MGMT 59000 CFA Spring 2025

Muskan Aggarwal

Deepak Saini

Rupali Kakadia

Shashank Sridhar

Dataset Description

For this project, we utilized the "Global Product Inventory Dataset 2025" dataset sourced from Kaggle. This publicly available dataset simulates a structured catalog of consumer goods typically found in online marketplaces and retail platforms. The dataset contains detailed product-level information sourced from a hypothetical global supplier, designed to support projects in logistics optimization, inventory planning, and supply chain analytics. Each row represents a unique product and includes metadata relevant to pricing, physical attributes, stock levels, and customer feedback.

The dataset includes over 10,000 product listings across diverse categories such as Electronics, Clothing, Home Appliances, and more. Each record contains detailed product-level metadata, including:

- Product ID: Unique identifier for each item
- Product Name: Textual title of the product
- Product Category: Category assigned to the product
- Price: Retail price in USD
- Product Ratings: Average customer rating (1 to 5 stars)
- Warranty Period: Number of years covered under warranty
- Stock Quantity: Current inventory available
- Product Dimensions: Physical dimensions in cm (Length × Width × Height)
- Expiration Date: For perishable or limited-time goods

This dataset was selected because it allowed us to apply a wide range of techniques from the MGMT 59000-144 course, including graph algorithms, dynamic programming, linear/integer programming, and machine learning. By simulating demand patterns and optimization scenarios on realistic product attributes, we were able to generate actionable business insights relevant to supply chain, product performance, and logistics planning.

Exploratory Data Analysis

To better understand the structure, quality, and analytical potential of the dataset, we conducted a comprehensive exploratory data analysis using Python libraries such as pandas, matplotlib, and seaborn.

The dataset sourced from Kaggle was clean, with no missing values. Key columns—including dates—were correctly typed, and temporal features such as **Product Age**, **Time to Expire**, and **Lifespan** were successfully engineered from the manufacturing and expiration dates.

Price ranged from approximately **$10 to $500**, with a relatively uniform distribution and a mean near **$254**. **Inventory levels** were evenly distributed, mostly falling between **1 and 100 units** per product. **Products** were well-distributed across three categories: **Electronics**, **Clothing**, and **Home Appliances**, with Home Appliances showing slightly higher average prices. **Customer ratings** were spread across the full 1–5 scale with a slight skew toward positive reviews, but with a notable concentration around **3**, indicating neutral sentiment. Importantly, ratings showed **no strong correlation with price or warranty**, emphasizing their subjectivity.

A **correlation heatmap** revealed generally weak relationships across features, except a strong correlation between **Time to Expire** and **Lifespan** ($\rho \approx 0.96$), as expected.

This EDA helped shape several of the subsequent insights, including demand modeling, rating prediction, and inventory planning.

Insights Observed:

Here is a brief overview of the insights we have gained from our analysis:

Insight Description	Technique Used	Area
Section 1: Product Relationships & Bundling		
1. Cluster products based on shared product tags to find similar item groups.	Graph Algorithms – Connected Components (tag-based similarity)	Graph Algorithms
2. Suggest complementary product bundles using tag overlap and cluster co-occurrence.	Co-occurrence Matrix + Network Analysis	Product Bundling
Section 2: Demand Forecasting & Inventory Optimization		
3. Predict daily demand using product features (price, rating, category).	Linear Regression (feature-based prediction)	Machine Learning
4. Optimize restocking strategy to avoid expired inventory using predicted demand.	Dynamic Programming-style Cost Minimization	Inventory Strategy
5. Schedule future inventory arrivals across months to avoid out-of-stock.	Time Series Distribution Planning	Logistics
Section 3: Logistics & Fulfillment		
6. Minimize shipment volume while ensuring category diversity.	Integer Linear Programming (Volume Optimization using PuLP)	Fulfillment Efficiency
7. Maximize product ratings under shipment volume constraint.	Binary Integer Programming (Knapsack Formulation using PuLP)	Shipment Prioritization
Section 4: Product Ratings & Performance		
8. Predict product rating based on price, category, and warranty.	Decision Tree Regressor	Machine Learning
9. Identify low-rated yet high-demand products for improvement.	Multi-Criteria Filtering (high demand + low rating)	Product Quality Insights

