Data-Driven Optimization of Inventory and Marketing for a D2C Artisanal Brand

A Mid-Term Report for the BDM Capstone Project

Submitted by:

Name: Aanuja Singh Chowhan
Roll number: 23f3002503

IITM Online BS Degree Program,
Indian Institute of Technology, Madras, Chennai
Tamil Nadu, India, 600036

Table Of Contents

Executive Summary

This capstone project, titled “Data-Driven Optimization of Inventory and Marketing for a D2C Artisanal Brand,” presents a structured, analytics-driven framework to support strategic decision-making at Hastvikas, a Direct-to-Consumer (D2C) platform championing Indian handicrafts. The objective is to enhance inventory efficiency and marketing precision through the intelligent application of data science methods.

The project is divided into two core phases, descriptive analytics and advanced applications: leveraging primary data collected from Amazon Seller Central and Shopify Insights. The initial phase involved Excel-based descriptive techniques to identify early patterns in product performance and return behavior. Now, with more detailed Shopify insights, it’s possible to optimise both inventory and marketing. Tools such as heatmaps, Pareto charts, and frequency analysis were used to highlight high-volume SKUs and refund-heavy items. However, these methods offered limited strategic insight for forward-looking actions.

The final phase introduced a suite of Machine Learning (ML) and Business Data Management (BDM) techniques to address key operational challenges:

- K-Means Clustering for dynamic SKU segmentation into high-potential, risky, or investigable product groups.
- Logistic Regression Modeling to flag return-prone SKUs prior to restocking decisions.
- Shopify Funnel Simulation to quantify revenue leakage at each stage of the customer journey.
- Search Query Matching using TF-IDF and cosine similarity to improve catalog alignment with real user search intent.
- SARIMA Time Series Forecasting to project weekly order volumes and support agile inventory planning.

Together, these models form an integrated, data-first decision system that enables Hastvikas to forecast demand, manage return risks, and optimize catalog discoverability. The outcome is a scalable blueprint for inventory and marketing optimization suited to the unique context of a D2C artisanal brand operating in competitive digital marketplaces.

2. Detailed Analysis of Processes and Methods Used

A comprehensive set of data-driven procedures was designed to address critical business objectives, including improving catalog discoverability, understanding customer drop-offs, and minimizing financial losses due to returns. Each method was selected based on its operational simplicity, interpretability, and readiness for integration with internal workflows. This section outlines the rationale and processing steps for each method.

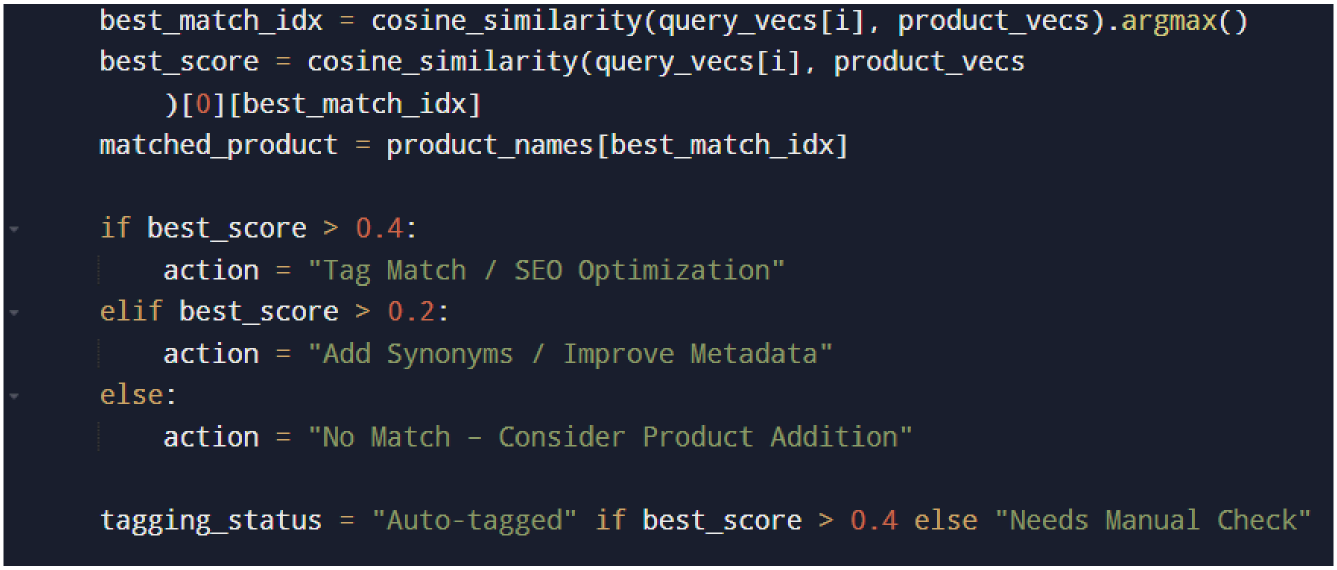
2.1 Search Query Matching and Catalog Optimization

Objective: To assess the alignment between user search behavior and the product catalog, and to identify opportunities for improving product visibility through metadata refinement.

Data Processing Steps:

- Collected high-frequency search queries from the website and compiled product titles from the live catalog.
- Cleaned and standardized the text data by lowercasing and removing noise to ensure comparability.
- Converted both datasets into numerical form using a weighting technique that assigns importance based on term uniqueness.
- Calculated similarity scores between each query and all product titles to identify the closest match.
- Classified the strength of each match into categories guiding action such as direct tagging, metadata enhancement, or new product consideration.

Reason for Selection: This approach allowed scalable analysis of unstructured text using unsupervised learning without requiring historical engagement data. It was especially useful for flagging missed tagging opportunities across the catalog.



```
    best_match_idx = cosine_similarity(query_vecs[i], product_vecs).argmax()
    best_score = cosine_similarity(query_vecs[i], product_vecs
        )[0][best_match_idx]
    matched_product = product_names[best_match_idx]

    if best_score > 0.4:
        action = "Tag Match / SEO Optimization"
    elif best_score > 0.2:
        action = "Add Synonyms / Improve Metadata"
    else:
        action = "No Match - Consider Product Addition"

    tagging_status = "Auto-tagged" if best_score > 0.4 else "Needs Manual Check"
```

