Research Report: Developing a Sales Prediction Model for Amazon India Garment Sales

Amir Shmaryahu

Bar-Ilan University

Introduction and Objectives

This project focuses on developing a **Classification Prediction Model** based on sales data for women's apparel from the Amazon India platform.

The primary research objectives are:

- **Sales Prediction:** To develop a machine learning model capable of predicting whether a specific product will be sold.
- **Business Insights Generation:** Processing and analyzing the data to draw conclusions about consumption patterns, seasonal trends, and the impact of product characteristics on sales success. This aims to facilitate **more accurate strategic decision-decision-making** across inventory planning, pricing, and targeted marketing.
- The research is based on a comprehensive dataset of real sales from Amazon India, documenting women's clothing transactions over several months.
- The model was developed using advanced **Machine Learning Classification** techniques, incorporating cleaning, normalization, and **Feature Engineering**.
- Beyond the prediction itself, emphasis was placed on **deriving analytical insights** from the data to identify relationships between product attributes (such as color, size, category, season) and the probability of a sale.

Data Sources and Structure

The analytical foundation of the project was built by merging **four main data tables**, each providing a complementary layer to the overall sales picture.

Table Name	*Description*	*Raw Rows Count*
Core Table – Amazon Sale Report	Contains the main internal sales data, including order details, SKU information, and quantities.	128,975
Supplementary Table 1 – International Sales	Includes sales originating outside the country to expand the prediction scope.	37,432
Supplementary Table 2 – Periodic Catalogs	Provides additional information on product catalogs and periodic product attributes.	1,330
Supplementary Table 3 – Sale Report (Inventory & Product Details)	Adds complementary data describing inventory and product specifications.	9,271

Data Preparation

1. Integration and Basic Cleaning

Table integration was performed in the following order:

- **Table 1 (March-June 2022 data)** served as the base table.
- **Table 2** was sorted chronologically (ascending by date), filtered to include only data from March to June 2022, and then injected into Table 1 using the **Date** key.
- **Table 3 (May data)** was merged into the unified table using the **Product Code** key.
- **Table 4** was merged using the **Product Code** key.
- **Result:** A preliminary dataset of **69 columns** with **over 130,000 rows**.

Important Note: Before merging, checks were performed for each key column and cleanups were executed to remove incorrect values, aiming to create the most identical keys possible for the table union.

Dataset / Source	*Columns*
Amazon Sale Report (base)	index, order_id, date, status, fulfilment, sales_channel, ship-service-level, style, sku, category, size, asin, courier_status, pcs, currency, amount, ship-city, ship-state, ship-postal-code, ship-country, promotion-ids, b2b, fulfilled-by, unnamed:_22
International Sale Report	months, customer, rate, gross_amt, date_clean, sku_from_date, sku_new
May Catalog	sku_may, style_id, catalog, category_may, weight, tp1, mrp_old, final_mrp_old, ajio_mrp, amazon_mrp, amazon_fba_mrp, flipkart_mrp, limeroad_mrp, myntra_mrp, paytm_mrp, snapdeal_mrp, tp2, sku_clean_may, sku_id_may, sku_size_may
Stock / Sale Report	sku_code, design_no, stock, category_stock, size_stock, color, sku_id_stock, sku_size_stock

2. Column Organization, Correction, and Normalization

This phase focused on comprehensive data cleaning, normalization, and arrangement:

- **Phase I – Data Cleaning and Basic Preparation:**
  - Key integrity (e.g., order_id and SKU) was verified, and erroneous or duplicate records were removed.
  - Values incorrectly entered into columns were moved to the correct location (e.g., product code in a date column).
  - Columns with empty content or irrelevant information were removed.
  - Column names were standardized (e.g., qty and pcs were unified into a quantity column).
  - Data types were normalized: numbers, texts, and categories.
  - New columns, such as **Price per Unit** (amount / pcs), were calculated.
  - New time-based columns—day of the week, month, season, and weekend flag—were created.

- Literal values were cleaned (e.g., similar colors were unified: "blue" and "dark blue") and sizes were standardized.
- City names were cleaned and unified using the **RapidFuzz** library.
- Columns potentially leading to data leakage (e.g., courier_status) were removed to prevent information from compromising the model.

Feature Engineering ***(file ML3.1)***

- New columns were added to improve the model's predictive capability:
    - Creation of new variables like SKU_ID, SKU_SIZE, and SKU_ID_SIZE.
    - Addition of Boolean (yes/no) variables like has_promotion and fulfilled_by_amazon.
    - Encoding of textual columns to language.
    - Integration of seasons with sales data to identify seasonality patterns.
- Key values were checked for each table, and null data was removed.

- **Summary of this phase:** A clean table of meaningful information was built, comprising **29 columns** and **130,425 rows**.

Column	*Data Type*	*Null Count*	*Unique Count*	*Description*
date	datetime64[ns]	0	91	Transaction or order date
fulfilment	category	0	1	Fulfilment type (e.g., merchant, FBA)
ship-service-level	category	0	2	Shipping service level (standard/express)
category	category	0	4	Product category
size	category	0	9	Product size (e.g., S, M, L, XL)
asin	string	0	7,190	Amazon Standard Identification Number (product ID)
pcs	category	1	6	Quantity of units per order
currency	string	7,798	1	Currency type (e.g., INR)

amount	float64	7,798	1,410	Total transaction amount
ship-state	string	33	47	State name in India
ship-postal-code	Int64	33	9,459	Postal code of destination
ship-country	category	33	1	Destination country
b2b	boolean	0	2	Indicates whether the sale is B2B
day	int64	0	31	Day of month
month	category	0	4	Month of transaction
weekday	category	0	7	Day of week
weekend	bool	0	2	Indicates whether transaction occurred on weekend
design_no	string	221	1,305	Product design or model number
stock	float64	221	293	Available stock quantity
status	category	0	2	Order status (e.g., Shipped/Pending)
fulfilment_by_amazon	bool	0	2	Indicates if fulfilled by Amazon
amazon_channel	bool	0	2	Sales channel indicator (Amazon or others)
ship-city_norm	string	37	234	Normalized city name
has_promotion	bool	0	2	Indicates if order had a promotion
seller_easy_ship	bool	0	2	Whether fulfilled via Easy Ship
category_stock_norm	string	225	17	Normalized product category-stock label
color_norm	string	225	25	Normalized color label
season	category	0	2	Seasonality classification (e.g., Summer/Winter)
unit_price	float64	7,824	748	Unit price per item (amount ÷ pcs)

