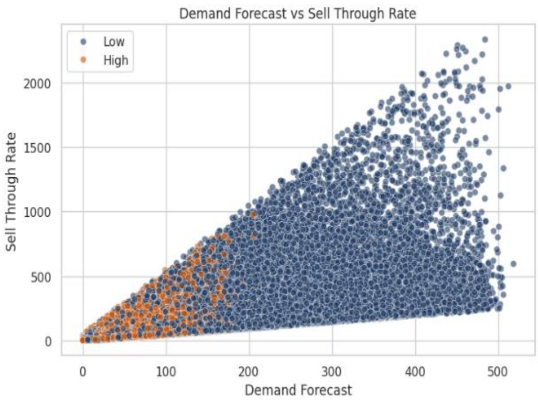


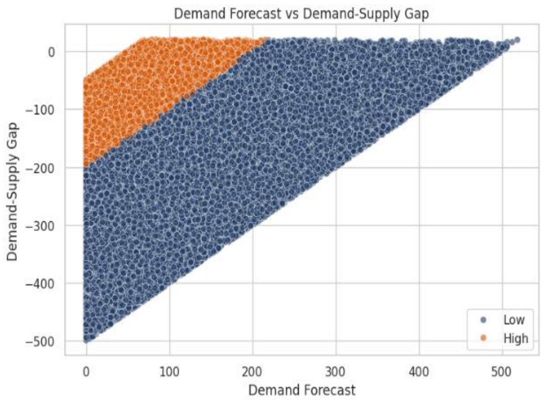
Demand Forecast vs DSI 30D:

- **High stockout risk** is mostly clustered in **low DSI 30D values**, indicating **fast inventory depletion**, leading to shortages.
- **Low stockout risk** (blue dots) appears more evenly spread across **higher DSI 30D values**, meaning inventory levels are better sustained.
- High-stockout-risk items may require **better reorder timing**, particularly those with rapid turnover, to prevent shortages.

Demand Forecast vs Dead Spoiled Ratio:

- **High stockout risk** aligns with **lower Dead Spoiled Ratios**, suggesting these items sell quickly, reducing excess stock accumulation.
- **Low stockout risk** exhibits **greater variation in spoilage**, indicating that slower-moving inventory may be more prone to wastage.
- **Fine-tuning inventory replenishment** to match demand pacing can help **reduce waste while minimizing stockouts**.



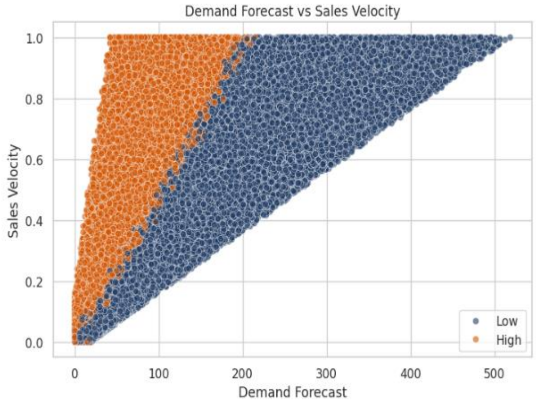


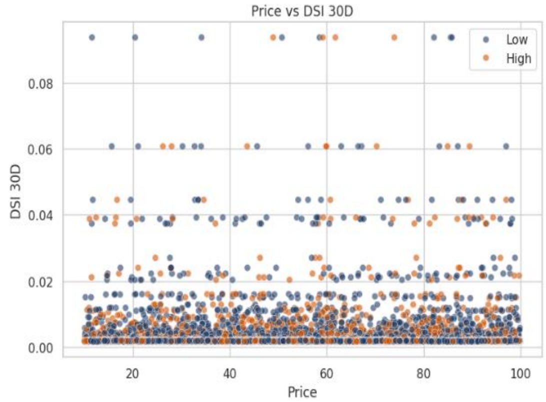
Demand Forecast vs Sell Through Rate:

- **High stockout risk** (orange dots) clusters in regions with **lower sell-through rates**, suggesting that demand forecasts may **underestimate replenishment needs**.
- **Low stockout risk** (blue dots) spans a **broader sell-through range**, indicating that inventory is adjusted more effectively based on forecasted demand.
- High-stockout-risk items might benefit from **dynamic inventory adjustments** to ensure demand forecasts accurately inform replenishment cycles.

Demand Forecast vs Demand-Supply Gap:

- **High stockout risk** aligns with **larger negative demand-supply gaps**, meaning supply is consistently falling short of forecasted demand.
- **Low stockout risk** appears more evenly distributed across **smaller gaps**, indicating better balance between inventory levels and expected demand.
- Fine-tuning **forecast models** and **real-time demand tracking** for high-risk items could minimize shortages by ensuring supply aligns more closely with demand.





Demand Forecast vs Sales Velocity:

- **High stockout risk** is concentrated in **low sales velocity** regions despite forecasted demand, indicating **inventory shortages may be limiting sales performance**.
- **Low stockout risk** is aligned with **higher sales velocity**, suggesting that inventory levels are effectively managed to meet demand.
- High-stockout-risk items may benefit from **demand-responsive replenishment cycles**, ensuring that forecasted demand translates efficiently into sales performance.

Price vs DSI 30D:

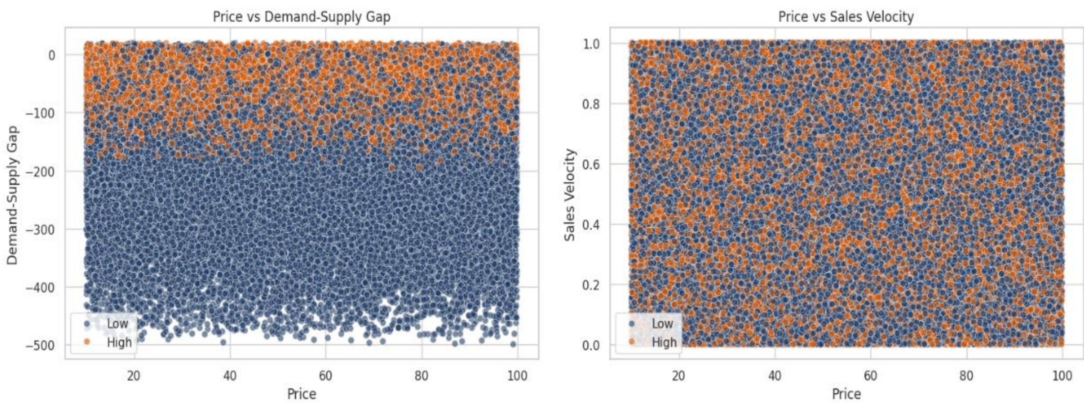
- **High stockout risk** tends to cluster in **lower DSI 30D ranges**, meaning these items **sell quickly, leading to frequent shortages**.
- **Low stockout risk** spans a wider range of **DSI 30D values**, suggesting better inventory coverage and stability across different price points.
- High-priced items with low stockout risk may need **inventory optimization strategies** to maintain efficient turnover rates while avoiding excessive overstocking.

Price vs Dead Spoiled Ratio:

- **High stockout risk** (orange dots) is mainly concentrated at **lower Dead Spoiled Ratios**, suggesting that stockout-prone items are more efficiently managed with minimal waste.
- **Low stockout risk** (blue dots) displays **greater variation in spoilage**, meaning excess inventory in some price ranges leads to higher spoilage.
- High-priced items with low spoilage suggest **better replenishment timing**, while lower-priced, low-risk items may need inventory adjustments to minimize waste.