Code Block 1: Search Query Matching and Catalog Optimization

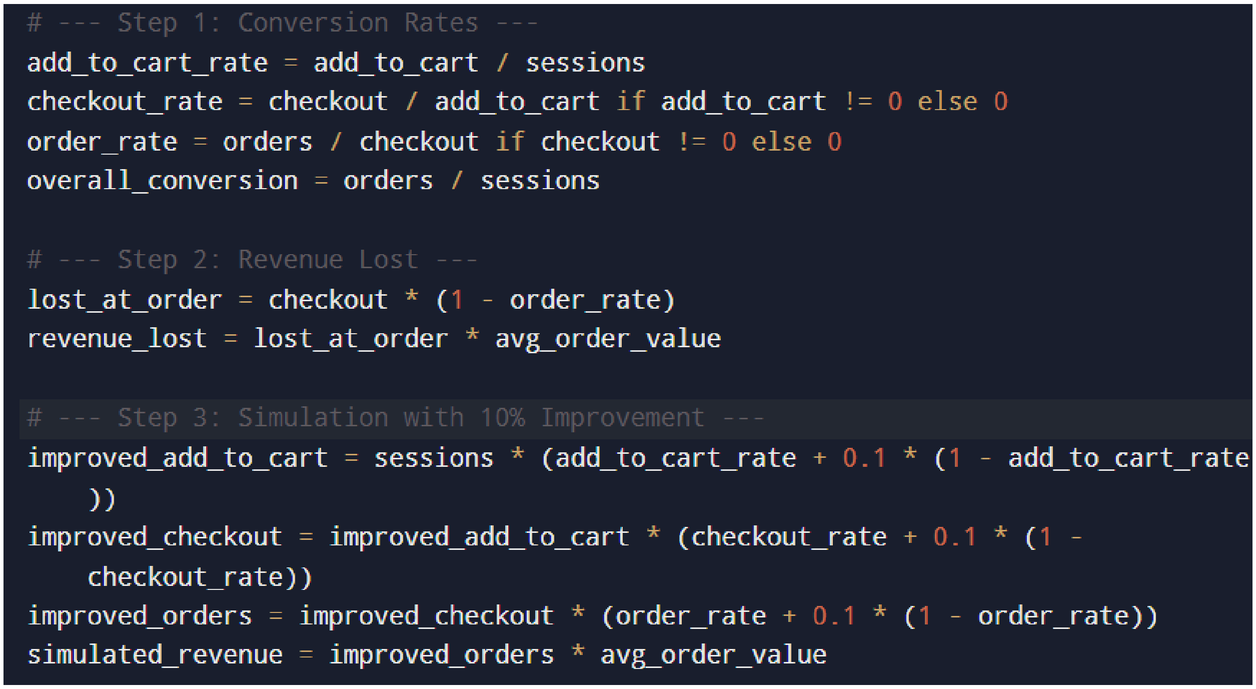
2.2 Customer Journey Funnel Breakdown and Drop-Off Simulation

Objective: To measure where users disengage in the path to purchase and to simulate the impact of small improvements in conversion efficiency at each stage.

Data Processing Steps:

- Extracted user flow data from platform analytics, covering sessions, cart additions, checkouts, and order completions.
- Calculated conversion rates between each sequential step using ratio-based formulas.
- Computed drop-off percentages to identify the most critical friction points in the journey.
- Simulated improved performance by applying a fixed efficiency uplift across stages and projected the resulting impact on conversions and revenue.
- Created visual charts with annotations to communicate findings to decision-makers.

Reason for Selection: This method provided an intuitive breakdown of user engagement across funnel stages. It facilitated quick comparisons between actual performance and projected scenarios without requiring deep behavioral modeling.



```
# --- Step 1: Conversion Rates ---
add_to_cart_rate = add_to_cart / sessions
checkout_rate = checkout / add_to_cart if add_to_cart != 0 else 0
order_rate = orders / checkout if checkout != 0 else 0
overall_conversion = orders / sessions

# --- Step 2: Revenue Lost ---
lost_at_order = checkout * (1 - order_rate)
revenue_lost = lost_at_order * avg_order_value

# --- Step 3: Simulation with 10% Improvement ---
improved_add_to_cart = sessions * (add_to_cart_rate + 0.1 * (1 - add_to_cart_rate
    ))
improved_checkout = improved_add_to_cart * (checkout_rate + 0.1 * (1 -
    checkout_rate))
improved_orders = improved_checkout * (order_rate + 0.1 * (1 - order_rate))
simulated_revenue = improved_orders * avg_order_value
```

Code Block 2: Customer Journey Funnel Breakdown and Drop-Off Simulation

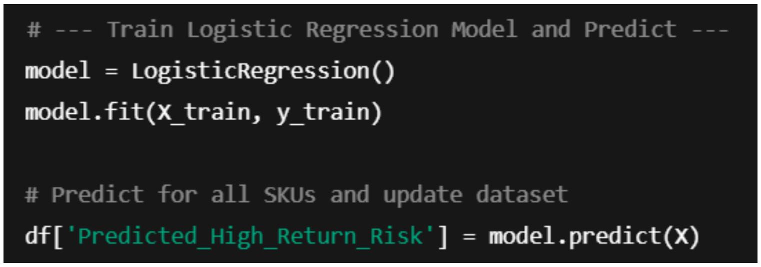
2.3 Return Risk Classification and Product Flagging

Objective: To predict which SKUs are likely to incur high returns and enable risk-aware inventory and promotion planning.

Data Processing Steps:

- Used historical order data containing total sales, number of returns, and refund amounts per product.
- Created a binary label based on whether a product's return ratio exceeded a defined threshold.
- Selected relevant numerical fields to act as input variables for classification.
- Applied a simple and interpretable algorithm to identify patterns associated with high-risk products.
- Evaluated prediction accuracy using standard metrics and appended the output to the SKU database.

Reason for Selection: This technique allowed fast and explainable identification of high-risk SKUs using structured data. It helped teams prioritize corrective action on a data-informed basis without requiring detailed customer return feedback.



```
# --- Train Logistic Regression Model and Predict ---
model = LogisticRegression()
model.fit(X_train, y_train)

# Predict for all SKUs and update dataset
df['Predicted_High_Return_Risk'] = model.predict(X)
```

Code Block 3: Return Risk Classification and Product Flagging

2.4 Product Segmentation Using Behavioral Clustering

Objective: To group SKUs by shared behavioral characteristics and assign collective strategies such as promotion, investigation, or redesign.

Data Processing Steps:

- Enhanced the dataset by calculating a new field representing average refund per return.
- Standardized all numerical features to ensure fair comparison across dimensions.
- Applied a grouping algorithm that formed natural clusters based on similarities in return ratio, refund amounts, and order volume.
- Reduced the number of dimensions for visual interpretation using a mathematical projection method.
- Assigned operational actions to each group based on average performance, enabling batch-level decisions.

