# Data and Business Analytics for BaSalam e-commerce

2025

# TEAM INFO

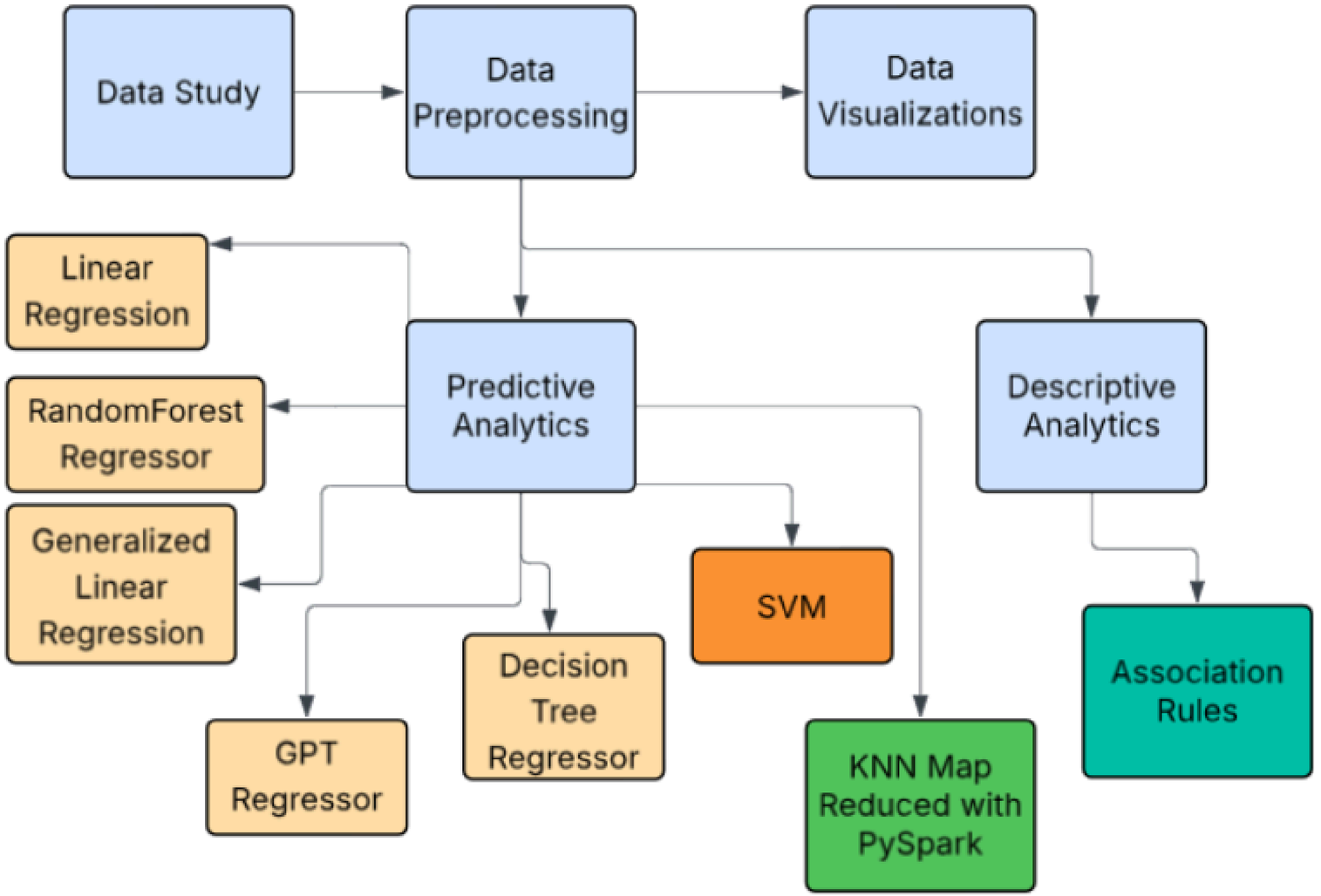
| TEAM 12 | | |
|---|---|---|
| **Name** | **Section** | **BN** |
| Moustafa Mohammed Elsayed | 2 | 23 |
| Mennatallah Ahmed Moustafa | 2 | 25 |
| Mostafa Hani Mostafa | 2 | 24 |
| Mohammad Alomar | 2 | 14 |