Data Envelopment Analysis (DEA) *(file ML3.2)*

This phase aimed to examine and understand the data following initial cleaning. It included:

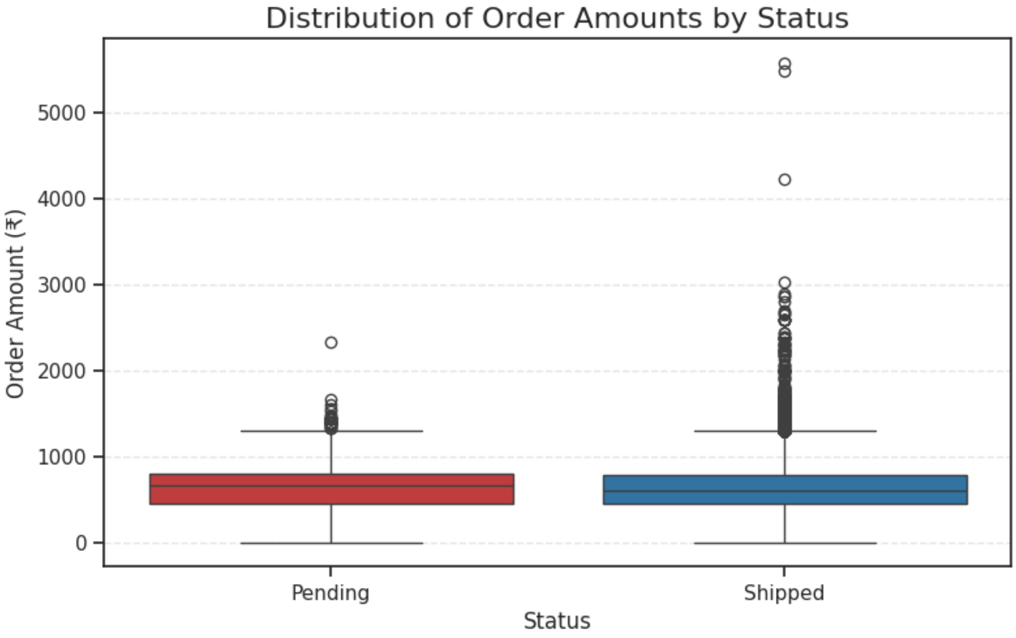
- Loading the clean file, checking the data structure, and verifying integrity.
- Creation of automatic graphs (using autoviz) to display categorical distributions, sales hotspots, and comparisons between promoted and non-promoted products.
- Statistical tests to identify significant relationships between variables.
- Creation of an interactive map showing sales data by cities in India.

- **DEA Objective:** To understand the data from a business and statistical perspective and identify main directions before model building.

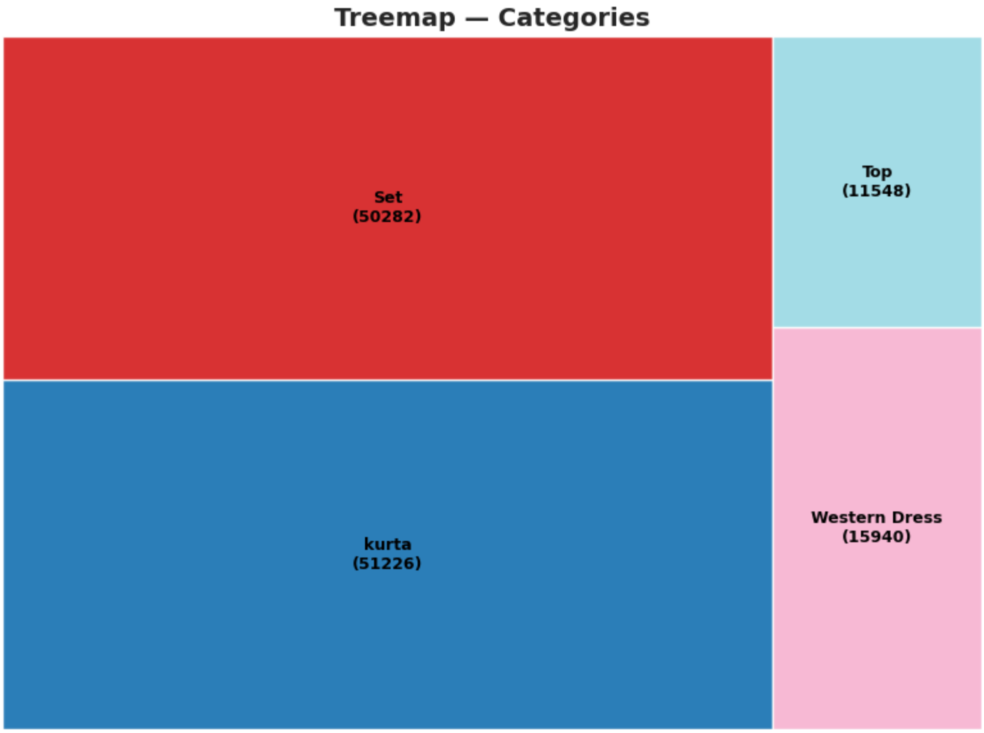
Numeric Columns	*Categorical Columns*	*String Columns*	*Boolean Columns*
amount	fulfilment	asin	b2b
ship-postal-code	ship-service-level	currency	Weekend
day	category	ship-state	fulfilment_by_amazon
stock	size	design_no	amazon_channel
unit_price	pcs	ship-city_norm	has_promotion
—	ship-country	category_stock_norm	seller_easy_ship
—	month	color_norm	—
—	weekday	—	—
—	status	—	—