Reason for Selection: This unsupervised technique made it possible to evaluate product performance at a strategic level. It helped consolidate insights across hundreds of SKUs into three clear action categories for the sourcing and catalog teams.

2.5 Structuring and Communicating Results for Operational Use

Objective: To convert analytical outputs into formats that support direct use by merchandising, marketing, and operations teams.

Data Processing Steps:

- Merged all processed data into clean Excel files, with additional columns for predicted risk scores, segment tags, and recommended actions.
- Generated a report containing structured tables, charts, and plain-language summaries for each method.
- Used visualizations such as bar charts and scatter plots to present key insights clearly and efficiently.
- Delivered the outputs in editable formats to facilitate quick integration with internal planning tools and team discussions.

Reason for Selection: Clear documentation and structured reporting ensured that insights moved beyond analysis and into implementation. This improved team adoption and decision-making speed across functions.

2.6 Time Series Forecasting for Weekly Orders Using SARIMA

Objective: To forecast the number of orders for the next 8 weeks using historical weekly data, enabling better planning for supply chain, marketing, and resource allocation.

Data Processing Steps:

1. Compiled and Parsed Historical Weekly Data:
Collected weekly order counts from January to June 2025 in the format "date range, number of orders." The starting date of each week was extracted to create a coherent time index for the time series.

2. Cleaned and Restructured Data:
Standardized the time series format by:

- Parsing only the start date from each weekly range to maintain uniform weekly intervals.
- Sorting the data chronologically to preserve the temporal sequence.
- Converting raw input into a pandas DataFrame with date indices and corresponding weekly order values.

3. Model Selection-SARIMA:

Selected the **Seasonal AutoRegressive Integrated Moving Average (SARIMA)** model due to:

- Weekly periodicity expected in order behavior (e.g., weekly spikes or lulls).
- The presence of trend and noise components in the historical data.
- Flexibility of SARIMA to model both seasonality and non-stationarity.

4. Model Configuration and Training:

Trained a SARIMA model with the configuration:

- Non-seasonal order: (1, 1, 1)
- Seasonal order: (1, 1, 1, 4)

  These values were chosen to balance model complexity with the limited number of historical data points. The seasonal period of 4 was used based on approximate monthly cycles (i.e., 4 weeks = 1 month).

5. Forecasting Future Values:

Predicted the number of orders for the next 8 weeks:

- The model generated point forecasts without requiring any exogenous variables.
- Confidence intervals were also available but not visualized in the current plot.

6. Visualization and Interpretation:

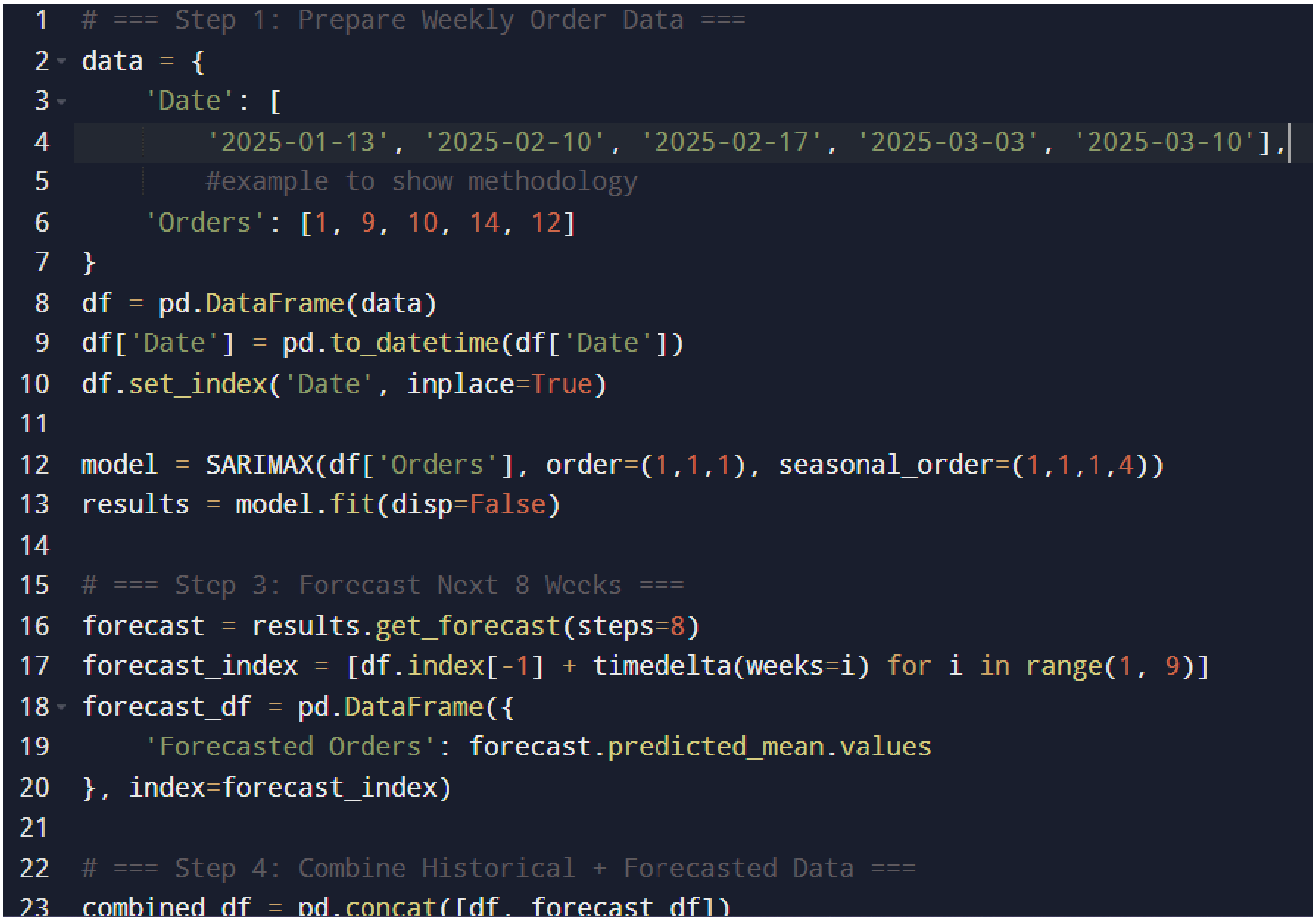
Plotted both historical and forecasted data in a time series chart:

- Solid lines represented actual historical weekly orders.
- Dashed lines indicated forecasted values from the SARIMA model.