Price vs Sell Through Rate:

- **High stockout risk** corresponds to **lower Sell Through Rates at higher prices**, indicating that expensive items might suffer demand fluctuations and replenishment delays.
- **Low stockout risk** spans a broader range with **higher sell-through rates**, suggesting better inventory alignment for price-sensitive demand.
- High-priced products might benefit from **demand forecasting refinements**, ensuring inventory meets expected sales velocity

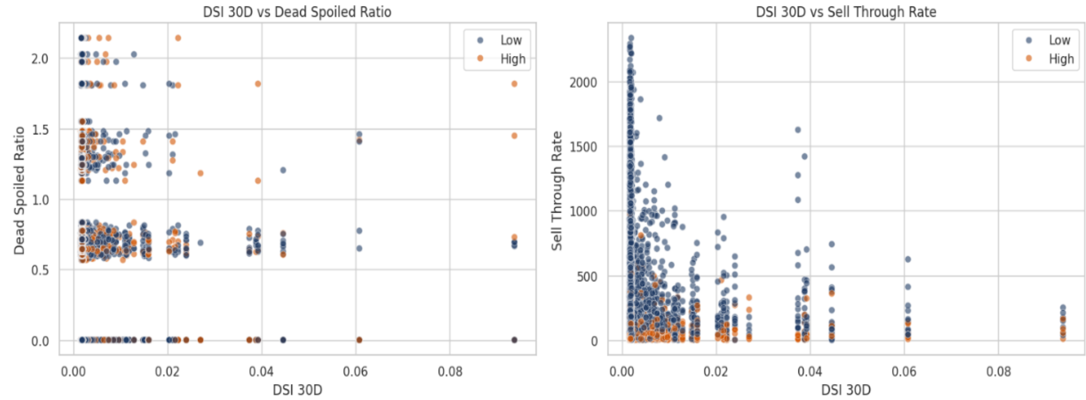


Price vs Demand-Supply Gap:

- **High stockout risk** (orange dots) is clustered around **larger negative demand-supply gaps**, indicating that supply frequently falls short of expected demand for certain price ranges.
- **Low stockout risk** (blue dots) is distributed across **smaller gaps**, showing better alignment between inventory levels and demand.
- High-priced items might suffer from **forecasting misalignment**, requiring **more responsive replenishment strategies** to minimize shortages.

Price vs Sales Velocity:

- **High stockout risk** tends to align with **lower sales velocity**, meaning inventory shortages may be restricting product movement.
- **Low stockout risk** spans a **broader velocity range**, indicating more stable inventory turnover across different price points
- High-priced products may benefit from **demand forecasting refinements**, ensuring that inventory levels support expected sales velocity.

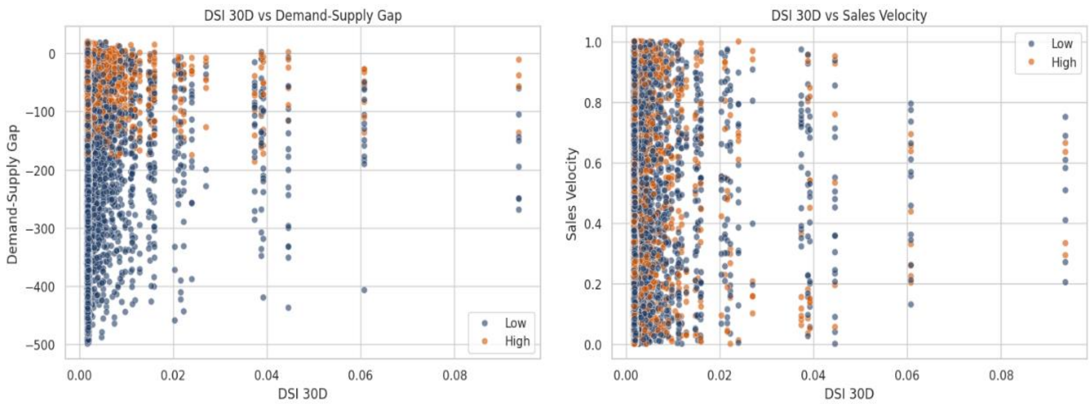


DSI 30D vs Dead Spoiled Ratio:

- **High stockout risk** clusters at **lower Dead Spoiled Ratios**, suggesting that inventory in this group is managed with **minimal spoilage** due to rapid turnover.
- **Low stockout risk** exhibits **greater spoilage variability**, indicating that excess inventory accumulation may lead to higher waste.
- High-stockout-risk items seem to have better inventory efficiency, while **low-stockout-risk items may need optimized ordering to reduce waste**.

DSI 30D vs Sell Through Rate:

- **High stockout risk** aligns with **higher Sell Through Rates**, meaning these items turn over quickly but risk frequent shortages.
- **Low stockout risk** covers **a broader range of sell-through rates**, suggesting more controlled inventory movement.
- Improving **reorder cycles for high-stockout-risk items** could **maintain availability while sustaining efficient turnover rates**.

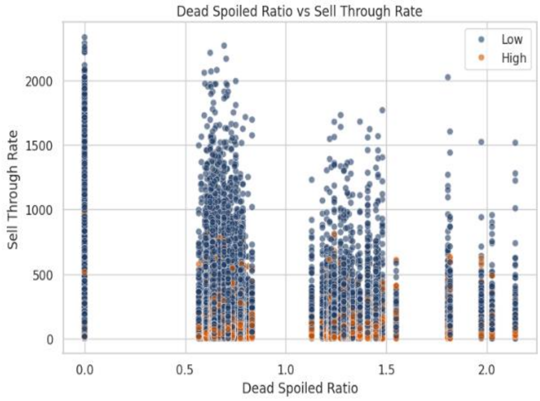


DSI 30D vs Demand-Supply Gap:

- **High stockout risk** (orange dots) clusters in regions with **larger negative demand-supply gaps**, meaning supply is frequently unable to meet demand.
- **Low stockout risk** (blue dots) is spread across **smaller gaps**, suggesting better balance between inventory replenishment and demand fluctuations.
- High-stockout-risk items may require **dynamic replenishment strategies**, ensuring that demand forecasts accurately drive supply decisions to minimize shortages.

DSI 30D vs Sales Velocity:

- **High stockout risk** tends to align with **higher sales velocity**, indicating that these items **sell quickly** but risk frequent shortages due to limited stock buffers
- **Low stockout risk** spans a **broader range of sales velocity**, reflecting better inventory stability and replenishment planning.
- Adjusting **reorder points and safety stock levels** could help prevent stockouts for fast-moving items while maintaining efficient turnover rates.

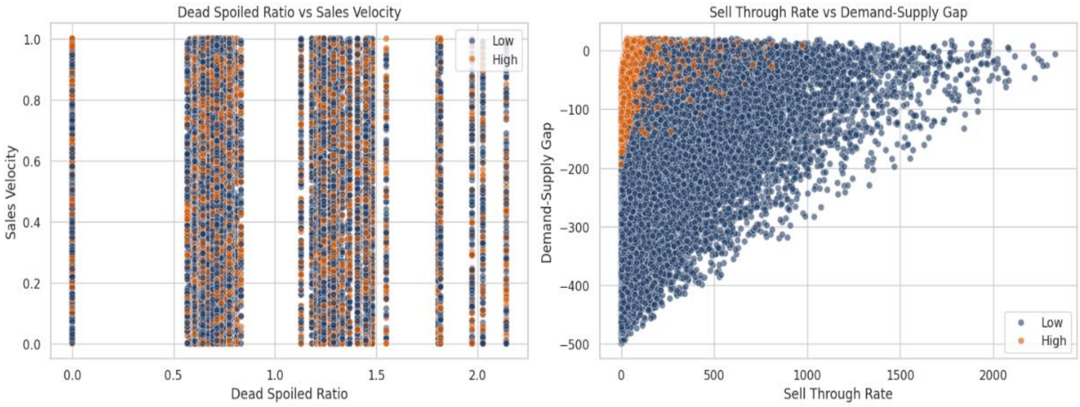


Dead Spoiled Ratio vs Sell Through Rate:

- **High stockout risk** (orange dots) clusters around **lower Dead Spoiled Ratios**, suggesting that these items are selling efficiently without excessive waste.
- **Low stockout risk** (blue dots) displays **higher Dead Spoiled Ratios in some cases**, indicating that inventory may sometimes exceed demand, leading to spoilage.
- High-stockout-risk items are **efficiently turning over**, but **low-stockout-risk items might need better stock monitoring** to minimize waste.