General Insights from Correlation Analysis:

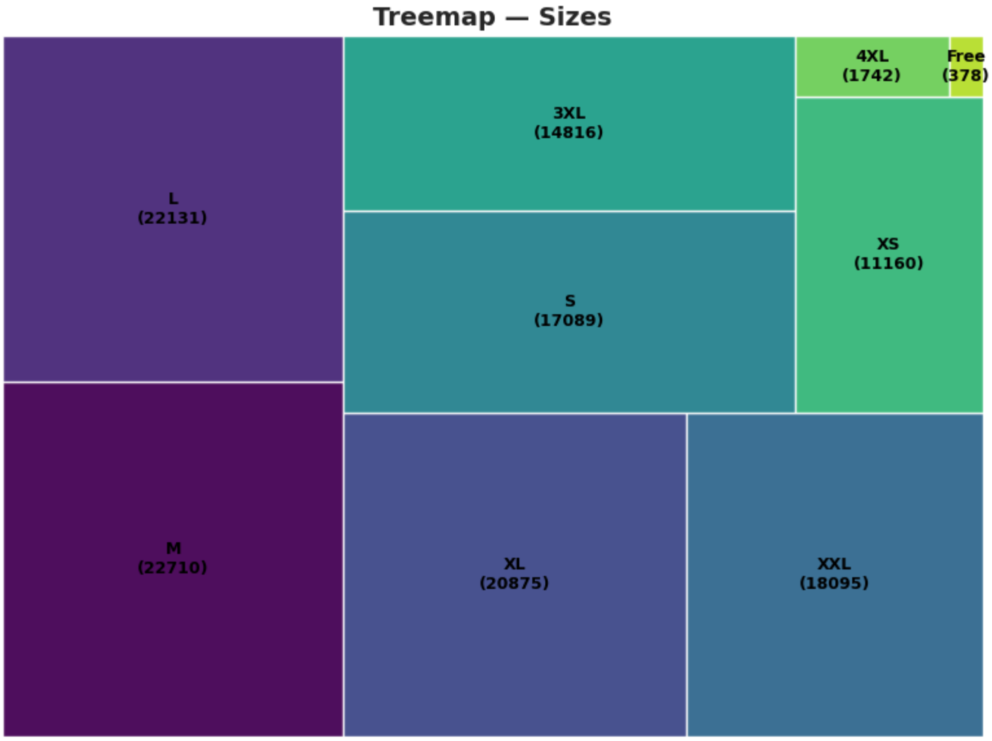
- **Cancellations vs. Sales:** Although the average amount is similar, outliers in non-canceled purchases are lower.



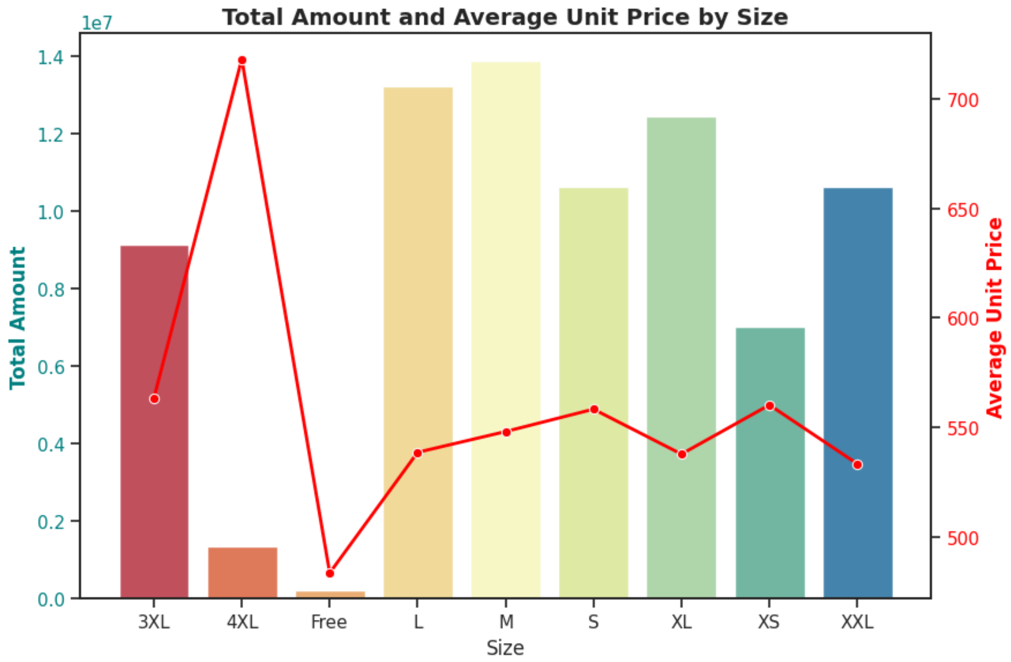
- **Leading Categories:** Most sales are in complementary sets and authentic attire (Set and kurta).