Reason for Selection: This time series approach offered a statistically sound and scalable method to project future demand without needing additional variables or assumptions. It was particularly valuable for:

- Planning for short-term demand fluctuations.
- Visualizing expected trends when historical patterns were sparse or volatile.

- Generating actionable insights despite limited data using unsupervised temporal modeling.



```
# === Step 1: Prepare Weekly Order Data ===
data = {
    'Date': [
        '2025-01-13', '2025-02-10', '2025-02-17', '2025-03-03', '2025-03-10'],
        #example to show methodology
    'Orders': [1, 9, 10, 14, 12]
}
df = pd.DataFrame(data)
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)

model = SARIMAX(df['Orders'], order=(1,1,1), seasonal_order=(1,1,1,4))
results = model.fit(disp=False)

# === Step 3: Forecast Next 8 Weeks ===
forecast = results.get_forecast(steps=8)
forecast_index = [df.index[-1] + timedelta(weeks=i) for i in range(1, 9)]
forecast_df = pd.DataFrame({
    'Forecasted Orders': forecast.predicted_mean.values
}, index=forecast_index)

# === Step 4: Combine Historical + Forecasted Data ===
combined_df = pd.concat([df, forecast_df])
```

Code Block 3: Time Series Forecasting for Weekly Orders Using SARIMA

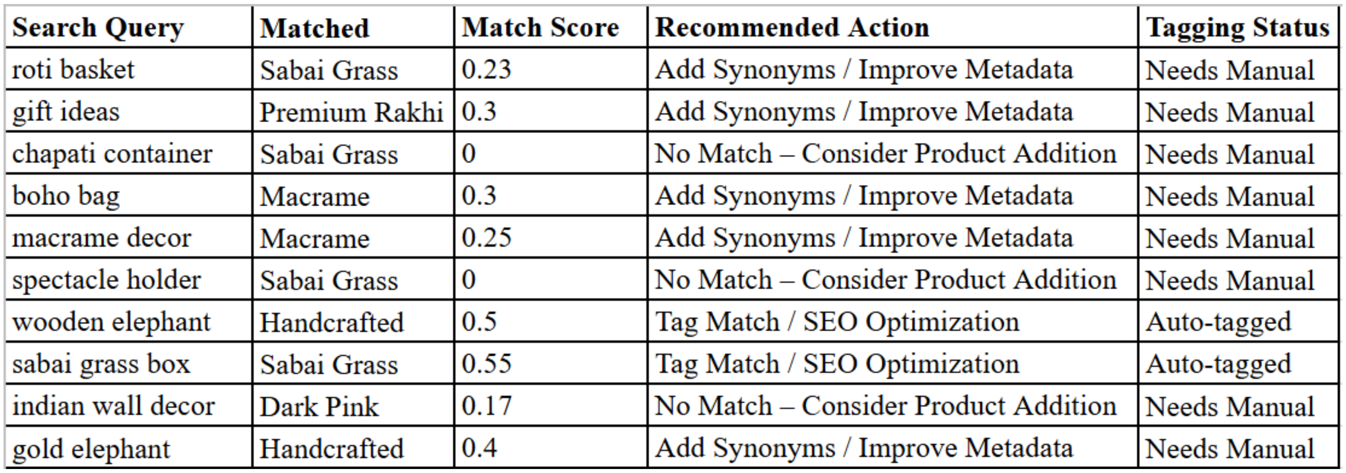
3. Results and Findings

This section outlines the key observations derived from structured analysis of the primary data collected from Hastvikas. The results are organized according to the specific analytical objectives of the project. Visual figure references have been included to support quantitative outputs.

3.1 Catalog Discoverability and Search Alignment

A comparative analysis of customer search queries and catalog product titles revealed the following:

- Out of the ten high-frequency search queries examined, only two had high alignment with existing product titles based on semantic similarity scores.
- Several queries such as *"roti basket"*, *"spectacle holder"*, and *"chapati container"* displayed low similarity scores (below 0.20), indicating limited alignment with any listed product.
- A number of queries matched products with moderate relevance, suggesting potential gaps in keyword coverage.
- Only two product titles were automatically classified as direct tag matches; the remaining required manual intervention for better metadata tagging.



Search Query	**Matched**	**Match Score**	**Recommended Action**	**Tagging Status**
roti basket	Sabai Grass	0.23	Add Synonyms / Improve Metadata	Needs Manual
gift ideas	Premium Rakhi	0.3	Add Synonyms / Improve Metadata	Needs Manual
chapati container	Sabai Grass	0	No Match – Consider Product Addition	Needs Manual
boho bag	Macrame	0.3	Add Synonyms / Improve Metadata	Needs Manual
macrame decor	Macrame	0.25	Add Synonyms / Improve Metadata	Needs Manual
spectacle holder	Sabai Grass	0	No Match – Consider Product Addition	Needs Manual
wooden elephant	Handcrafted	0.5	Tag Match / SEO Optimization	Auto-tagged
sabai grass box	Sabai Grass	0.55	Tag Match / SEO Optimization	Auto-tagged
indian wall decor	Dark Pink	0.17	No Match – Consider Product Addition	Needs Manual
gold elephant	Handcrafted	0.4	Add Synonyms / Improve Metadata	Needs Manual