Dead Spoiled Rate vs Demand-Supply Gap:

- **High stockout risk** aligns with **larger negative demand-supply gaps**, meaning these products are frequently understocked relative to demand.
- **Low stockout risk** appears spread across **smaller gaps**, showing better inventory alignment with demand trends.
- Refining **demand-driven replenishment** could ensure better inventory control while avoiding spoilage and stockouts.

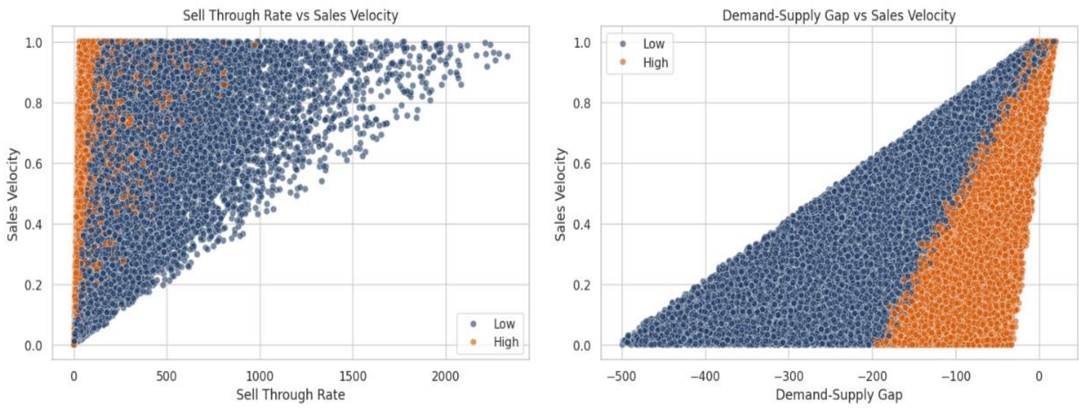


Dead Spoiled Ratio vs Sales Velocity:

- **High stockout risk** clusters at **lower Dead Spoiled Ratios**, indicating efficient turnover with minimal waste
- **Low stockout risk** shows a **broader spread**, meaning excess inventory might lead to higher spoilage rates.
- High-stockout-risk items appear well-optimized for turnover, but low-stockout-risk items could benefit from refined replenishment to reduce waste.

Sell Through Rate vs Demand-Supply Gap:

- **High stockout risk** aligns with **larger negative demand-supply gaps**, showing persistent inventory shortages in response to demand fluctuations.
- **Low stockout risk** spans a **smaller demand-supply gap range**, indicating better alignment between stock levels and demand pacing.
- Refining **real-time inventory tracking and adaptive replenishment strategies** could help prevent stockouts while minimizing surplus.



Sell Through Rate vs Sales Velocity:

- **High stockout risk** (orange dots) appears more scattered at **lower sales velocity**, indicating that despite sell-through rate variations, inventory shortages may be limiting movement.
- **Low stockout risk** (blue dots) aligns more closely with **higher sales velocity**, suggesting better stock availability enables smoother sales performance.

- For high-stockout-risk items, ensuring **timely replenishment cycles** could support **steady inventory turnover and sales flow**

Demand-Supply Gap vs Sales Velocity:

- **High stockout risk** tends to cluster in **larger negative demand-supply gaps**, meaning demand consistently outpaces available supply.
- **Low stockout risk** spans **smaller gaps**, suggesting better inventory balance with demand fluctuations
- Implementing **adaptive stocking strategies**, possibly informed by real-time demand patterns, could mitigate stockout risks while optimizing replenishment.

4.4. Multi-collinearity

4.4.1 Introduction

Multicollinearity refers to a situation where two or more independent variables in a dataset are highly correlated, potentially distorting the effect of each variable in a predictive model. Detecting and addressing multicollinearity is important to ensure model stability, interpretability, and reliable coefficient estimates.

4.4.2. Multivariate Analysis

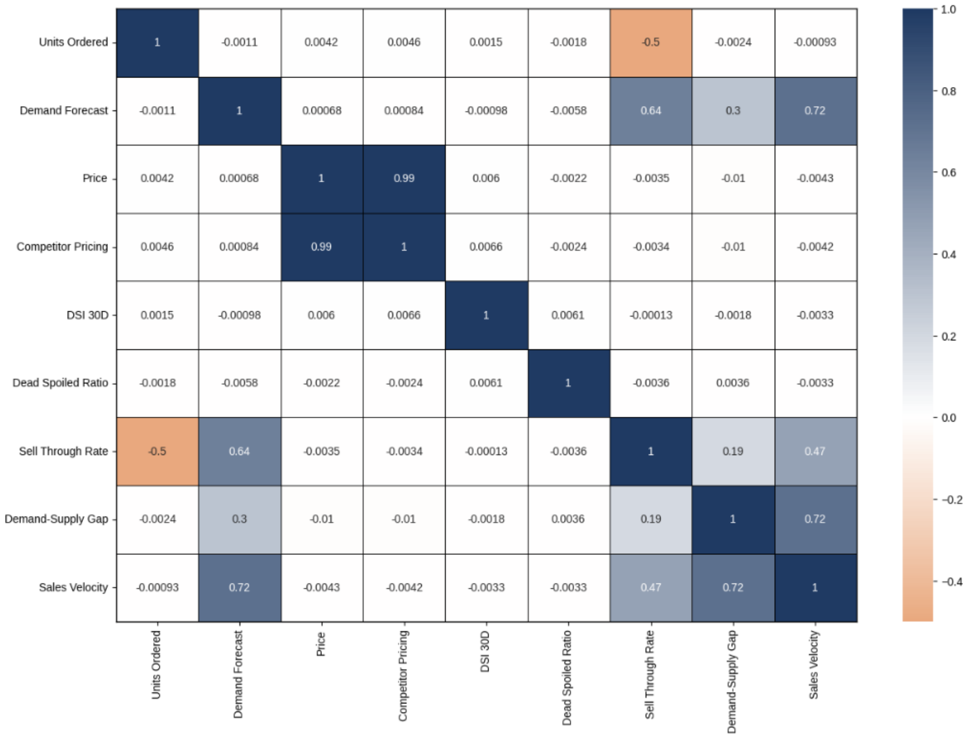


Fig: heatmap of feature correlations

- **Competitor Pricing & Price (0.99 correlation)** show redundancy, one can be removed to prevent multicollinearity issues.

- **Demand Forecast (0.64 correlation with Sell Through Rate)** suggests that future demand plays a crucial role in stock management.
- **Units Ordered (-0.5 correlation with Sell Through Rate)** indicates that higher order quantities don't necessarily reduce stockouts.