Insight 1: Clustering Products Based on Shared Tags

Methodology:

To uncover logical groupings among products, we constructed an undirected graph where each product was represented as a node. Edges were added between products that shared at least one common tag. An inverted index was used to map each tag to a list of associated products, and all pairwise connections were created for products sharing a tag. Once the graph was built, we applied connected component analysis using the networkx.connected_components() function to identify clusters of interconnected products. Each connected component represented a cluster of products that shared one or more tags.

Key Findings:

The analysis resulted in 6,310 unique product clusters. While the majority of these clusters were small, a few contained a notable number of products. The largest cluster had 18 products, and several others included 12 to 13 items. Products in the largest cluster included laptops, monitors, smartphones, and headphones, all of which were linked through tags such as "UFU", "C15", and "O1G". A visual subgraph of this cluster confirmed strong internal connectivity, indicating that shared tags effectively grouped related or complementary products.

Business Implications:

Tag-based clusters enhance product recommendations, support cross-selling through curated bundles, and improve search relevance. Clusters also enable category-level insights for marketing and inventory planning.

Insight 2: Suggest Complementary Product Bundles

How the Insight Was Obtained:

This insight builds on the graph-based clusters generated in Insight 1. Products within each cluster were analyzed for partial tag overlap, using Jaccard similarity to measure how similar their tag sets were. Each product's tags were split into lists, and Jaccard similarity was computed for all pairs within a cluster. Pairs that shared some—but not all—tags were flagged as complementary. This helped distinguish truly complementary products (which add value together) from highly similar or redundant ones. Product pairs with the highest similarity scores were then ranked and visualized using bar plots and a tag-based network graph.

Key Findings:

The top recommended bundles had a Jaccard similarity score of 0.667, indicating strong but not complete tag overlap. These included: Monitor + Smartphone; Smartphone + Headphones; Laptop + Smartphone. These combinations often crossed product categories, suggesting potential for effective

cross-selling. The network graph visually reinforced the tag-level relationships among these bundled items.

Business Implications:

These complementary pairs are ideal for "Bundle & Save" promotions or "Frequently Bought Together" suggestions. They boost average order value, support inventory turnover, and enhance customer experience by recommending value-adding combinations rather than duplicates.

Insight 3: Predict Daily Demand Using Product Attributes

How the Insight Was Obtained:

A synthetic daily demand variable was generated based on product attributes to simulate real-world behavior.

The features used included: Product ratings; Price; Days to expiration (derived from expiration date); Warranty period;

Preprocessing steps included: Converting expiration dates to datetime; Calculating Days to Expire from a fixed reference date; One-hot encoding the Product Category to incorporate categorical information;

The dataset was split into training and testing sets using train_test_split. A linear regression model from sklearn was trained to predict daily demand using the engineered features.

Model performance was evaluated using: R-squared to assess goodness of fit; Mean Absolute Error (MAE) to quantify prediction accuracy

This approach replicated a real-world forecasting scenario where multiple product attributes influence short-term demand at a per-item level.

Key Findings:

The model achieved a reasonable R^2 score and low MAE, indicating that key product features—especially shelf life and pricing—are predictive of daily demand. Demand tends to drop as the expiration date approaches or as price increases, aligning with common consumer behavior.

Business Implication:

Demand prediction models can support smarter restocking, dynamic pricing, and inventory prioritization strategies.

Insight 4: Optimize Restocking with Predicted Demand

How the Insight Was Obtained:

This analysis applied a dynamic programming-style cost minimization strategy to determine whether products nearing expiration should be retained or discarded. The goal was to evaluate the trade-off between the expected profit from future sales and the combined cost of holding inventory and losing unsold stock to expiration.

For each product, daily demand was estimated using a predictive formula based on product ratings, price, and category. Days to Expiration was computed by comparing the product's expiration date to the current date. Using these values, the model: Calculated the expected number of units sold before expiration; Estimated holding costs (based on inventory days) and expiration penalties for unsold units; Calculated net cost if inventory is held (holding cost + expiry loss – profit); Compared this against the cost of discarding all stock immediately.