Table 1: Search Query-Catalog Match Scores with Recommended Tagging Actions

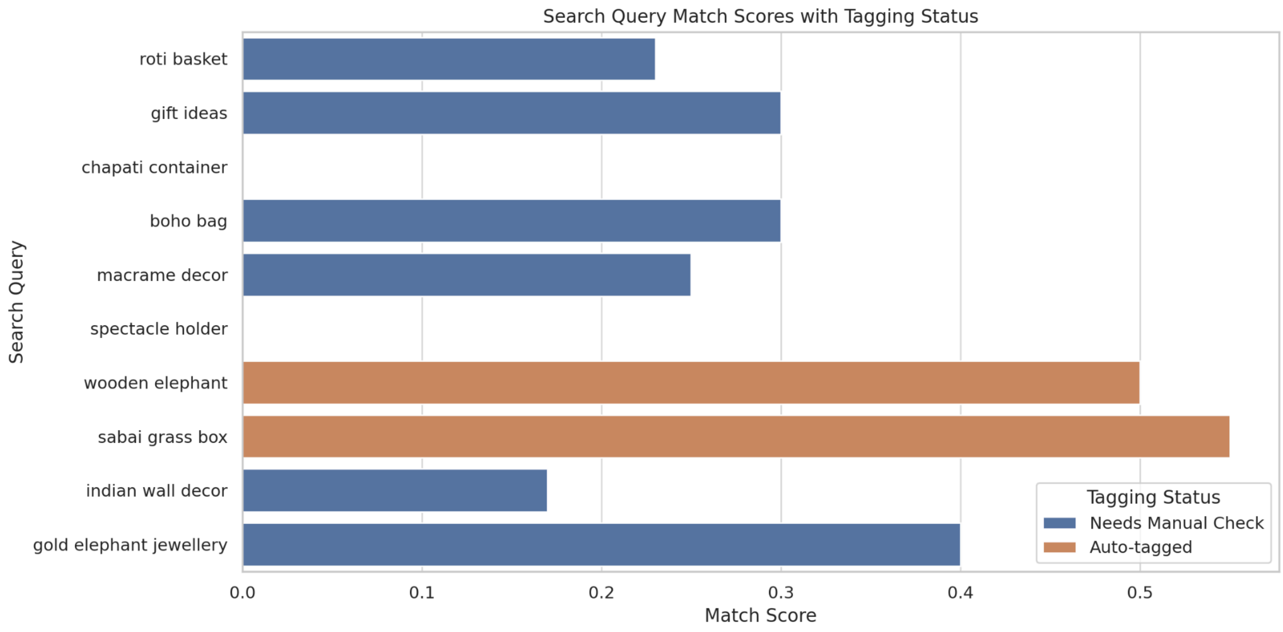


Figure 2: Visualisation of Search Query-Product Similarity Matrix with Action Flags

3.2 Purchase Funnel Drop-Offs

The stage-wise funnel data showed a substantial decrease in user count across the sales journey:

- The total number of sessions recorded during the data collection period was 3,351.
- Only 42 users proceeded to add a product to their cart, followed by 15 users who initiated checkout.
- Ultimately, only one order was completed during the period.
- The calculated conversion rates were lowest at the first transition point (session to cart), followed by a steep decline from checkout initiation to completed orders.

A simulation applying a 10 percent improvement in conversion efficiency at each stage demonstrated a significantly higher projected volume of completed purchases and associated revenue.

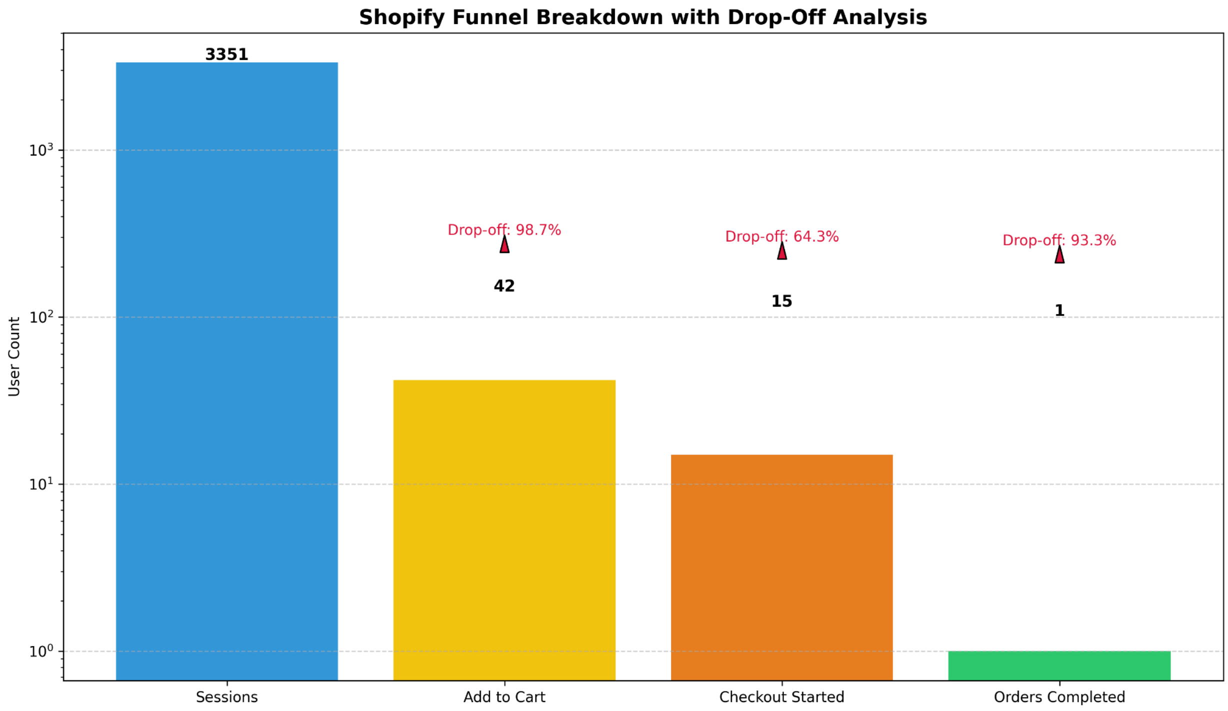
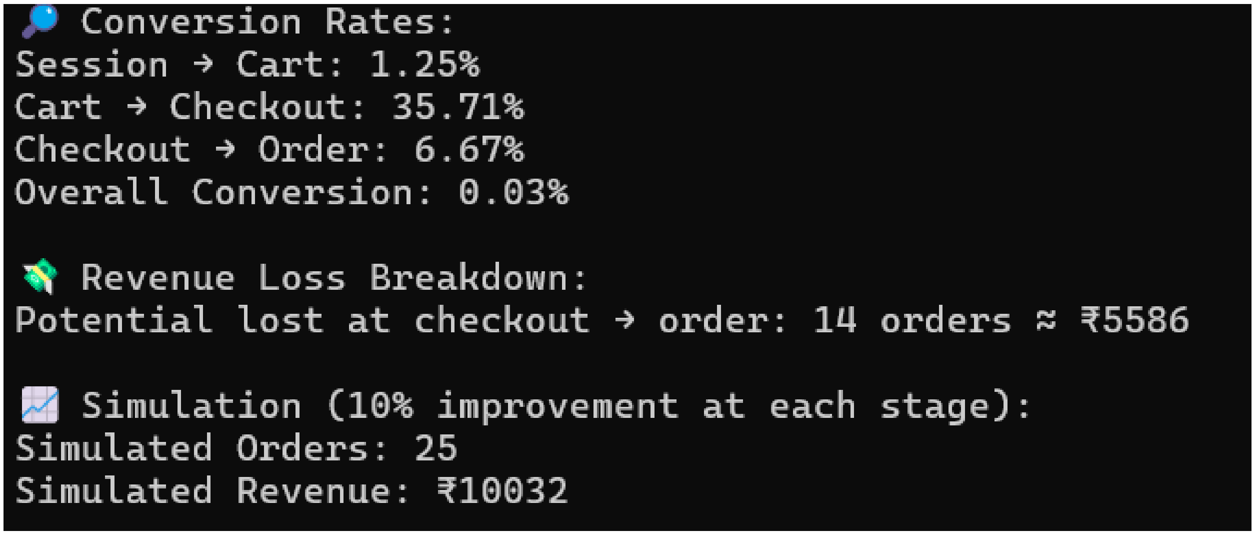