# PROJECT GOALS

1. Predict product performance using factors such as price, customer ratings, and weekly sales.

2. Analyze historical data to identify trends and insights influencing product success.

# PROJECT PIPELINE



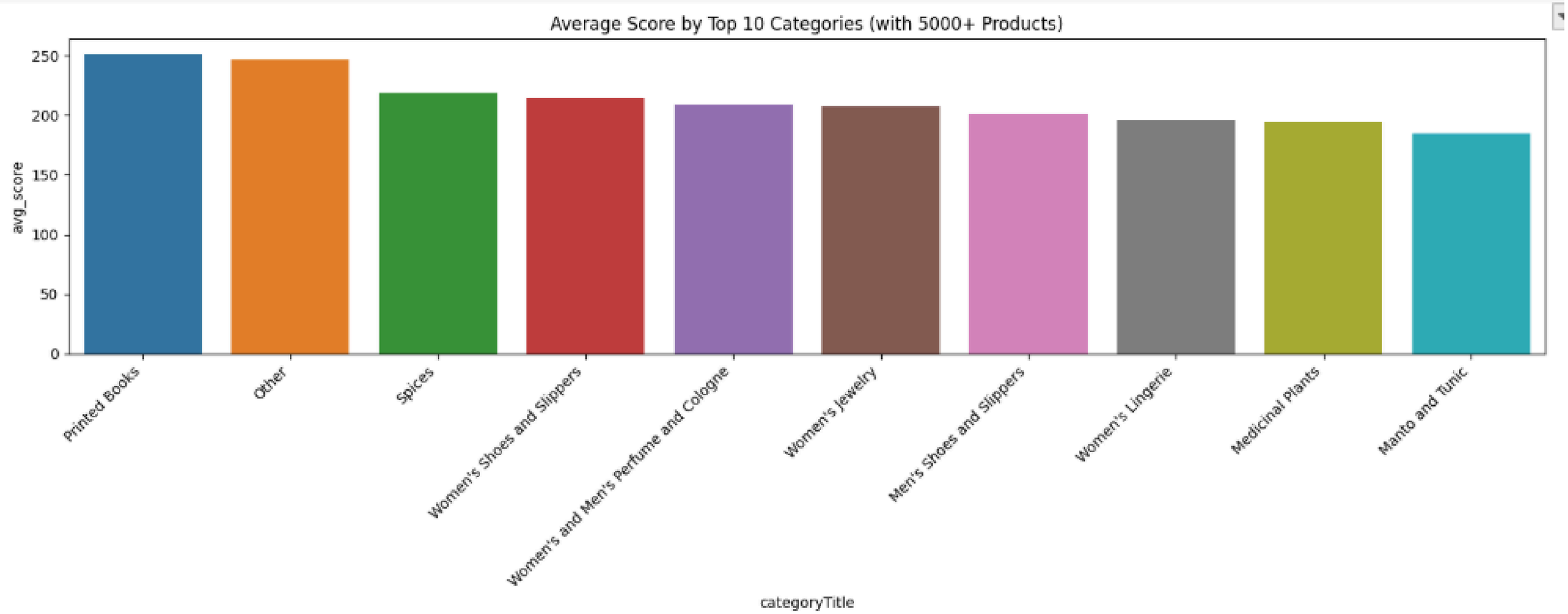
# Detailed Analysis

Business Analysis

Technical Analysis

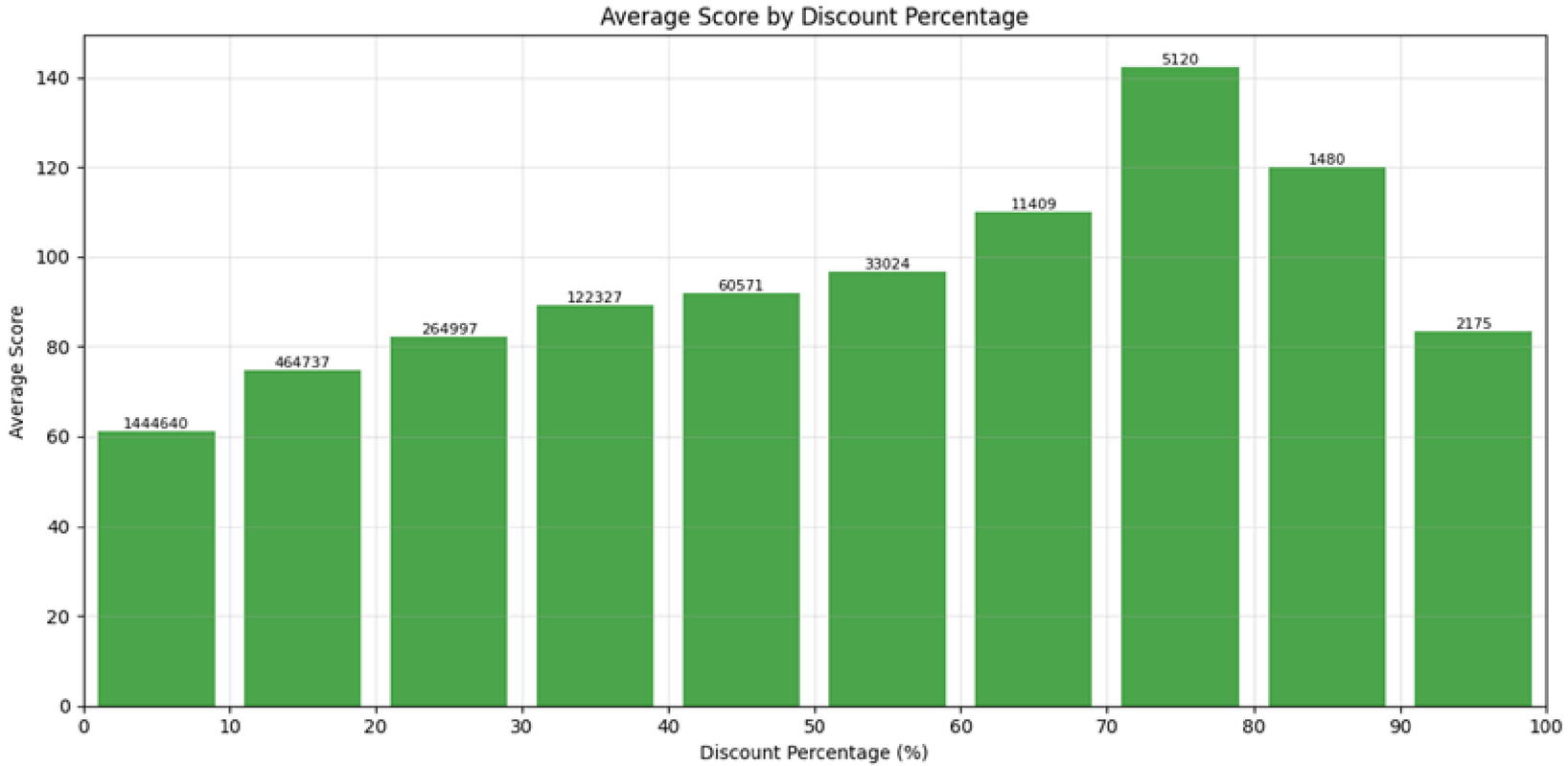
# CATEGORY ANALYSIS

- Having a product in one of these categories leads to better score



# DISCOUNT ANALYSIS

- Increasing discounts make products appear more valuable "good deal"
- **Maximum satisfaction** around 70-80% discounts
- Extreme discounts (especially 90-100%) trigger skepticism:
  - Customers suspect products being defective/expired, "Too good to be true" mentality