The action with the lower total cost was selected for each product: "Hold & Sell" or "Discard All".

Visual tools such as bar plots, scatter plots, and heatmaps were used to analyze how decision outcomes varied with stock levels and shelf life.

Key Findings:

Around 68% of products were profitable to retain, especially those with moderate stock and longer shelf life. Products with large stock and short time to expiration were more likely to be flagged for disposal. A heatmap of stock quantity versus expiry window revealed that high cost-to-hold clusters emerged when inventory was both large and close to expiration. The scatter plot showed a clear cost break-point, highlighting which items crossed from profit to loss zones.

Business Implication:

Helps reduce losses from overstock and expired goods by supporting automated cleanup and smarter discard decisions.

Insight 5: Schedule Monthly Replenishment to Avoid Stockouts

How the Insight Was Obtained:

This insight was generated using a time-series style inventory simulation, applying logic similar to rolling horizon planning used in dynamic inventory control systems. The model simulates monthly inventory levels over a 6-month window, guided by forecasted daily demand from Insight 3.

Predicting daily demand per product using a weighted formula based on product ratings, price, and encoded category. Simulating stock consumption each month as:

$$\text{Monthly Consumption} = \text{Daily Demand} \times 30$$

If projected stock for a month fell below a reorder threshold (15 days' worth of demand), the model triggered a replenishment to bring inventory back up to 30 days' coverage. This logic was repeated month-by-month, recording both projected inventory levels and the quantity restocked in each cycle.

Key Findings:

Products like Laptops and Headphones, which had low starting stock and relatively high daily demand, required multiple replenishment cycles to avoid stockouts.

In contrast, products like Smartphones—which began with ample stock—did not require any restocking over the six-month period.

The simulation helped quantify the trade-off between high initial stock and the efficiency of staggered monthly replenishments.

Business Implication:

Enables proactive restocking, minimizing stockouts and reducing excess inventory through smarter, automated planning.

Insight 6: Minimize Shipment Volume While Ensuring Product Diversity

How the Insight Was Obtained:

Minimize the total shipment volume of selected products. This is expressed mathematically as:

$$\min \sum_i x_i \cdot \text{Volume}_i$$

where x_i is the number of units shipped for product i, and Volume_i is its volume in cm^3.

Decision Variables: $x_i \in Z$ Integer variable representing the number of units to ship for product i. It is bounded by the product's available stock. $Y_c \in \{0,1\}$ Binary variable indicating whether any product from category c is selected (1) or not (0)

Constraints: Minimum Units Constraint: The total number of units shipped must be at least 100; Category Linkage Constraint: If a product from category cc is selected, then ycy_c must be active; Diversity Constraint: Ensure that the shipment includes products from at least 3 distinct categories

This ILP model ensures that the shipment is both efficient in volume and diverse in product mix, using only highly rated products (rating ≥ 3). It reflects real-world constraints found in logistics operations such as truck space, SKU representation requirements, and customer satisfaction filtering.

Key Findings:

The model selected exactly 100 units across 3 product categories, satisfying both volume efficiency and diversity goals. Products chosen were highly rated and had lower unit volumes, optimizing both customer satisfaction and shipping cost.

Business Implication:

This approach enables logistics teams to pack shipments efficiently while maintaining product variety, which is valuable for promotional bundles, retail mix balance, or customer engagement.

Insight 7: Maximize Product Ratings Within Shipment Volume Constraints

How the Insight Was Obtained:

The objective is to maximize the total customer rating score of the selected products while staying within a fixed shipment volume. Formally:

$$\max \sum_i x_i \cdot \text{Rating}_i$$

where $x_i \in \{0,1\}$ indicates whether product ii is included in the shipment.

Decision Variables: $X_i \in \{0,1\}$ A binary variable indicating whether product ii is selected (1) or not (0). This forms a 0-1 knapsack problem, where each product has a "value" (rating) and a "cost" (volume).