Figure 3: Stage-wise Funnel Chart of User Drop-Offs


```
🔎 Conversion Rates:
Session → Cart: 1.25%
Cart → Checkout: 35.71%
Checkout → Order: 6.67%
Overall Conversion: 0.03%

💸 Revenue Loss Breakdown:
Potential lost at checkout → order: 14 orders ≈ ₹5586

📈 Simulation (10% improvement at each stage):
Simulated Orders: 25
Simulated Revenue: ₹10032
```

Figure 4: Simulation Output with Projected Improvement

3.3 Return Risk Prediction

A predictive classification model was developed using historical order, return, and refund data. The findings were as follows:

- SKUs with a return ratio exceeding 15 percent were successfully flagged using the model.
- The model identified a subset of products that, despite low total order volumes, accounted for a significant proportion of total refunds.
- The classification output was evaluated using a confusion matrix and associated performance metrics (precision, recall, accuracy), which indicated reasonable reliability.
- A return risk label was assigned to each SKU in the dataset for further review.

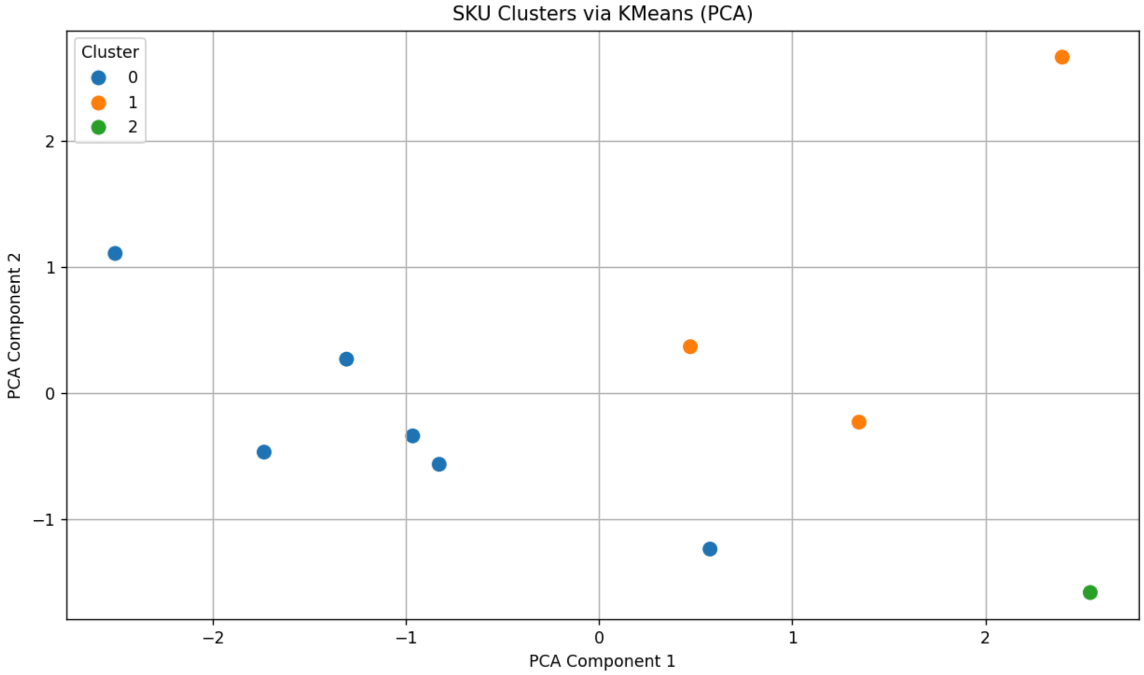


Figure 5: SKU Clusters Visualized Using K-Means Clustering on PCA-Reduced Features

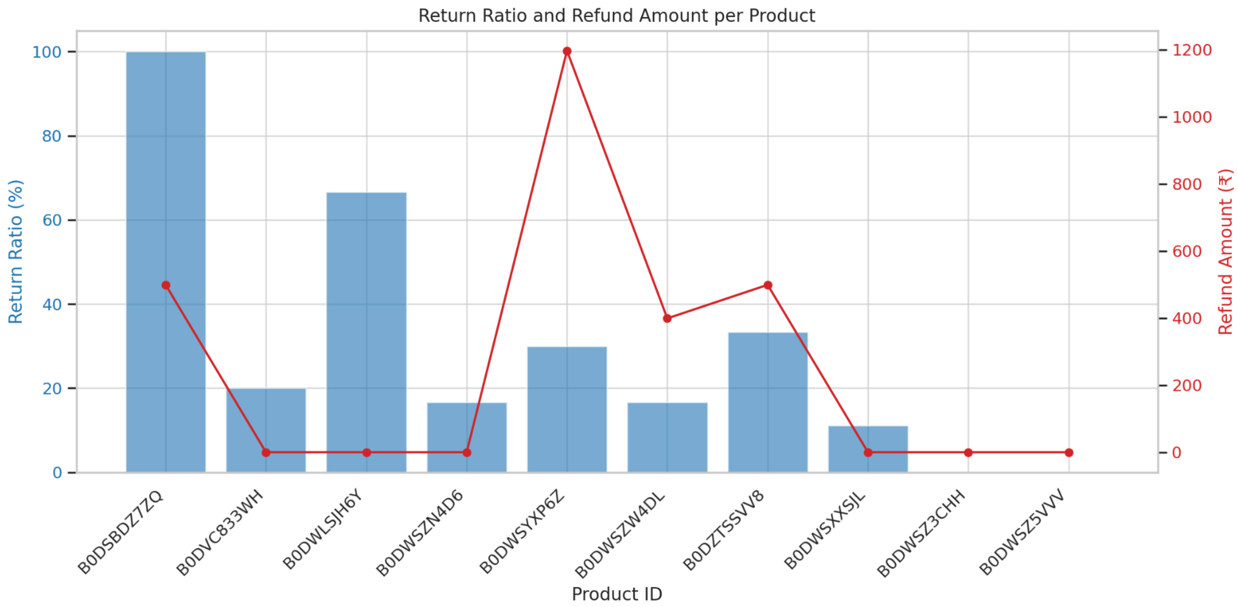


Figure 6: Comparison of Return Ratio and Refund Amount Across Products

3.4 Product Segmentation via Clustering

An unsupervised clustering approach was applied to segment SKUs based on behavioral variables. The process yielded the following cluster structure:

- Three distinct clusters were identified using historical data on order volume, return ratio, refund amounts, and engineered features.
- Cluster sizes were relatively balanced, each representing a unique SKU behavior pattern.
- Dimensionality reduction for visualization showed clear separation among clusters in a two-dimensional feature space.
- Each SKU was tagged with its cluster label and relevant performance indicators.