4.4.3. Variance Inflation Factor (VIF)

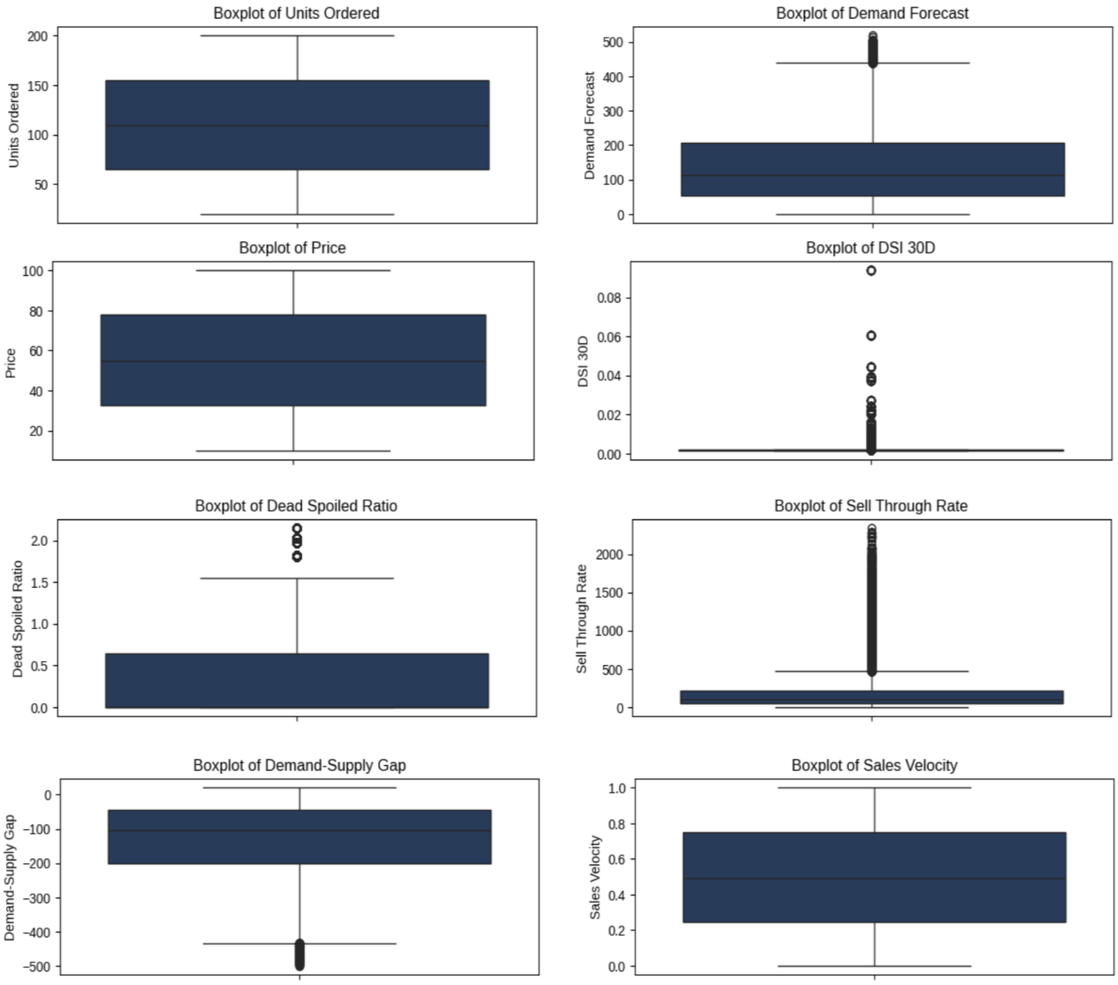
- To assess multicollinearity among independent variables, we computed the **Variance Inflation Factor (VIF)** for each feature.
- Variables exhibiting a VIF value greater than the commonly accepted threshold (typically > 5) were considered to introduce multicollinearity.

	Feature	VIF
0	Competitor Pricing	82.223216
1	Price	82.221978
2	const	32.828288
3	Sales Velocity	5.181832
4	Demand Forecast	3.892487
5	Sell Through Rate	2.986866
6	Demand-Supply Gap	2.715637
7	Units Ordered	1.745752
8	Dead Spoiled Ratio	1.000139
9	DSI 30D	1.000128

Table: VIF of features

- The **high VIF values** for features like **Price** and **Competitor Pricing** indicate strong multicollinearity, suggesting that these variables share redundant information, which can distort model coefficients.
- One of these features can be removed to enhance model robustness.

4.5. Presence of outliers and its treatment



- **Demand Forecast**, **DSI 30D**, **Dead Spoiled Ratio**, **Sell Through Rate**, **Demand-Supply Gap** are having outliers.
- Outliers are treated by applying suitable transformation:
  - **Log Transformation**: DSI 30D (skewness of 21.272), Sell Through Rate (skewness of 3.107)
  - **Square Root Transformation**: Demand Forecast (skewness of 0.897), Dead Spoiled Ratio (skewness of 1.349)
  - **Power Transformation (Box-Cox or Yeo-Johnson)**: Demand-Supply Gap (skewness: -0.884)

4.6. Statistical significance of variables

	Feature	Test	P-value	Statistically Significant
0	Demand Forecast	Mann-Whitney U	0.000000	True
1	Demand-Supply Gap	Mann-Whitney U	0.000000	True
2	Sell Through Rate	Mann-Whitney U	0.000000	True
3	Price	Mann-Whitney U	0.003272	True
4	Discount	Chi-square	0.017735	True
5	Holiday/Promotion	Chi-square	0.056179	False
6	Sales Velocity	Mann-Whitney U	0.110650	False
7	Weather Condition	Chi-square	0.130931	False
8	Dead Spoiled Ratio	Mann-Whitney U	0.133447	False
9	Seasonality	Chi-square	0.358216	False
10	Category	Chi-square	0.363235	False
11	DSI 30D	Mann-Whitney U	0.498650	False
12	Region	Chi-square	0.575236	False
13	Is Weekend	Chi-square	0.720389	False
14	Month	Chi-square	0.781167	False
15	Units Ordered	Mann-Whitney U	0.877862	False

Table: Statistical significance of features

- **Demand Forecast**, **Demand-Supply Gap**, **Sell Through Rate**, and Price are statistically **significant** predictors of stockout risk (p-values < 0.01, Mann-Whitney U test).
- **Discount** is also statistically **significant** (p = 0.0177, Chi-square test), indicating its impact on sales performance.
- Other features tested, including Holiday/Promotion, Sales Velocity, Weather Condition, and Category, were not statistically significant based on their respective tests.

4.7. Class imbalance and its treatment

	Stockout Risk	Proportion
0	Low	0.666183
1	High	0.333817

Table: Class distribution of target variable

- Target variable (Stockout Risk) exhibited noticeable class imbalance.
- To address this, we apply **Synthetic Minority Oversampling Technique (SMOTE)** to augment the minority class.
- This helped balance the class distribution and is expected to enhance model performance, particularly with respect to recall and precision for the minority class.

4.8. Feature Engineering

4.8.1. Transformation

- Transformation is done to reduce skewness (**refer section 4.5**).

4.8.2. Scaling

- All numerical features are to be scaled by applying standardization (using **StandardScaler()**).

4.8.3. Feature selection:

- Removed features with high VIF to prevent redundancy and instability during model building.

5. ASSUMPTIONS

5.1 Introduction

Before applying machine learning models, certain assumptions about the data and modeling process were considered. These assumptions help ensure the validity of the results and include aspects such as feature independence, absence of multicollinearity, and consistent data distribution. Addressing these assumptions enhances the robustness and reliability of the predictive models.

5.2. Classification Models

- **Logistic Regression**
  - Linearity of Logits: Assessed using the Box-Tidwell test; non-linear features were transformed if needed.
  - Multicollinearity: Checked using VIF.
  - No extreme outliers: Outliers were detected and transformed.
  - Class imbalance: Handled using SMOTE.
- **Decision Tree / Random Forest**
  - No strict assumptions on feature distribution or linearity.
  - Outliers and scaling are not sensitive in tree-based models.
  - Handled class imbalance with SMOTE or class weight parameter.
- **Bagging & Boosting Models (e.g., Random Forest, XGBoost)**
  - No specific assumptions regarding normality or linearity.
  - Outliers and missing values handled during preprocessing.
  - Feature scaling is generally not required but was applied uniformly for consistency.