Constraints: Volume Constraint: The combined volume of all selected products must not exceed 300,000 cm^3; Minimum Product Count Constraint: At least 10 products must be selected

This integer program prioritizes the inclusion of highly rated products while respecting strict shipment capacity, making it well-suited for constrained logistics scenarios like container packing or express fulfillment.

Key Findings:

The optimization selected 150+ products with an average rating above 4, all within a strict 300,000 cm^3 volume limit. High-rated products with reasonable size were prioritized, while bulky or low-rated items were excluded.

Business Implication:

This model is ideal for high-value shipments (e.g., express orders, curated sets), helping maximize customer satisfaction when space is limited and quality matters.

Insight 8: Predict Product Ratings Using Product Features

How the Insight Was Obtained:

We developed a supervised regression model using a Decision Tree Regressor, trained on product-level features. The dataset included price, warranty period, and product category, with categorical variables one-hot encoded to prepare them for the model. The data was split into training and test sets (80/20), and a decision tree with a controlled depth of 5 was trained to reduce overfitting. Model

performance was evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). We also extracted feature importance scores to understand which variables most influenced predicted ratings.

Key Findings:

The model achieved an MAE of 1.214 and an RMSE of 1.422—respectable given the limited number of input features and the simplicity of the model. Price emerged as the dominant predictor, accounting for approximately 85% of the model's decision weight, indicating that customers tend to associate value or expectations heavily with price. Product category and warranty period had marginal but non-negligible effects. A scatter plot comparing actual and predicted ratings showed solid central alignment but weaker performance at the extremes, suggesting room for improvement in modeling highly polarized ratings.

Business Implication:

This model helps estimate expected product ratings when reviews are unavailable, flagging high-priced items with low predicted satisfaction for closer review or repositioning.

Insight 9: Identify Low-Rated Yet High-Demand Products

How the Insight Was Obtained:

We used a multi-criteria filtering approach that combined product ratings and predicted daily demand. Predicted demand values were carried over from Insight 3, calculated using a weighted formula incorporating price, customer rating, and product category. Products were flagged if they met two criteria:

(1) a low average rating (≤ 2.5), and

(2) high predicted demand (at or above the 75th percentile across the dataset).

This simple yet effective rule-based method allowed us to isolate underperforming products that still drive significant sales volume.

Key Findings:

The filtering process identified more than ten products that were both poorly rated and in high demand. These included smartphones, monitors, and headphones with predicted daily demand above 0.25 units and average ratings below 2.5. The results were validated with a scatter plot comparing predicted demand and product ratings, where the flagged products clustered in a clear risk zone. A follow-up bar chart highlighted the top 10 such products by predicted demand, underscoring their commercial impact despite customer dissatisfaction.

Business Implication:

These products need urgent review for quality or positioning issues. The method is simple to automate and supports ongoing product health monitoring.