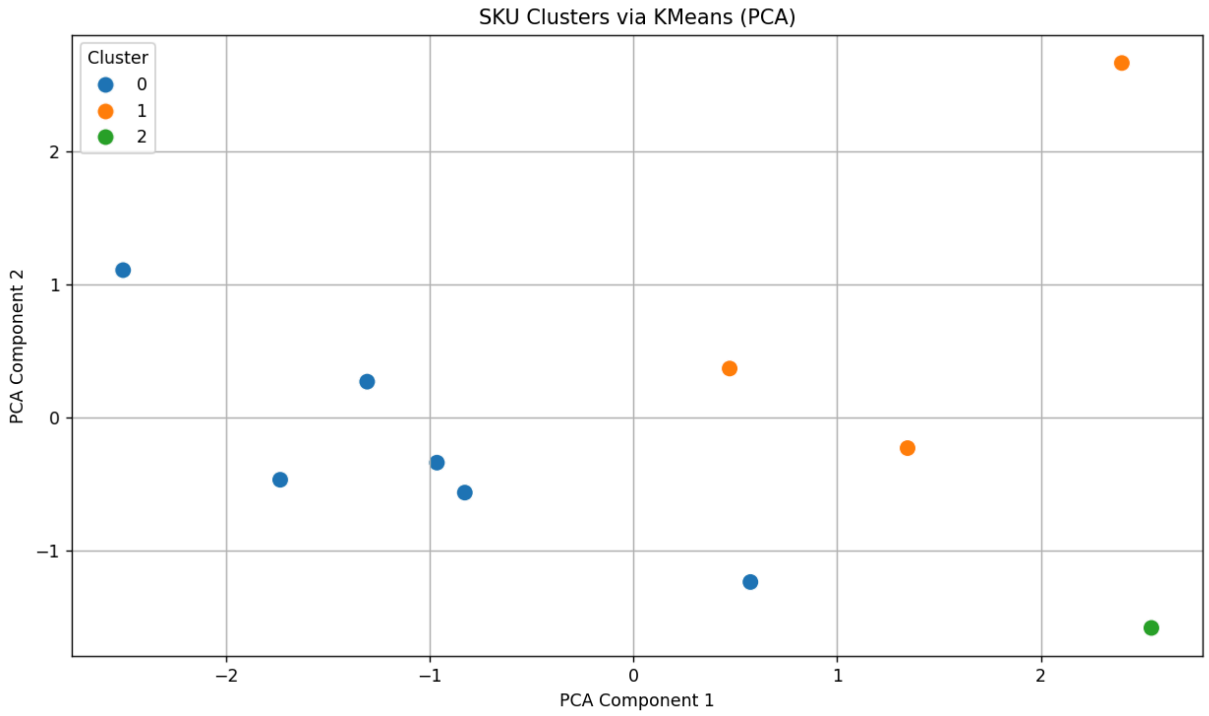


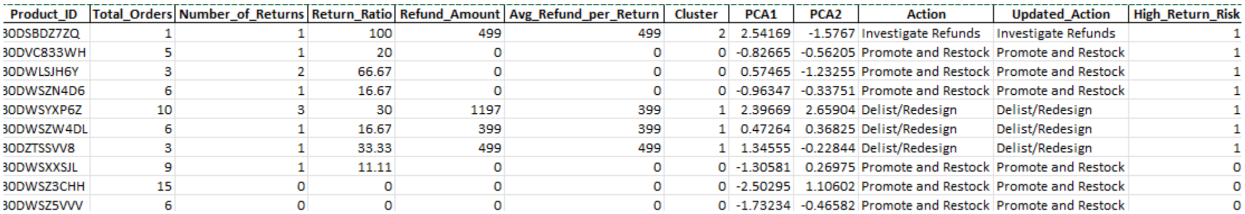
Figure 7: PCA Scatter Plot of SKU Clusters

3.5 Consolidated SKU Dataset

A unified SKU-level dataset was created by combining outputs from all analytical processes. Each product record included:

- The semantic match score with top search queries.
- Return risk prediction label.
- Cluster assignment based on product behavior.

The final dataset allows cross-referencing of discoverability, return behavior, and segmentation attributes at the individual SKU level.



Product_ID	Total_Orders	Number_of_Returns	Return_Ratio	Refund_Amount	Avg_Refund_per_Return	Cluster	PCA1	PCA2	Action	Updated_Action	High_Return_Risk
3ODSBDZ7ZQ	1	1	100	499	499	2	2.54169	-1.5767	Investigate Refunds	Investigate Refunds	1
3ODVC833WH	5	1	20	0	0	0	-0.82665	-0.56205	Promote and Restock	Promote and Restock	1
3ODWLSJH6Y	3	2	66.67	0	0	0	0.57465	-1.23255	Promote and Restock	Promote and Restock	1
3ODWSZN4D6	6	1	16.67	0	0	0	-0.96347	-0.33751	Promote and Restock	Promote and Restock	1
3ODWSYXP6Z	10	3	30	1197	399	1	2.39669	2.65904	Delist/Redesign	Delist/Redesign	1
3ODWSZW4DL	6	1	16.67	399	399	1	0.47264	0.36825	Delist/Redesign	Delist/Redesign	1
3ODZTSSVV8	3	1	33.33	499	499	1	1.34555	-0.22844	Delist/Redesign	Delist/Redesign	1
3ODWSXXSJL	9	1	11.11	0	0	0	-1.30581	0.26975	Promote and Restock	Promote and Restock	0
3ODWSZ3CHH	15	0	0	0	0	0	-2.50295	1.10602	Promote and Restock	Promote and Restock	0
3ODWSZ5VVV	6	0	0	0	0	0	-1.73234	-0.46582	Promote and Restock	Promote and Restock	0

Figure 8: Snapshot of Final Enriched SKU Dataset

3.5 Weekly Order Forecasting for Amazon Seller Central data

The final output consists of a time series combining 17 weeks of historical order data with 8 weeks of SARIMA-based forecasts. Historical weekly orders ranged from 1 to 15 units, showing fluctuations without a clear seasonal trend. The forecasted period spans from mid-June to early August 2025, with predicted order volumes increasing from approximately 13 to a peak of over 30 before stabilizing around 27 units. The chart displays this transition clearly, with solid lines indicating actual data and a dashed orange line marking forecasted values. All data points follow a consistent weekly interval, anchored on the week's starting date.