6. BASELINE MODEL

```
Classification report of train data is:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93     17024
           1       0.97      0.97      0.97     34146

    accuracy                           0.96     51170
   macro avg       0.95      0.95      0.95     51170
weighted avg       0.96      0.96      0.96     51170


Classification report of test data is:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      7378
           1       0.96      0.97      0.96     14552

    accuracy                           0.95     21930
   macro avg       0.95      0.95      0.95     21930
weighted avg       0.95      0.95      0.95     21930
```

Fig: classification reports of train and test sets

- The baseline model achieved accuracies of 96% on train data and 95% on test data
- These high accuracy values suggest that the models are performing well on the training data, but the possibility of overfitting should be considered, especially if the models are not generalizing well to unseen data
- Regularization or cross-validation techniques will be explored to improve the model's generalization to unseen data

7. ALTERNATIVE APPROACHES TO STOCKOUT RISK CLASSIFICATION

7.1. Introduction

To improve the accuracy and reliability of the stockout risk prediction model, we considered several alternatives for refining the parameters and thresholds. These changes aimed to enhance the model’s ability to forecast stockouts more effectively by incorporating additional factors and adjusting key metrics.

7.2. Alternatives

- **Alternative 1: Adjust Inventory Level Threshold**
  Modify the inventory level threshold to the **25th percentile (162 units)** to better capture stockout risks at a more precise inventory level.
- **Alternative 2: Focus on Inventory Level and Demand Forecast**
  Limit the stockout risk determination to **inventory level** and **demand forecast** only, optimizing for simplicity and efficiency.
- **Alternative 3: Expand Parameters for Stockout Risk**
  Change the model to use **inventory level**, **units sold**, and **demand forecast** as parameters, improving prediction accuracy.

7.3. Threshold Selection

Initially, a threshold inventory level of 200 units was considered to classify stockout risk ("Yes" if inventory < 200). However, after detailed data analysis and business consultation, this threshold was refined to 162 units based on the following reasons:

- **Data-Driven Optimization**: Exploratory Data Analysis (EDA) revealed that 162 better captured the inflection point where actual stockouts started to occur historically, improving the model's ability to identify true stockout risk cases.
- **Business Context Alignment**: The threshold 162 was aligned with the business's operational definitions of critical stock levels and safety stock policies, making predictions more actionable and relevant for inventory managers.
- **Better Model Performance**: Models trained on this clear binary classification showed more consistent and reliable predictive performance, with improved precision and recall balance compared to more complex definitions.
- **Simplicity and Interpretability**: The inventory-level-only threshold at 162 units is straightforward and easy to explain to business stakeholders and inventory managers, facilitating faster adoption.

7.4. Class Imbalance

Stockout Risk	Proportion
No (0)	0.751902
Yes (1)	0.248098

Table: Class distribution of target variable

To enhance clarity and avoid potential misinterpretation, the original labels **Low** and **High** for stockout risk were redefined as **No (0)** and **Yes (1)**, respectively. This adjustment improves interpretability, especially in the context of binary classification. As shown in the table, the dataset exhibits a noticeable class imbalance, with approximately 75% of the records labeled as "No (0)" and only 25% as "Yes (1)".

7.5. Rejected Alternatives and Their Limitations

- **Alternative 2: Focus on Inventory Level and Demand Forecast**

  Though incorporating demand forecast added predictive richness, combining it with inventory level introduced complexity in defining a clear risk threshold, complicating interpretation and operational use.

- **Alternative 3: Expand Parameters for Stockout Risk**

  Adding units sold data increased dimensionality and potential noise, requiring more sophisticated feature engineering and model tuning. This complexity did not translate into proportionate gains in prediction accuracy or business utility.

7.6. Why threshold choice matters?

- A too-high threshold (e.g., 200) might flag too many false positives, causing unnecessary stock replenishment and increased holding costs.
- A too-low threshold risks missing early warnings, leading to stockouts and lost sales.

- By selecting 162, the models provide more reliable predictions that balance business costs of overstocking and stockouts, supporting better decision-making.

8. FEATURE REFINEMENT AND OVERFITTING MITIGATION STRATEGY

Based on the performance metrics observed in **Section 6**, the model displayed signs of **overfitting**, with exceptionally high accuracy on training data compared to test data. To address this, we conducted a **feature evaluation** and made the following refinements:

- **Removed features** that were either:
    - Likely to cause **data leakage**, such as **Demand Forecast**.
    - **Redundant or weakly correlated** with the target variable, including **Sell Through Rate**, **Price**, **Dead Spoiled Ratio**, and **DSI 30D**.
- **Introduced a derived feature**:
    - **Relative Price Difference;** calculated as the **ratio of Price to Competitor Pricing**. This was added to capture competitive positioning and provide more meaningful pricing insight to the model.
- These steps were implemented to improve generalization, reduce noise, and enhance the model's predictive robustness.

The dataset included a set of features highly relevant for modeling stockout risk, such as **Units Ordered, Discount, Holiday/Promotion, Category, Region, Weather Condition, Seasonality, Month, Demand-Supply Gap**, and **Sales Velocity** — all of which contribute to understanding inventory movement and demand patterns.

9. MODEL DEVELOPMENT AND COMPARATIVE PERFORMANCE ANALYSIS

9.1. Introduction

In this section, we present the development and evaluation of various classification models to predict stockout risk. To address the class imbalance inherent in the dataset, SMOTE (Synthetic Minority Oversampling Technique) was applied during model training. Each model underwent hyperparameter tuning to optimize performance. We report detailed classification metrics -including precision, recall, F1-score, and ROC AUC - for both training and testing datasets, along with visualized confusion matrices. Finally, we conduct a comparative analysis across all models to identify the most effective approach for reliable and generalizable stockout risk prediction.

9.2. Model Evaluation Before and After SMOTE

9.2.1. Introduction

To address the class imbalance observed in the dataset, we evaluated model performance both **before** and **after applying SMOTE (Synthetic Minority Over-sampling Technique)**. The objective was to observe its effect on improving the minority class (stockout = "Yes") prediction. This section includes:

- Classification reports
- ROC AUC scores for train and test sets

9.2.2. Performance Before and after SMOTE

a. Logistic Regression

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     38485
           1       0.86      0.77      0.81     12685

    accuracy                           0.91     51170
   macro avg       0.89      0.86      0.88     51170
weighted avg       0.91      0.91      0.91     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     16479
           1       0.86      0.77      0.82      5451

    accuracy                           0.91     21930
   macro avg       0.90      0.87      0.88     21930
weighted avg       0.91      0.91      0.91     21930

ROC AUC Score (Train): 0.9662769737263579
ROC AUC Score (Test): 0.9682274919723771
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       0.94      0.91      0.92     38485
           1       0.91      0.95      0.93     38485

    accuracy                           0.93     76970
   macro avg       0.93      0.93      0.93     76970
weighted avg       0.93      0.93      0.93     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.97      0.91      0.94     16479
           1       0.77      0.93      0.84      5451

    accuracy                           0.91     21930
   macro avg       0.87      0.92      0.89     21930
weighted avg       0.92      0.91      0.92     21930

ROC AUC Score (Train): 0.9703546198388427
ROC AUC Score (Test): 0.9691380530909022
```

Fig: classification reports before and after SMOTE

b. Gaussian Naive Bayes

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.82      0.86      0.84     38485
           1       0.50      0.42      0.46     12685

    accuracy                           0.75     51170
   macro avg       0.66      0.64      0.65     51170
weighted avg       0.74      0.75      0.75     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.82      0.86      0.84     16479
           1       0.51      0.43      0.46      5451

    accuracy                           0.75     21930
   macro avg       0.66      0.64      0.65     21930
weighted avg       0.74      0.75      0.75     21930

ROC AUC Score (Train): 0.8174189812011283
ROC AUC Score (Test): 0.8165240998898005
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       0.84      0.67      0.75     38485
           1       0.73      0.87      0.79     38485

    accuracy                           0.77     76970
   macro avg       0.78      0.77      0.77     76970
weighted avg       0.78      0.77      0.77     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.67      0.78     16479
           1       0.46      0.85      0.60      5451

    accuracy                           0.71     21930
   macro avg       0.69      0.76      0.69     21930
weighted avg       0.81      0.71      0.73     21930

ROC AUC Score (Train): 0.8269479681834773
ROC AUC Score (Test): 0.8162121781852542
```