Appendix

Fig 1: Distribution of Product Prices

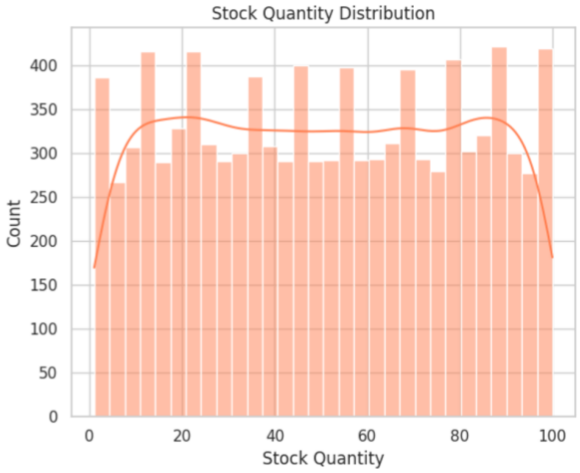

Fig 2: Distribution of Stock Quantities

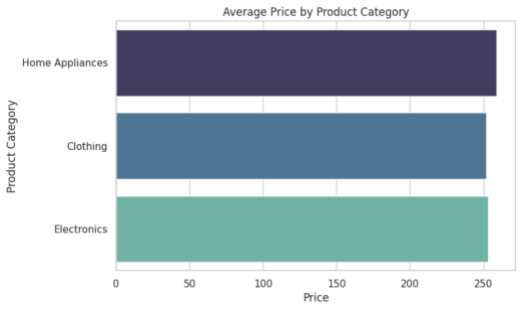

Fig 3: Average Price by Product Category

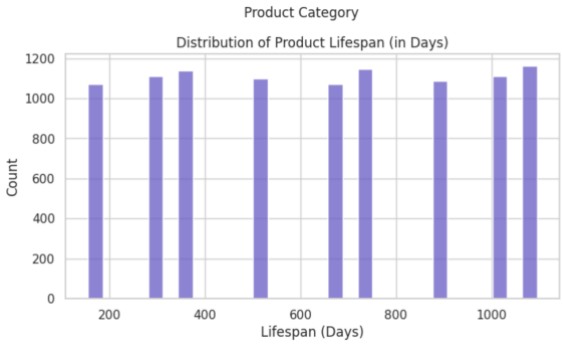

Fig 4: Product Lifespan Histogram

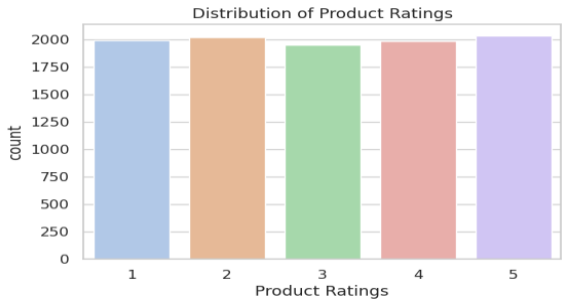

Fig 5: Distribution of Product Ratings

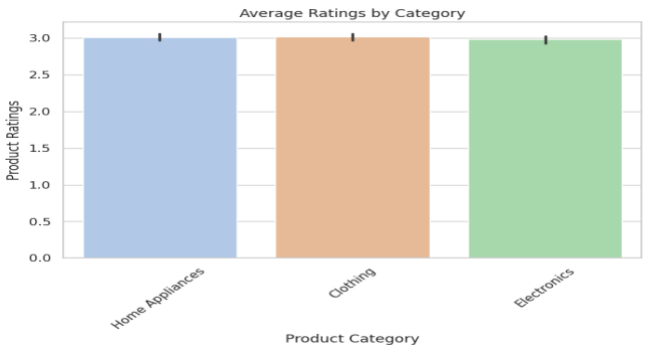

Fig 6: Average Ratings by Category

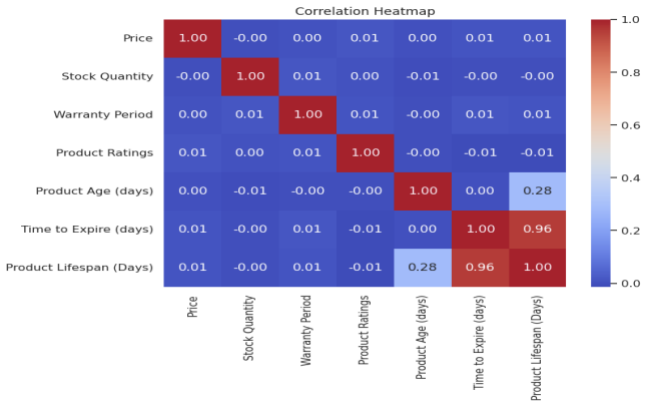

Fig 7: Correlation Heatmap

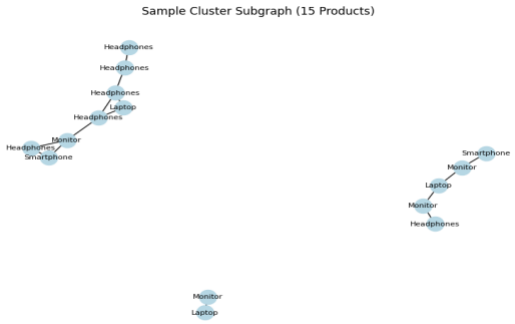

Fig 8: Sample Cluster Graph (Insight 1)