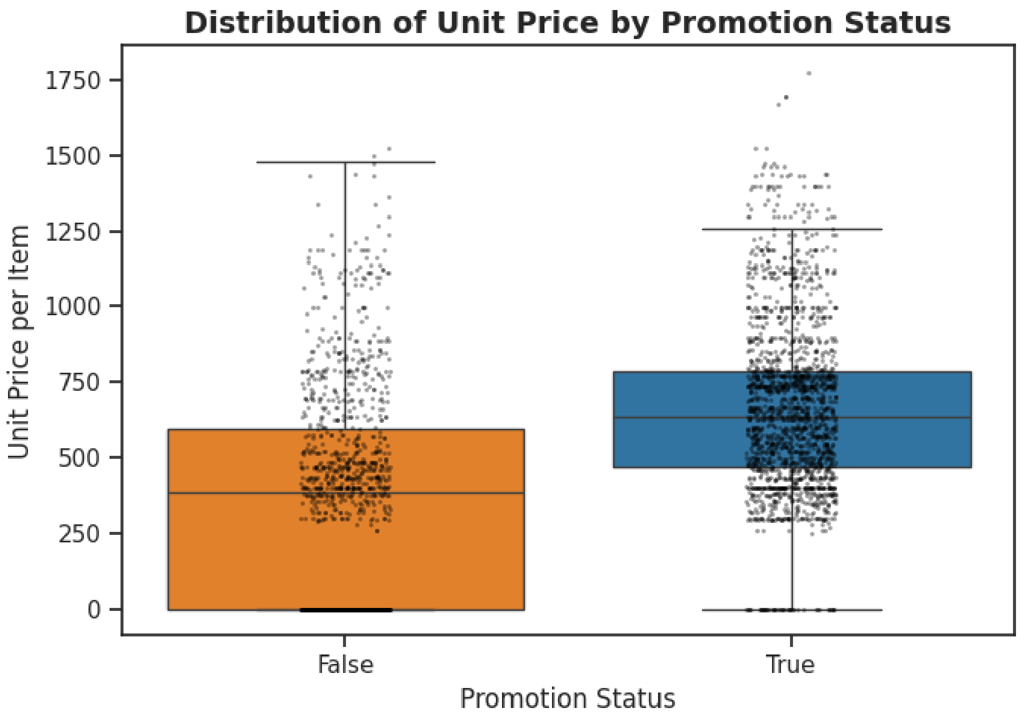
- **Sizes:** Most sizes purchased on Amazon are large sizes (**Large and above**).



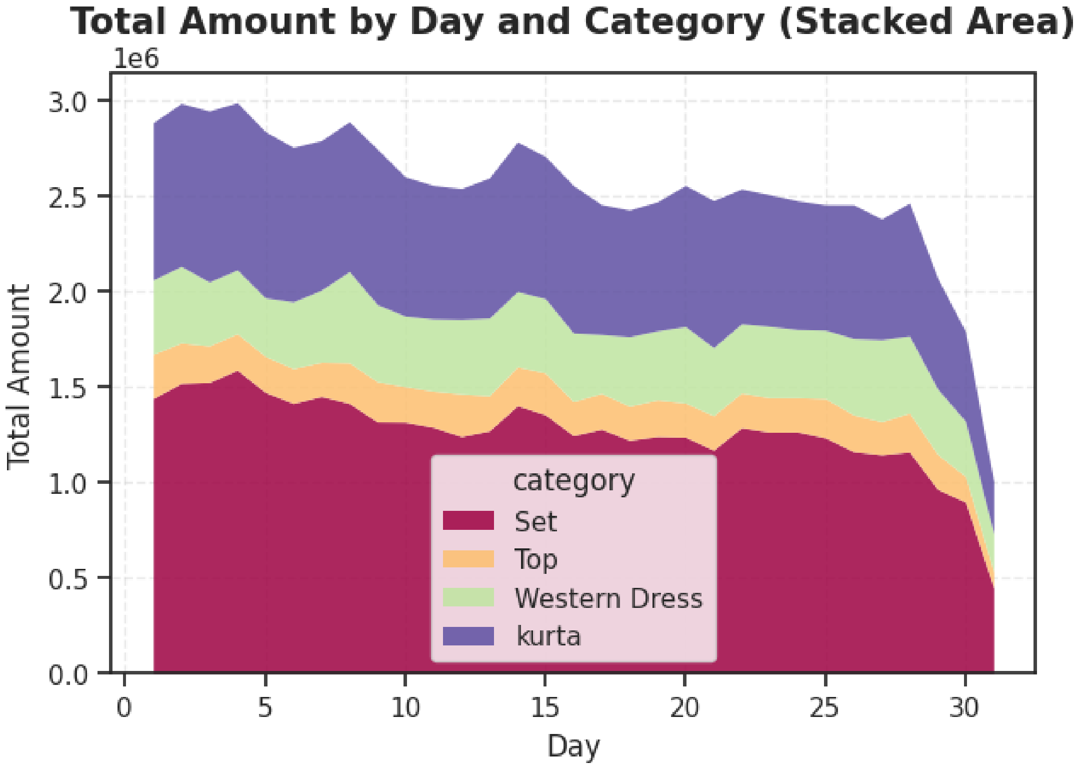
- **Prices by Size:** The ratio of average prices is similar to the purchase price behavior, except for extra-large sizes where the average price is higher (likely due to rarity).



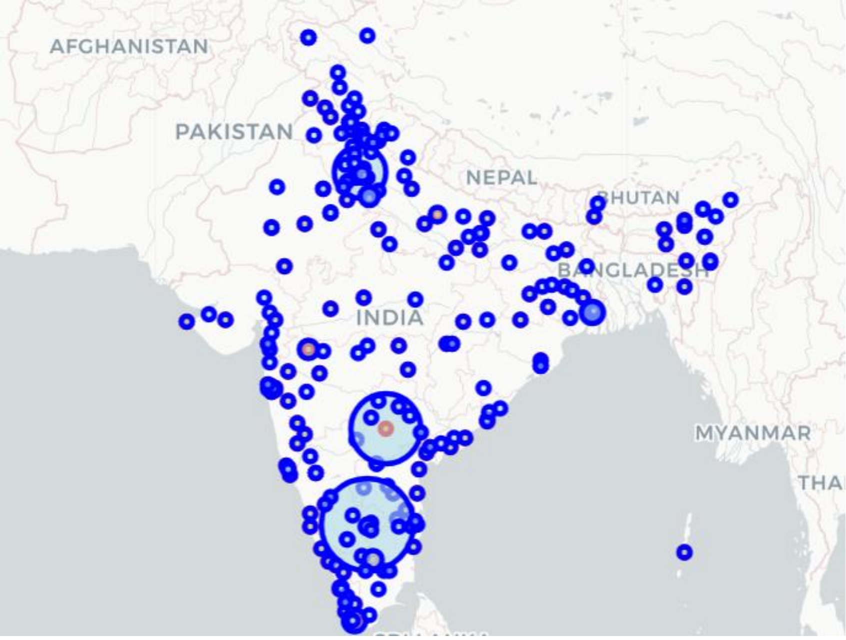
- **Effect of Promotions:** When promotions are active, more quantities are sold, and the average price of sold products is **significantly higher**. When there are no promotions, fewer and cheaper products are bought.