Fig: classification reports before and after SMOTE

c. Linear Support Vector Classifier

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94     38485
           1       0.87      0.73      0.79     12685

    accuracy                           0.91     51170
   macro avg       0.89      0.85      0.87     51170
weighted avg       0.90      0.91      0.90     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94     16479
           1       0.87      0.74      0.80      5451

    accuracy                           0.91     21930
   macro avg       0.89      0.85      0.87     21930
weighted avg       0.91      0.91      0.90     21930

ROC AUC Score (Train): 0.9646200309730655
ROC AUC Score (Test): 0.9667355579577279
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       0.95      0.90      0.92     38485
           1       0.90      0.95      0.93     38485

    accuracy                           0.92     76970
   macro avg       0.93      0.92      0.92     76970
weighted avg       0.93      0.92      0.92     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.97      0.90      0.94     16479
           1       0.76      0.93      0.84      5451

    accuracy                           0.91     21930
   macro avg       0.87      0.92      0.89     21930
weighted avg       0.92      0.91      0.91     21930

ROC AUC Score (Train): 0.9700418060560556
ROC AUC Score (Test): 0.9688143754593064
```

Fig: classification reports before and after SMOTE

d. Random Forest Classifier

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38485
           1       1.00      1.00      1.00     12685

    accuracy                           1.00     51170
   macro avg       1.00      1.00      1.00     51170
weighted avg       1.00      1.00      1.00     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96     16479
           1       0.92      0.80      0.85      5451

    accuracy                           0.93     21930
   macro avg       0.93      0.89      0.90     21930
weighted avg       0.93      0.93      0.93     21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.9820106985838305
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38485
           1       1.00      1.00      1.00     38485

    accuracy                           1.00     76970
   macro avg       1.00      1.00      1.00     76970
weighted avg       1.00      1.00      1.00     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95     16479
           1       0.86      0.85      0.86      5451

    accuracy                           0.93     21930
   macro avg       0.91      0.90      0.91     21930
weighted avg       0.93      0.93      0.93     21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.9809657013146901
```

Fig: classification reports before and after SMOTE

e. XGBoost Classifier

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98     38485
           1       0.95      0.91      0.93     12685

    accuracy                           0.97     51170
   macro avg       0.96      0.95      0.95     51170
weighted avg       0.97      0.97      0.97     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.95      0.96      0.96     16479
           1       0.89      0.86      0.87      5451

    accuracy                           0.94     21930
   macro avg       0.92      0.91      0.92     21930
weighted avg       0.94      0.94      0.94     21930

ROC AUC Score (Train): 0.9955931005066805
ROC AUC Score (Test): 0.9850107811091025
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     38485
           1       0.97      0.97      0.97     38485

    accuracy                           0.97     76970
   macro avg       0.97      0.97      0.97     76970
weighted avg       0.97      0.97      0.97     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     16479
           1       0.88      0.87      0.87      5451

    accuracy                           0.94     21930
   macro avg       0.92      0.91      0.91     21930
weighted avg       0.94      0.94      0.94     21930

ROC AUC Score (Train): 0.997324652775111
ROC AUC Score (Test): 0.985383174589911
```

Fig: classification reports before and after SMOTE

f. LightGBM Classifier

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     38485
           1       0.86      0.77      0.81     12685

    accuracy                           0.91     51170
   macro avg       0.89      0.86      0.88     51170
weighted avg       0.91      0.91      0.91     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     16479
           1       0.86      0.77      0.82      5451

    accuracy                           0.91     21930
   macro avg       0.90      0.87      0.88     21930
weighted avg       0.91      0.91      0.91     21930

ROC AUC Score (Train): 0.9662619055824903
ROC AUC Score (Test): 0.9682112162476174
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       0.97      0.96      0.96     38485
           1       0.96      0.97      0.96     38485

    accuracy                           0.96     76970
   macro avg       0.96      0.96      0.96     76970
weighted avg       0.96      0.96      0.96     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.96      0.95      0.96     16479
           1       0.86      0.89      0.88      5451

    accuracy                           0.94     21930
   macro avg       0.91      0.92      0.92     21930
weighted avg       0.94      0.94      0.94     21930

ROC AUC Score (Train): 0.9962848377963004
ROC AUC Score (Test): 0.9863298718251051
```

Fig: classification reports before and after SMOTE

g. Multi-Layer Perceptron (MLP) Classifier

Train Report Before SMOTE:

	precision	recall	f1-score	support
0	0.96	0.97	0.97	38485
1	0.92	0.89	0.90	12685
accuracy			0.95	51170
macro avg	0.94	0.93	0.94	51170
weighted avg	0.95	0.95	0.95	51170

Test Report Before SMOTE:

	precision	recall	f1-score	support
0	0.95	0.96	0.96	16479
1	0.87	0.86	0.86	5451
accuracy			0.93	21930
macro avg	0.91	0.91	0.91	21930
weighted avg	0.93	0.93	0.93	21930

ROC AUC Score (Train): 0.9913307843193186
ROC AUC Score (Test): 0.9829685784219805

Train Report After SMOTE:

	precision	recall	f1-score	support
0	0.97	0.95	0.96	38485
1	0.95	0.97	0.96	38485
accuracy			0.96	76970
macro avg	0.96	0.96	0.96	76970
weighted avg	0.96	0.96	0.96	76970

Test Report After SMOTE:

	precision	recall	f1-score	support
0	0.96	0.94	0.95	16479
1	0.83	0.89	0.86	5451
accuracy			0.93	21930
macro avg	0.90	0.91	0.90	21930
weighted avg	0.93	0.93	0.93	21930

ROC AUC Score (Train): 0.994989814378748
ROC AUC Score (Test): 0.9823134415366227

Fig: classification reports before and after SMOTE

h. Decision Tree Classifier

Train Report Before SMOTE:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	38485
1	1.00	1.00	1.00	12685
accuracy			1.00	51170
macro avg	1.00	1.00	1.00	51170
weighted avg	1.00	1.00	1.00	51170

Test Report Before SMOTE:

	precision	recall	f1-score	support
0	0.95	0.95	0.95	16479
1	0.84	0.84	0.84	5451
accuracy			0.92	21930
macro avg	0.89	0.89	0.89	21930
weighted avg	0.92	0.92	0.92	21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.8930656829360347

Train Report After SMOTE:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	38485
1	1.00	1.00	1.00	38485
accuracy			1.00	76970
macro avg	1.00	1.00	1.00	76970
weighted avg	1.00	1.00	1.00	76970

Test Report After SMOTE:

	precision	recall	f1-score	support
0	0.96	0.93	0.95	16479
1	0.81	0.88	0.84	5451
accuracy			0.92	21930
macro avg	0.88	0.91	0.89	21930
weighted avg	0.92	0.92	0.92	21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.9055718351766926

Fig: classification reports before and after SMOTE

i. AdaBoost Classifier

Train Report Before SMOTE:

	precision	recall	f1-score	support
0	0.93	0.98	0.95	38485
1	0.92	0.78	0.84	12685
accuracy			0.93	51170
macro avg	0.92	0.88	0.90	51170
weighted avg	0.93	0.93	0.93	51170

Test Report Before SMOTE:

	precision	recall	f1-score	support
0	0.93	0.98	0.95	16479
1	0.92	0.77	0.84	5451
accuracy			0.93	21930
macro avg	0.92	0.87	0.89	21930
weighted avg	0.93	0.93	0.92	21930

ROC AUC Score (Train): 0.9811321489634327
ROC AUC Score (Test): 0.9802729031592484