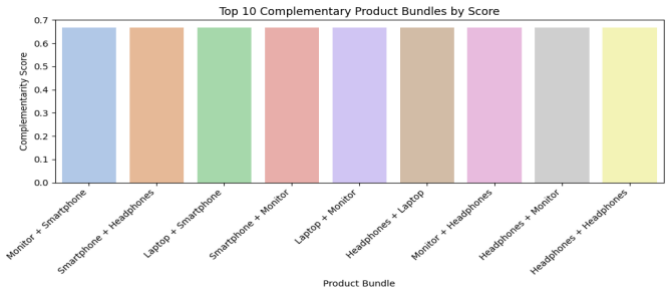

Fig 9: Top 10 Complementary Product Bundles by Score (Insight 2)

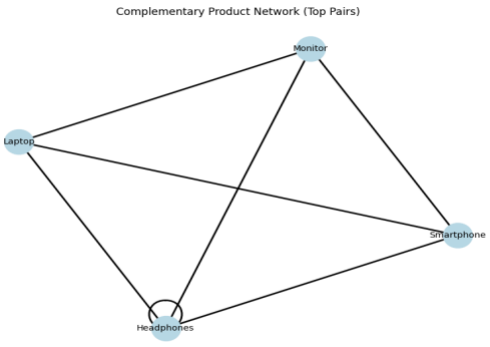

Fig 10: Complementary Product Network (Insight 2)

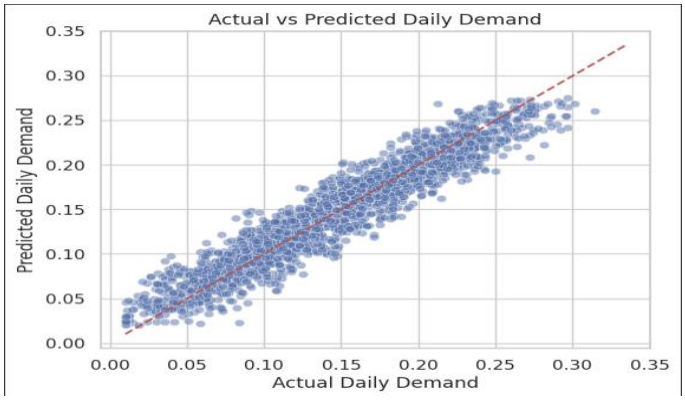

Fig 11: Actual vs Predicted Daily Demand (Insight 3)

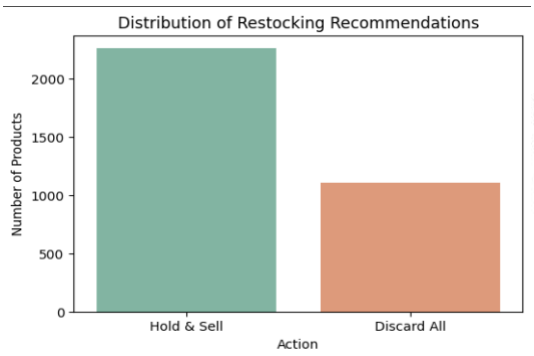

Fig 12: Distribution of Restocking Recommendations (Insight 4)

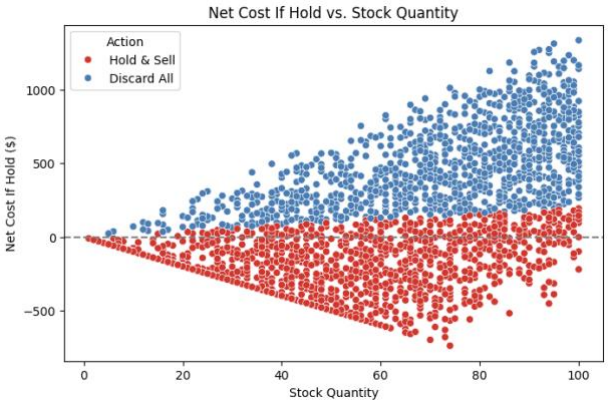

Fig 13: Net Cost If Hold vs. Stock Quantity (Insight 4)