- **Monthly Trends:** Sales volume noticeably decreases towards the end of the month.



- **Geographic Distribution (Cancellations):** No anomalous locations were observed for cancellations, suggesting **no correlation** between location and purchase cancellation.



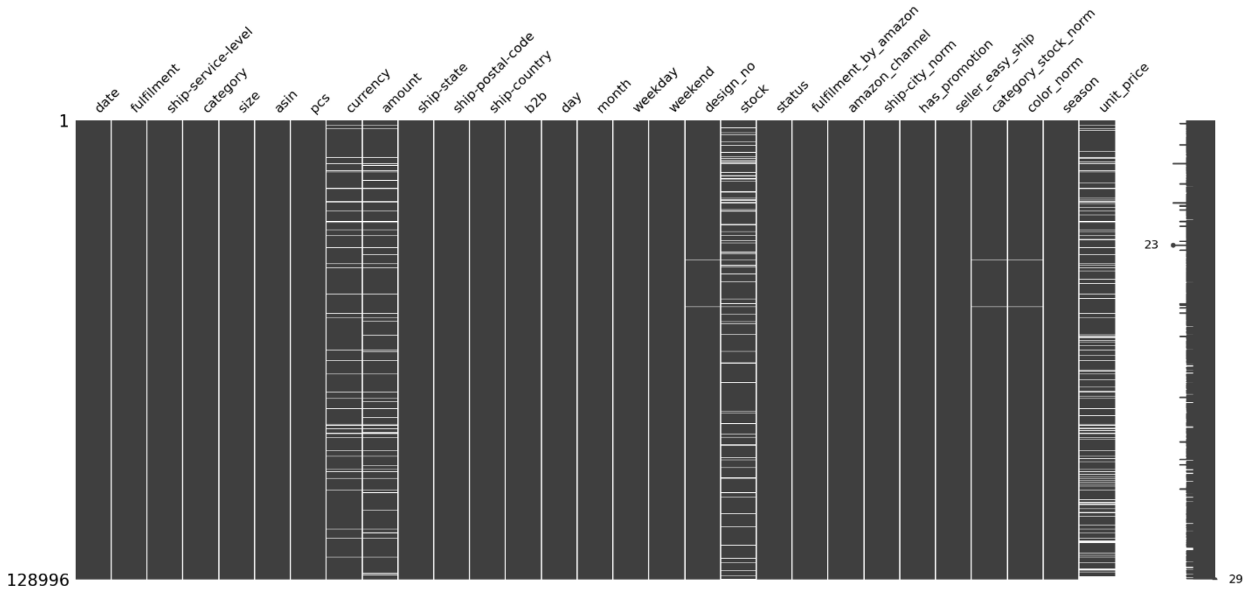
Data Cleansing & Imputation *(Advanced Cleaning and Completion file ML3.2)*

The goal of this process (ML3.3) was to create a balanced, clean dataset, free of outliers, without missing values, and standardized, suitable for building accurate and stable models.

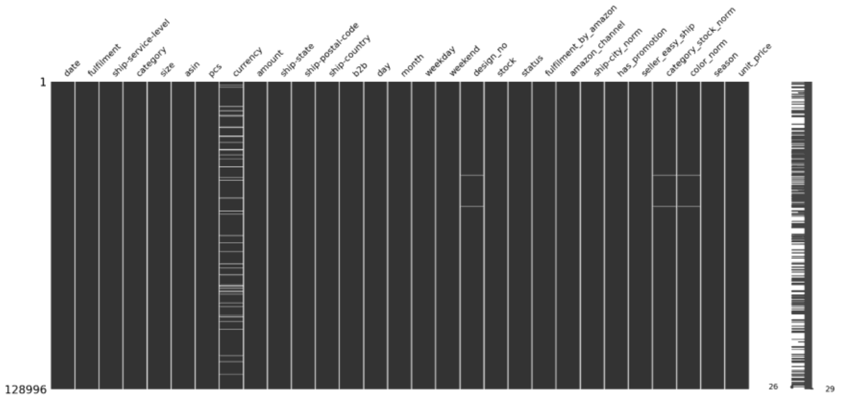
1. Outlier Detection and Treatment

- **Continuous Variables Identification:** Only continuous (numerical) variables were selected and uniformly converted to float64.
- **Graphical Analysis:** A **Boxplot** was created for each numerical variable to identify outliers.
- **Statistical Testing:** Four central variables were chosen: amount, gross_amt, stock, and unit_price. A function was defined to calculate the **IQR** (Interquartile Range) for outlier identification.
- **Tests Performed: KS Test** (Kolmogorov–Smirnov) and **Point-Biserial** correlation were used to measure the impact of outliers on distribution and correlation with the target variable (status: Shipped / Pending). Features showing a change in distribution but no change in correlation were marked for removal, as their outliers represented only statistical noise.