Train Report After SMOTE:

	precision	recall	f1-score	support
0	0.98	0.87	0.92	38485
1	0.88	0.98	0.93	38485
accuracy			0.92	76970
macro avg	0.93	0.92	0.92	76970
weighted avg	0.93	0.92	0.92	76970

Test Report After SMOTE:

	precision	recall	f1-score	support
0	0.99	0.87	0.92	16479
1	0.71	0.97	0.82	5451
accuracy			0.89	21930
macro avg	0.85	0.92	0.87	21930
weighted avg	0.92	0.89	0.90	21930

ROC AUC Score (Train): 0.9844469574871528
ROC AUC Score (Test): 0.9795976387018211

Fig: classification reports before and after SMOTE

j. K-Nearest Neighbors Classifier

Train Report Before SMOTE:

	precision	recall	f1-score	support
0	0.86	0.97	0.91	38485
1	0.84	0.53	0.65	12685
accuracy			0.86	51170
macro avg	0.85	0.75	0.78	51170
weighted avg	0.86	0.86	0.85	51170

Test Report Before SMOTE:

	precision	recall	f1-score	support
0	0.80	0.93	0.86	16479
1	0.61	0.31	0.41	5451
accuracy			0.78	21930
macro avg	0.70	0.62	0.64	21930
weighted avg	0.75	0.78	0.75	21930

ROC AUC Score (Train): 0.9193766395325024
ROC AUC Score (Test): 0.7649531467861417

Train Report After SMOTE:

	precision	recall	f1-score	support
0	0.99	0.84	0.91	38485
1	0.86	0.99	0.92	38485
accuracy			0.92	76970
macro avg	0.93	0.92	0.92	76970
weighted avg	0.93	0.92	0.92	76970

Test Report After SMOTE:

	precision	recall	f1-score	support
0	0.92	0.78	0.84	16479
1	0.54	0.79	0.64	5451
accuracy			0.78	21930
macro avg	0.73	0.79	0.74	21930
weighted avg	0.83	0.78	0.79	21930

ROC AUC Score (Train): 0.9896025155978746
ROC AUC Score (Test): 0.8590795705822576

Fig: classification reports before and after SMOTE

k. Extra Trees Classifier

```
Train Report Before SMOTE:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38485
           1       1.00      1.00      1.00     12685

    accuracy                           1.00     51170
   macro avg       1.00      1.00      1.00     51170
weighted avg       1.00      1.00      1.00     51170

Test Report Before SMOTE:
              precision    recall  f1-score   support

           0       0.89      0.98      0.93     16479
           1       0.92      0.62      0.74      5451

    accuracy                           0.89     21930
   macro avg       0.90      0.80      0.84     21930
weighted avg       0.90      0.89      0.89     21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.9654399178670376
```

```
Train Report After SMOTE:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     38485
           1       1.00      1.00      1.00     38485

    accuracy                           1.00     76970
   macro avg       1.00      1.00      1.00     76970
weighted avg       1.00      1.00      1.00     76970

Test Report After SMOTE:
              precision    recall  f1-score   support

           0       0.93      0.96      0.95     16479
           1       0.88      0.77      0.82      5451

    accuracy                           0.92     21930
   macro avg       0.90      0.87      0.88     21930
weighted avg       0.92      0.92      0.92     21930

ROC AUC Score (Train): 1.0
ROC AUC Score (Test): 0.9713628567187723
```

Fig: classification reports before and after SMOTE

9.2.3. Comparative Analysis of Model Performance

To address the class imbalance and improve the model's sensitivity to minority class predictions, SMOTE (Synthetic Minority Over-sampling Technique) was applied. This technique synthetically generated samples of the underrepresented class during training, resulting in a more balanced dataset.

Model	F1 (Before)	F1 (After)	Recall (Before)	Recall (After)	ROC AUC (Before)	ROC AUC (After)	Δ Recall	Overfitting Risk
Logistic Regression	0.82	0.84	0.77	0.93	0.97	0.97	0.16	No
Gaussian NB	0.46	0.60	0.43	0.85	0.82	0.82	0.42	No
Linear SVC	0.80	0.84	0.74	0.93	0.97	0.97	0.19	No
Random Forest	0.85	0.86	0.80	0.85	0.98	0.98	0.05	Yes
XGBoost	0.87	0.87	0.86	0.87	0.99	0.99	0.01	No
LightGBM	0.82	0.88	0.77	0.89	0.97	0.99	0.12	No
Multilayer Perceptron	0.86	0.86	0.86	0.89	0.98	0.98	0.03	Slightly
Decision Tree	0.84	0.84	0.84	0.88	0.89	0.91	0.04	Yes
AdaBoost	0.84	0.82	0.77	0.97	0.98	0.98	0.20	No
K-Nearest Neighbors	0.41	0.64	0.31	0.79	0.77	0.86	0.48	No
Extra Trees	0.74	0.82	0.62	0.77	0.97	0.97	0.15	Yes

Table: comparison of model performance metrics on test data before and after tuning

The comparative summary improves interpretability by clearly showing which models benefitted from SMOTE and by how much.

- **Significant improvements** were observed in models such as **Gaussian Naive Bayes**, **KNN**, and **AdaBoost**, indicating enhanced detection of stockout cases.

- Models like **XGBoost**, **MLP**, and **Random Forest** showed **minimal change**, as they already handled imbalance reasonably well.
- Some tree-based models (e.g., **Random Forest**, **Extra Trees**) exhibited **overfitting**, as reflected in perfect training scores but slightly weaker generalization.

Based on the improved recall and balanced performance observed across models, we proceed with SMOTE-applied data for hyperparameter tuning.

9.3. Hyperparameter Tuning

To enhance model performance and minimize bias-variance trade-offs, hyperparameter tuning was conducted on each classifier using randomized/grid search methods. The tuning was performed **after applying SMOTE**, ensuring that models learn from a balanced dataset.

Key parameters such as tree depth, learning rate, regularization strength, and kernel type were optimized based on **recall**, which is the primary metric for this stockout risk prediction problem. This is because **recall focuses on correctly identifying the positive class ("Yes" for stockout)** — a critical aspect in retail operations where **failing to predict a stockout can lead to lost sales, customer dissatisfaction, and inventory disruptions**. By maximizing recall, the model prioritizes minimizing false negatives, ensuring stockout risks are flagged more reliably. The table below summarizes the **best hyperparameters** chosen for each model.

Model	Best Hyperparameters
Logistic Regression	C: 0.01, penalty: l2, solver: liblinear
Gaussian Naive Bayes	Naive Bayes has no hyperparameters to tune
Random Forest Classifier	n_estimators: 200, min_samples_split: 2, max_depth: 10, class_weight: balanced
XGBoost Classifier	classifier__max_depth: 5, classifier__n_estimators: 100
Linear SVC	class_weight: balanced, C: 0.1
LightGBM Classifier	classifier__learning_rate: 0.05, classifier__max_depth: 5, classifier__n_estimators: 100
MLP Classifier	classifier__alpha: 0.0001, classifier__hidden_layer_sizes: (100, 50), classifier__learning_rate_init: 0.01
Decision Tree Classifier	class_weight: balanced, criterion: entropy, max_depth: 5, min_samples_split: 2
AdaBoost Classifier	learning_rate: 0.5, n_estimators: 50
KNN Classifier	classifier__n_neighbors: 3, classifier__p: 1, classifier__weights: distance
Extra Trees Classifier	class_weight: balanced, max_depth: 10, min_samples_split: 2, n_estimators: 100

Table: best hyperparameters for each classification model

The following table summarizes the performance metrics of all tuned classifiers, with a primary focus on recall for the minority class ("Yes"), alongside F1-score, accuracy, and ROC-AUC score.