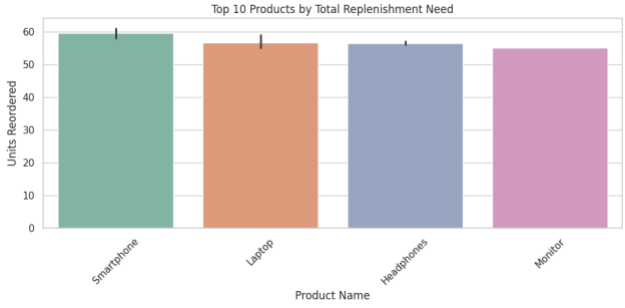

Fig 14: Top 10 Products by Total Replenishment Need (Insight 5)

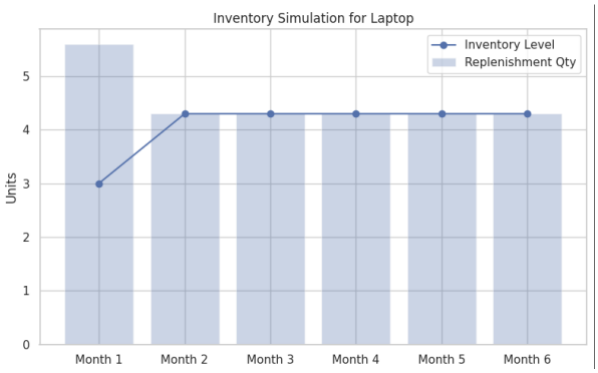

Fig 15: Inventory Simulation for Laptop (Insight 5)

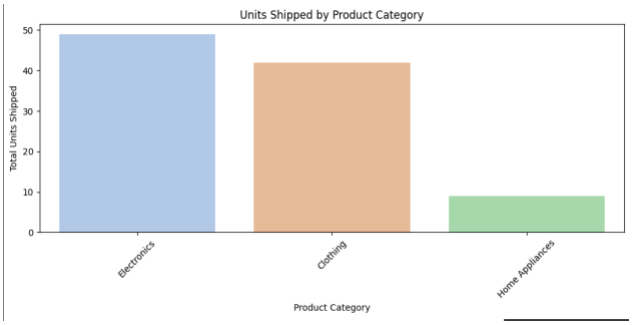

Fig 16: Units Shipped by Product Category (Insight 6)

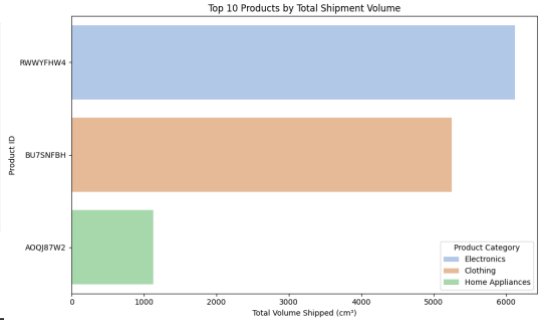

Fig 17: Top 10 Products by Total Shipment Volume(Insight 6)

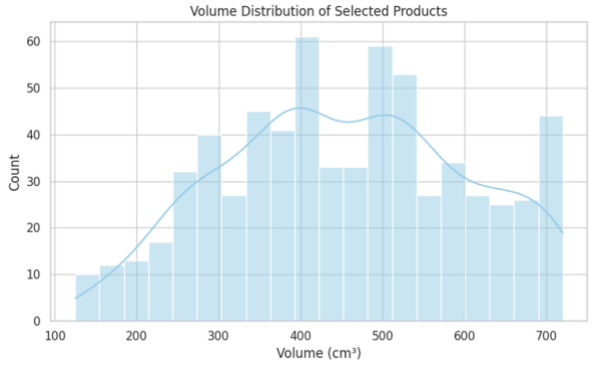

Fig 18: Volume Distribution of Selected Products (Insight 7)