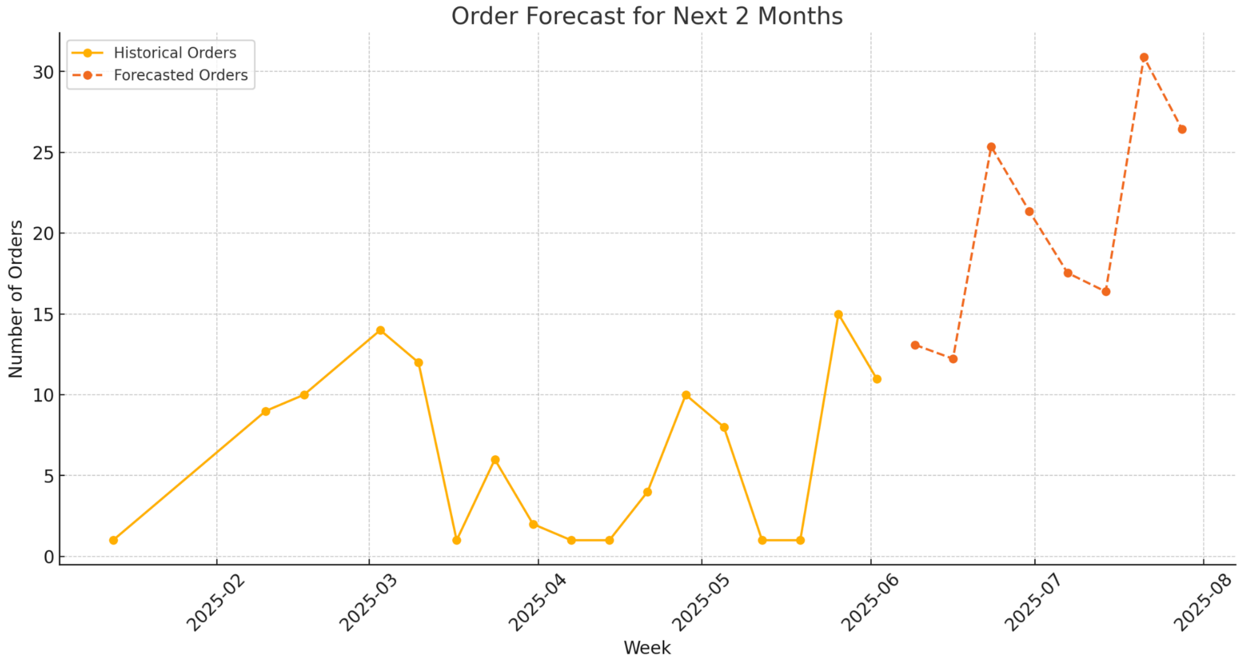


Figure 9: Forecasted Weekly Order Trend with Historical Comparison (Jan–Aug 2025)

4. Interpretation of Results and Recommendations

This section outlines high-confidence interpretations and operational recommendations based on analytical outputs. All suggestions are grounded in quantitative patterns observed during the analysis.

4.1 Search Query Alignment and Catalog Metadata Optimization

Interpretation: Out of 10 high-frequency user search queries analyzed, only 2 matched catalog SKUs with high semantic similarity (match score ≥ 0.50). The majority of queries, such as *“roti basket,” “chapati container,”* and *“spectacle holder,”* had similarity scores close to or below 0.25. Only 2 products were auto-tagged; all others required metadata improvements or product consideration.

Recommendation 4.1.1: Systematically enrich product titles and metadata with high-frequency search terms currently showing low similarity scores. Use synonym expansion and keyword tagging to bridge alignment gaps.

Recommendation 4.1.2:
Create a standing pipeline to re-evaluate top weekly search queries against the live catalog, flagging SKUs for SEO updates or new product inclusion.

4.2 Funnel Drop-Off and Conversion Leakage

Interpretation:
Of 3,351 sessions recorded, only 42 converted to cart additions (1.25%), 15 reached checkout (0.45%), and just 1 converted to a completed purchase (0.03%). The steepest drop-off occurred at the session-to-cart stage.

Recommendation 4.2.1: Prioritize UX improvements and friction reduction at the browsing-to-cart transition, including quicker product previews, CTA placements, and trust-building mechanisms (e.g., badges, shipping info).

Recommendation 4.2.2: Run controlled A/B tests simulating a 10% uplift in each stage, as modeled, to validate conversion sensitivity and fine-tune high-impact improvements.

4.3 Return Risk Modeling and SKU-Level Labeling

Interpretation:
The logistic regression model successfully flagged SKUs with a return ratio above 15%, even in cases where total order volume was low but refund share was high. This enabled early identification of low-volume, high-loss items.

Recommendation 4.3.1: Integrate return risk labels into pre-restocking checklists, and deprioritize restocking of high-return SKUs unless justified by high-margin potential.

Recommendation 4.3.2: Review packaging, sizing, or product communication for high-risk SKUs and test adjusted versions to reduce post-sale returns.

4.4 Product Segmentation via Behavioral Clustering

Interpretation:
Three clearly separable clusters were identified based on return ratio, refund amount, and order volume. Each cluster represents a strategic SKU group—likely candidates for promotion, caution, or deeper investigation.

Recommendation 4.4.1: Assign differentiated inventory and marketing actions at the cluster level: promote stable performers, review high-risk SKUs, and investigate anomalous cases with low sales but high refund values.

Recommendation 4.4.2: Incorporate cluster insights into catalog dashboards for sourcing and merchandising teams, ensuring SKU strategies are performance-aligned.

4.5 Order Forecasting for Demand Planning

Interpretation: SARIMA forecasts indicated a projected rise in weekly order volume from 13 to 30 units over the 8-week horizon, with stabilization around 27 units. This suggests a gradual growth trend with moderate volatility.

Recommendation 4.5.1: Plan short-term inventory allocation to meet the upper bound of projected demand (~30 units/week), particularly for high-demand SKUs.

Recommendation 4.5.2: Re-evaluate forecasting monthly, incorporating fresh data to capture any shifts due to seasonality, campaign launches, or external trends.

ALL ADDITIONAL DATASETS AND RESOURCES:
https://drive.google.com/drive/folders/1sa3v2cDNYqsaPw0yC4i9FxqZB3CZ26JG?usp=sharing