Model	Train Recall (Yes)	Test Recall (Yes)	Train F1 (Yes)	Test F1 (Yes)	Train Accuracy	Test Accuracy	Train ROC AUC	Test ROC AUC
Logistic Regression	0.96	0.94	0.93	0.84	0.93	0.91	0.97	0.97
Gaussian NB	0.88	0.85	0.79	0.59	0.77	0.71	0.83	0.82
Linear SVC	0.95	0.93	0.93	0.84	0.92	0.91	0.97	0.97
Random Forest	0.99	0.97	0.92	0.78	0.92	0.86	0.99	0.97
XGBoost	0.96	0.87	0.96	0.88	0.96	0.94	0.99	0.99
LightGBM	0.97	0.94	0.95	0.86	0.95	0.92	0.99	0.99
MLP	0.99	0.86	0.99	0.85	0.99	0.92	0.99	0.98
Decision Tree	0.98	0.97	0.91	0.78	0.91	0.87	0.96	0.96
AdaBoost	0.99	0.98	0.94	0.83	0.93	0.90	0.99	0.98
KNN	1	0.55	1	0.53	1	0.76	1	0.76
Extra Trees	0.98	0.95	0.88	0.69	0.86	0.79	0.98	0.95

Table: performance metrics of models after hyperparameter tuning

Note: Gaussian Naive Bayes does not have tunable hyperparameters in its standard form and was used with its default configuration.

The tuned models generally demonstrate improved recall and stability on the test set. While some models like **KNN** and **Decision Tree** show signs of overfitting, models such as **XGBoost**, **MLP**, and **Logistic Regression** strike a better balance between performance and generalization, making them strong candidates for deployment.

10. BUSINESS INSIGHTS AND MODEL DEPLOYMENT RECOMMENDATIONS

10.1. Introduction

To bridge model evaluation with practical implementation, this section interprets the tuned model results through a business lens. The focus lies on identifying models that not only perform well statistically but also align with the operational goals of minimizing stockouts, ensuring inventory availability, and maintaining supply chain efficiency. Each model is assessed for its deployment feasibility, interpretability, and reliability in real-world scenarios. Since the business goal is to **accurately predict stockout risk** and **prevent missed inventory events**, the key focus is on **recall** for the **“Yes”** class (i.e., detecting actual stockouts).

10.2. Model-wise Business Insights

a. Logistic Regression

- High recall and balanced ROC AUC scores.
- Easy to interpret, fast to deploy, and low in resource consumption.
- Ideal for **baseline deployment** or environments requiring **explainability**.
- Best suited for business dashboards or rule-based decision systems.

b. Gaussian Naive Bayes

- Lower test recall and significant drop in F1-score.
- High false-negative rate makes it **unsuitable for stockout-sensitive operations**.
- Not recommended for deployment in this case.

c. Linear SVC

- Consistent recall and AUC scores across train and test.
- Reliable in **high-dimensional data** but lacks probability outputs (limiting risk scoring).
- May be considered when **speed and linear decision boundaries** are sufficient.

d. Random Forest

- Excellent recall on train set but noticeable drop in F1 on test set → **overfitting**.
- Good for initial pilots with frequent model retraining.
- Use only with **model monitoring in place**.

e. XGBoost

- Strong across all metrics, high AUC, stable test recall.
- Handles non-linear patterns and imbalance effectively.
- Highly recommended for **production-level deployment** in real-time stockout prediction.

f. LightGBM Classifier

- **Performs consistently** across key metrics, making it suitable for operational deployment in **supply chain systems** to anticipate stockouts with high reliability.
- Offers **fast computation and scalability**, ideal for **real-time inventory dashboards or automated alert systems** in large retail environments.
- Easily integrates into production pipelines and supports **periodic retraining**, enabling businesses to **adapt to changing demand patterns** with minimal overhead.

g. MLP (Neural Network)

- High recall and balanced generalization.
- Works well with large, complex datasets — suitable for **warehouse management platforms**.
- Slightly more difficult to interpret and tune over time.

h. Decision Tree

- Great recall but lower F1 and accuracy on test → **overfitting risk**.
- Suitable in **rule-based or transparent systems**, but not ideal for dynamic predictions.
- May be considered only with pruning and retraining strategies.

i. AdaBoost

- Stable AUC and accuracy, but modest F1 and slight overfit.
- Good candidate for **ensemble learning pipelines**.
- Reasonable option for systems combining interpretability and moderate risk tolerance.

j. KNN

- Severe overfitting, poor generalization on test.
- Not suitable for deployment; highly sensitive to noise and large datasets.
- Avoid for this use case.

k. Extra Trees

- Performs moderately, decent recall but relatively lower F1.
- May be used in **experimental models** or A/B testing, but not as a standalone solution.

11. FINAL MODEL SELECTION

After evaluating all models based on key performance metrics - with a primary focus on recall for the stockout class ("Yes"), as well as F1-score, accuracy, and ROC AUC **-** the **XGBoost Classifier** is selected as the **best overall model** for deployment because of:

- **High and Balanced Performance:**
  XGBoost achieved **0.87 recall**, **0.87 F1-score**, and **0.99 ROC AUC** on the test set - striking a strong balance between correctly identifying stockouts and **minimizing false positives**.
- **Robustness and Industry Maturity:**
  With a proven track record in production systems, XGBoost offers stability**,** flexibility, and excellent support for **handling imbalanced data**, which aligns well with stockout risk prediction.
- **Scalability and Fine Control:**
  Its ability to **handle large datasets** and provide granular control over tuning makes it ideal for supply chain environments where demand patterns evolve and model updates are routine.

12. STRATEGIC BUSINESS RECOMMENDATIONS TO MINIMIZE STOCKOUTS

12.1. Introduction

While predictive modeling significantly enhances a company's ability to anticipate stockouts, its true impact is realized only when combined with robust inventory and supply chain strategies. Below are key operational recommendations categorized into practices to avoid and those to adopt.

12.2. What NOT to do

- **Rely solely on historical average demand**
  Using only past averages fails to account for variability, seasonality, and demand spikes — leading to understocking during high-demand periods.
- **Use static reorder points**
  Fixed thresholds don't adapt to real-time changes in inventory levels or evolving sales trends, increasing the likelihood of misalignment.
- **Ignore supplier lead time variability**
  Not accounting for unpredictable supplier delays can cause stock to run out before replenishment arrives.
- **Neglect inventory visibility**
  Lack of real-time tracking or poor system integration leads to inaccurate stock data, compromising decision-making and forecasting.

12.3. What TO do

- **Adopt dynamic, data-driven reorder points**
  Leverage predictive models and real-time data to adjust reorder levels based on current inventory, demand patterns, and seasonality.
- **Maintain safety stock buffers**
  Hold calculated buffer stock based on demand and supply variability to reduce the risk of unexpected stockouts.
- **Enhance supplier collaboration**
  Establish clear communication and performance metrics with suppliers to minimize delivery uncertainties.
- **Integrate inventory systems**
  Ensure synchronization between procurement, sales, and warehouse systems for seamless and accurate inventory control.
- **Continuously monitor and update ML models**
  Regularly retrain stockout prediction models using fresh data to reflect market shifts and changing customer behaviour.

13. CONCLUSION

This project aimed to tackle the issue of stockouts in the retail supply chain by building a predictive model using historical inventory and sales data. Through a structured approach that included exploratory data analysis, data preprocessing, class balancing with SMOTE, and model tuning, we were able to develop accurate and reliable classification models.

Among all the models evaluated, **XGBoost** emerged as the most effective, demonstrating strong recall and overall performance, making it well-suited for identifying potential stockouts. To support practical implementation, additional business strategies were recommended to work alongside the predictive model, ensuring better inventory management and supply chain efficiency.

By combining machine learning with operational insights, this solution provides a scalable framework that retailers can adopt to proactively reduce stockouts, enhance customer satisfaction, and make more informed inventory decisions.