2. Imputation of Missing Values

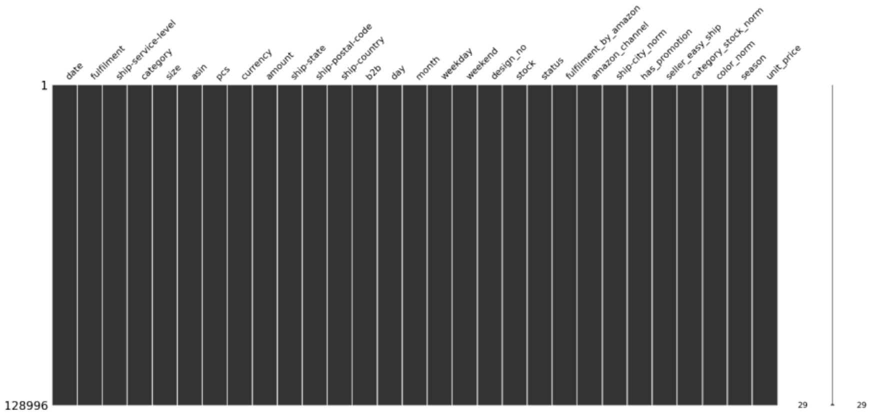


- **Numerical Missing Values:** The **MICE** (Multiple Imputation by Chained Equations) method was used via IterativeImputer from scikit-learn. After imputation, a lower bound (**clip**) was applied to prevent negative values.



- **Non-Numerical Missing Values:**
  - Categorical columns: A new category, **"unknown,"** was added to fill missing values.
  - Textual columns: Completed with the value **"unknown"**.
  - Boolean columns: Completed with **False**.

- **Final Cleaning of the Target Variable (status):** Rows where status == 'unknown' were completely deleted and checks using missingno.matrix confirmed that no missing values remained.



- **Final Step:** After all cleaning, filtering, and imputation steps were completed, the final dataset was saved to the file final_df.ML3.3.pkl.

Feature Selection and Predictive Modeling *(file ML3.4)*

This phase aimed to build an accurate prediction model capable of forecasting operational and business variables, such as shipment success, returns, or order status.

1. Encoding and Feature Selection

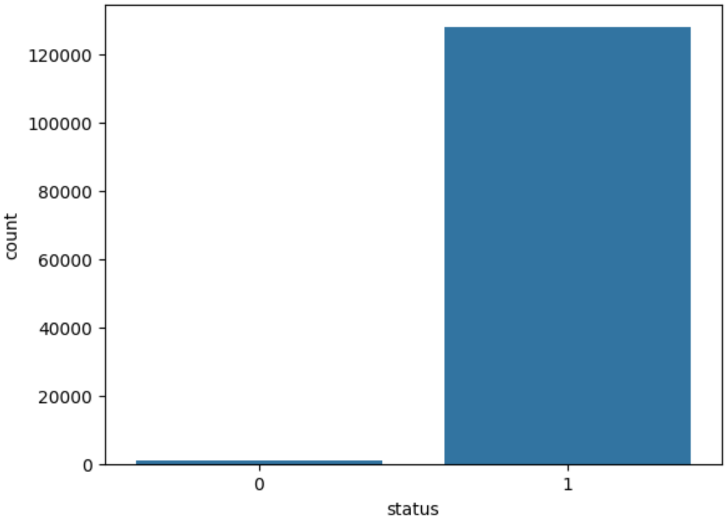
- **Encoding:** All textual columns were converted to numerical values using **Label Encoding** to enable the model to learn from them.
- **Feature Selection:** Only the most significant variables were retained, selected by **at least five** different models from the following methods:
  - Lasso Regression (L1)
  - Ridge Regression (L2)
  - SVM (L1 penalty)
  - Decision Tree / Random Forest / Gradient Boosting / AdaBoost (using Feature Importance metric)

Feature	*Lasso*	*Ridge*	*SVM*	*GradientBoost*	*RandomForest*	*LogisticRegression*	*DecisionTree*	*AdaBoost*	*Sum*
ship-postal-code	1	0	1	1	1	1	1	1	7
pcs	0	1	1	1	1	1	1	1	7
currency	0	1	1	1	1	1	1	1	7
amount	1	0	1	1	1	1	1	1	7
stock	1	0	1	1	1	1	1	1	7
weekday	0	0	1	1	1	1	1	1	7
day	1	0	1	1	1	1	1	1	7
design_no	1	0	1	1	1	1	1	1	7
size	0	0	1	1	1	1	1	1	6
asin	1	0	1	1	1	1	1	0	6
unit_price	1	0	1	1	1	1	1	0	6
has_promotion	0	0	1	1	1	1	1	1	6
season	0	0	1	1	1	1	1	1	6
ship-city_norm	0	0	1	1	1	1	1	1	6
color_norm	0	0	1	1	1	1	1	0	5

ship-state	0	0	1	1	1	1	1	0	5
category_stock_norm	0	0	1	1	1	1	1	0	5
month	0	0	0	1	1	1	0	1	4
category	0	0	1	1	1	1	0	0	4
seller_easy_ship	0	0	0	1	1	1	1	0	4
fulfilment_by_amazon	0	0	0	1	1	1	1	0	4
weekend	0	0	1	0	1	1	1	0	4
ship-service-level	0	0	0	0	1	1	0	1	3
amazon_channel	0	0	1	0	1	1	0	0	3
ship-country	0	0	0	0	1	1	0	0	2
b2b	0	0	0	0	1	1	0	0	2
fulfilment	0	0	0	0	0	0	0	0	0

2. Class Imbalance

The target variable quantity was found to be at a ratio of **less than 15%** between positive and negative.



After testing four data completion options (oversampling/undersampling) and comparing prediction results, the decision was made **to use the data as is** (without artificial balancing) as this method achieved a better result.

Technique	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Original	**0.999612**	**0.999714**	**0.999896**	**0.999805**
ROS	0.716390	0.998112	0.715662	0.833611
RUS	0.696080	0.997834	0.695359	0.819579
SMOTE	0.662774	0.995353	0.663396	0.796158
SMOTETomek	0.728560	0.998108	0.727947	0.841885

3. Data Splitting and Modeling

The data was split into three parts:

- **Train Set (70%):** For model training.
- **Dev Set (15%):** For model and hyper-parameter selection.
- **Test Set (15%):** For final performance evaluation.