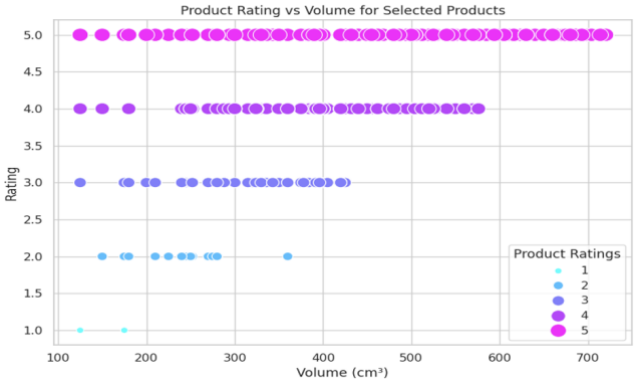

Fig 19: Product Rating vs Volume for Selected Products (Insight 7)

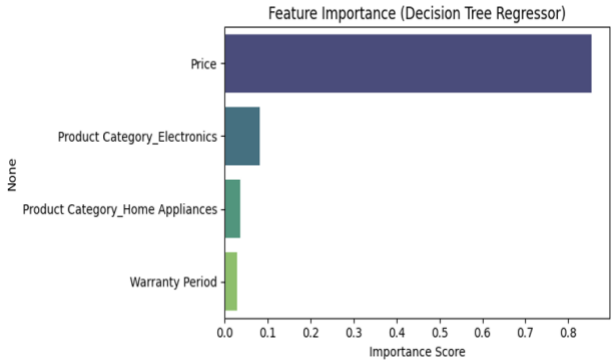

Fig 20: Feature Importance (Decision Tree Regressor) (Insight 8)

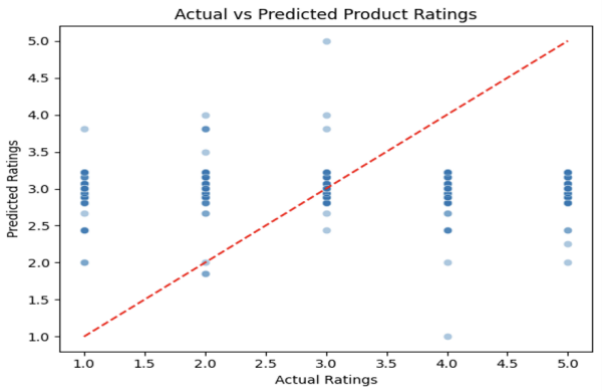

Fig 21: Actual vs Predicted Product Ratings (Insight 8)

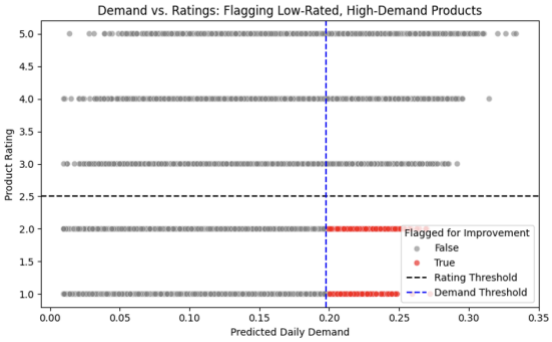

Fig 22: Demand vs. Ratings (Insight 9)

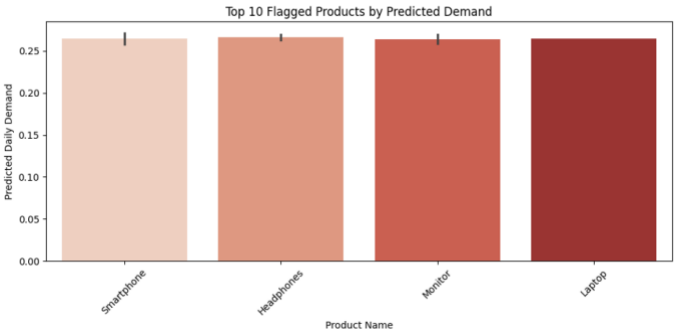

Fig 23: Top 10 Flagged Products by Predicted Demand (Insight 9)