4. Selection of the Winning Model

We ran a forecast on several models, and the results we received are described in the table below.

Model	*Accuracy*	*Precision*	*Recall*	*F1*	*ROC_AUC*	*LogLoss*
LogisticRegression	0.992403	0.992403	1.000000	0.996187	0.781278	0.042093
LinearSVC	0.992403	0.992403	1.000000	0.996187	0.603597	0.273690
DecisionTreeClassifier	0.999328	0.999479	0.999844	0.999662	0.965908	0.024215
Decision Tree	0.999225	0.999479	0.999740	0.999609	0.965856	0.027941
RandomForestClassifier	0.999483	0.999532	0.999948	0.999740	0.992855	0.005692
AdaBoostClassifier	0.998863	0.999167	0.999688	0.999427	0.994659	0.339034
GradientBoostingClassifier	0.999483	0.999532	0.999948	0.999740	0.997526	0.002640
XGBClassifier	0.999380	0.999427	0.999948	0.999688	0.997342	0.003227

The **GradientBoostingClassifier** and **RandomForestClassifier** models achieved very similar results.

Criterion	*Best Metric / Model*
Accuracy	RandomForestClassifier = 0.999483 ≈ GradientBoostingClassifier = 0.999483
Precision / Recall / F1	GradientBoostingClassifier and RandomForestClassifier ≈ 0.9997
ROC_AUC	GradientBoostingClassifier = 0.997526 (highest)
LogLoss	GradientBoostingClassifier = 0.002640 (lowest, indicating highest prediction confidence)

Gradient Boosting was chosen as the winner due to:

- Displaying the **highest ROC_AUC**, meaning better discrimination between classes.
- The **lowest LogLoss**, indicating more stable and confident predictions.
- Accuracy identical to RandomForest, but with lower variance and better tuning capability.

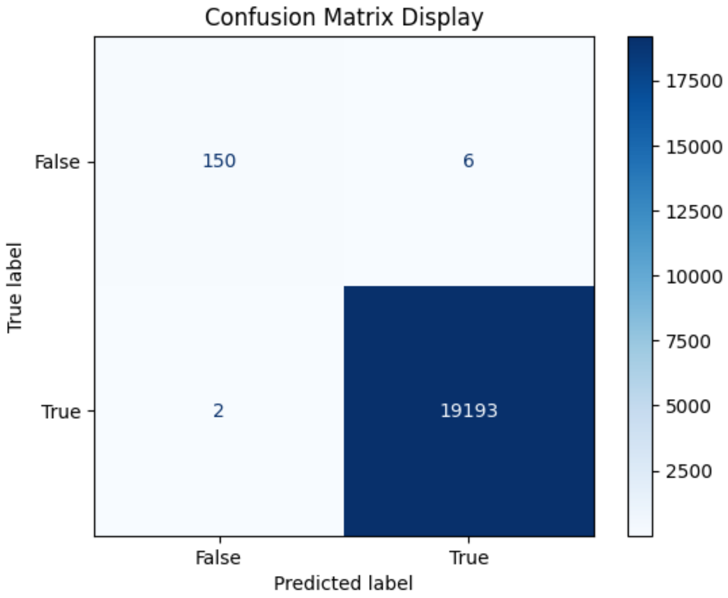
5. Hyper-parameter Tuning

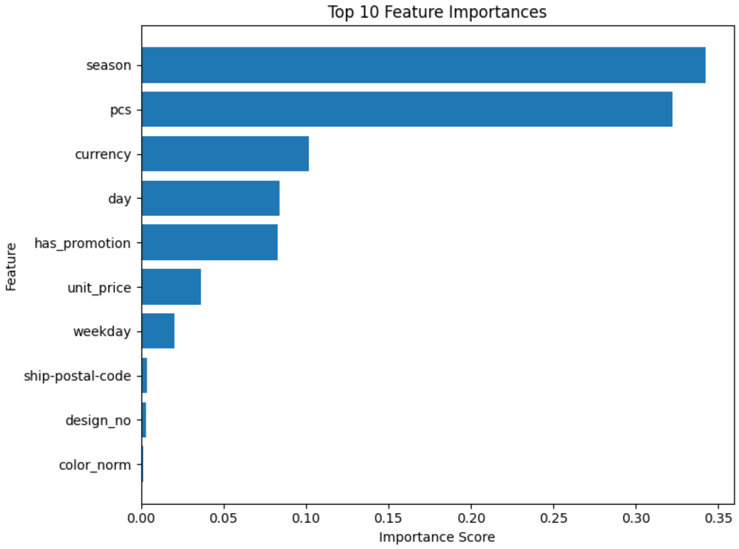
After model selection, a comparison was made between running the model with default parameters and running it with parameters suggested by **RandomizedSearchCV**. Results showed a **1% improvement** in favor of using the default parameters.

Conclusions and Business Insights

1. Excellent Model Performance:

An R2 value of 0.9483 indicates that the model explains approximately **95% of the variance** in the data—a very strong result suggesting high predictive capability. The **Accuracy** is nearly perfect in both sets (≈1.00). **Class 1 (Shipped)** is classified almost flawlessly (Recall = 1.00). **Class 0 (Pending)** is slightly less accurate (Recall ≈0.94–0.96). The model is **stable** with no clear overfitting between the development and test sets. The low LogLoss (0.002640) suggests high confidence in its forecasts.





2. Dominant Variables:

season and **pcs (quantity)** are the two most influential variables—together, they account for **over 66%** of the total contribution to the model. **Interpretation:** Seasonality and the number of units are key factors in determining sale status or performance.

3. Secondary and Marginal Factors:

- **Secondary factors:** currency, day, and has_promotion have a medium impact.
- **Marginal factors:** ship-postal-code, design_no, and color_norm have a very low impact, meaning that geographic location or design color do not significantly contribute to the prediction.

4. General Interpretation:

The model is accurate and interpretable: it accurately identifies the most influential factors (mainly season and quantity), maintains good generalization, and is highly suitable for business or operational forecasting applications.

Model Evaluation Summary

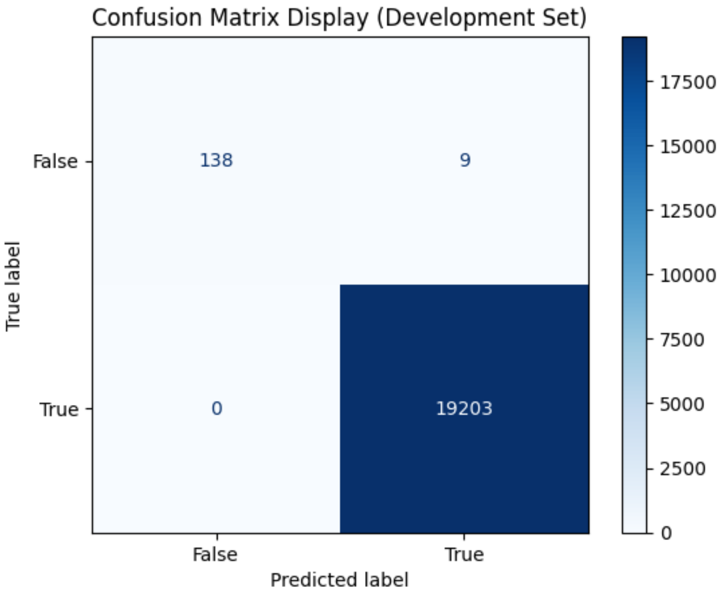
Metric	*Precision (0)*	*Recall (0)*	*F1-Score (0)*	*Precision (1)*	*Recall (1)*	*F1-Score (1)*	*Accuracy*	*Macro Avg F1*	*Weighted Avg F1*
Test Set	0.99	0.96	0.97	1.00	1.00	1.00	1.00	0.99	1.00
Dev Set	1.00	0.94	0.97	1.00	1.00	1.00	1.00	0.98	1.00

Confusion Matrix (Test Set)

Actual \ Predicted	*0*	*1*
0	150	6
1	2	19193

Confusion Matrix (Development Set)

Actual \ Predicted	*0*	*1*
0	138	9
1	0	19203



- Accuracy is nearly perfect in both sets (≈ 1.00).
- Class 1 (Shipped) is classified almost flawlessly (Recall = 1.00).
- Class 0 (Pending) is slightly less accurate (Recall ≈ 0.94–0.96).
- The model is stable with no clear overfitting between Dev and Test.

Final Conclusion: Gradient Boosting Model as a Management Tool

The model allows management, inventory, and marketing teams to use the forecast not only to know **if** a product will be sold, but primarily to understand **why** it will be sold. The **Gradient Boosting** model provides the optimal combination of accuracy, stability, and generalization capability, making it the final model chosen for the AmazonProject-ML.

The analysis highlighted three key factors for sales success: **Seasonality**, **Unit Quantity**, and **Price/Promotion**.

1. Seasonality-Based Optimization (Season - Over 33% of Impact)

The fact that **Season** is the most influential variable indicates a critical need for dynamic inventory management and pricing aligned with the calendar.

Key Insight	*Recommended Operational Action*
Seasonal Sales Peaks Identification	Map seasonal high-demand periods (e.g., holidays or festival seasons in India) and significantly increase marketing budgets and stock depth in advance.
Old Inventory Prevention	Products unsold during their season (Off-Season Items) should be dynamically repriced or included in clearance promotions promptly.
Catalog Planning	Align product designs and displayed categories in catalogs with temperature trends and seasonal consumer preferences.

2. Inventory and Quantity Strategy (PCS - Over 33% of Impact)

The variable **pcs** (number of units purchased in an order) is another critical factor. The insight that successful sales are related to a specific unit quantity or the ordering of **complementary sets/authentic attire** (as found in the DEA analysis) necessitates adjusting the sales strategy.

Key Insight	*Recommended Operational Application*
Encouraging Multiple Purchases	Implement an aggressive Cross-Selling and Bundling strategy, promoting complementary product packages as the primary offer instead of individual units.
Level-Based Inventory Management	Monitor inventory levels closely, especially for items sold as part of a set, to prevent out-of-stock components from halting larger transactions.
Handling Large Sizes	Since most purchases are size L and above, allocate more inventory and pricing focus to these sizes, as they are both less common and highly demanded.

3. Pricing, Promotions, and Purchase Timing (Promotion, Currency, Day)

The model confirms the positive effect of promotions (Has Promotion) on sales amount and quantity sold.

Key Insight	*Recommended Operational Application*
Power of Promotions	Use promotions strategically to raise the **Average Order Value (AOV)**, not only for stock clearance, since higher-priced products tend to sell more during promotional periods.
Marketing Timing	The observed drop in sales near month-end calls for intensified marketing efforts (e.g., “End-of-Month” campaigns) or reallocating part of the marketing budget to the start of the month.
Currency Understanding	Analyze the currency variable to determine whether it reflects international sales or rare currencies, and adjust exchange rates and transaction fees accordingly.

Reuse of Marginal Data:

Variables found to have a **low impact** (color_norm, ship-postal-code, design_no) may suggest that significant resources should not be invested in **cleaning** this data in the future, or they can be used as a basis for more focused marketing analysis (e.g., tailoring colors to specific regions despite the low impact in the overall model).

The cleaning and processing of data in the previous stages led to an exceptionally accurate prediction model. This model is ready for business use for sales status forecasting, logistical performance analysis, and generating data-driven operational insights.