# VARIATION ANALYSIS

- The average score increases when the product has variations of different sizes/colors

| has_variation | average_score |
|---|---|
| false | 68.169 |
| true | 90.199 |

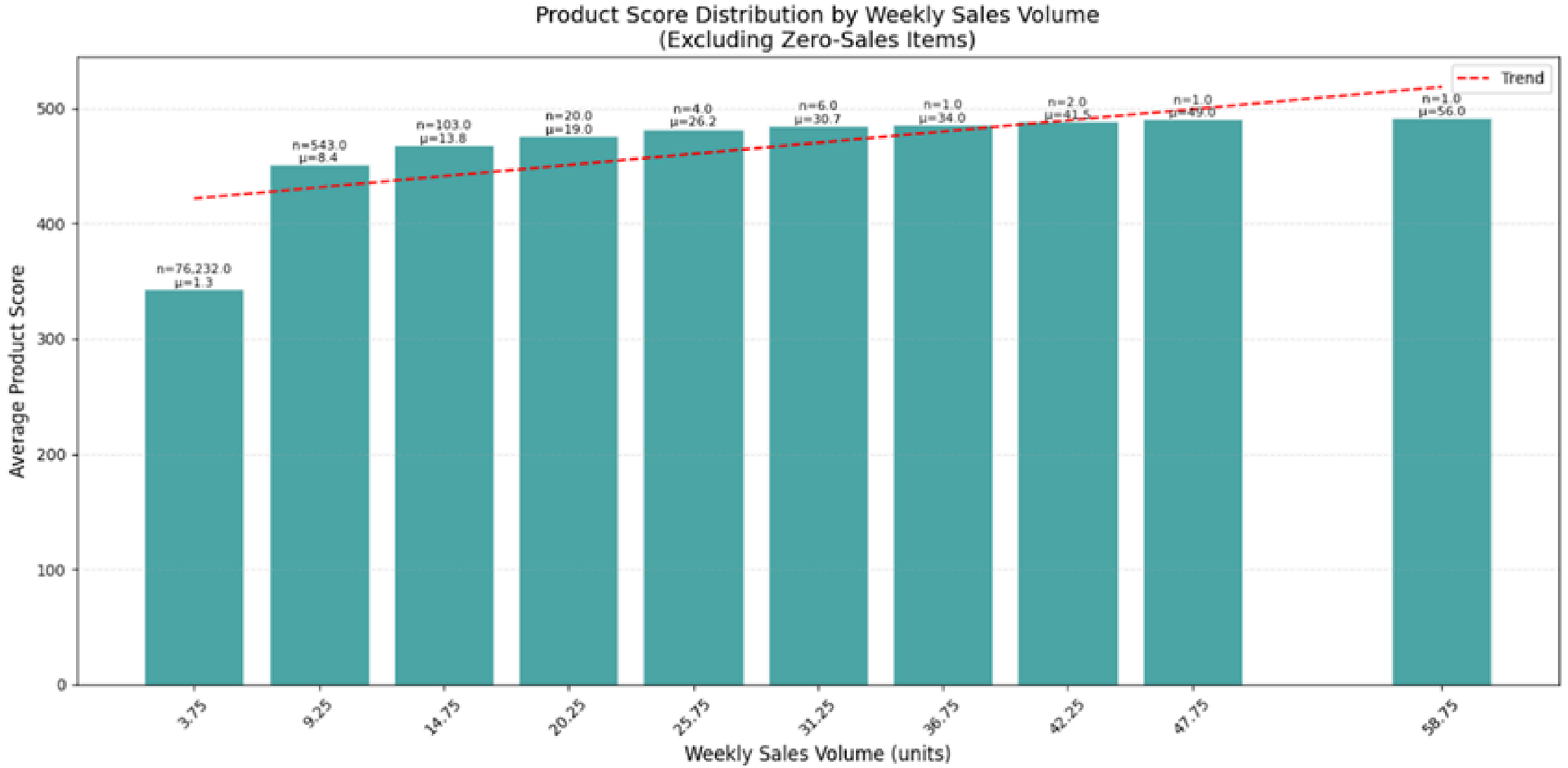
# WEIGHT VS FREESHIPPING ANALYSIS

- High weight means high price which probably has Free shipping

| isFreeShipping | average_weight | average_price |
|---|---|---|
| false | 2639.84 | 2719.07 |
| true | 3182.76 | 7034.415 |

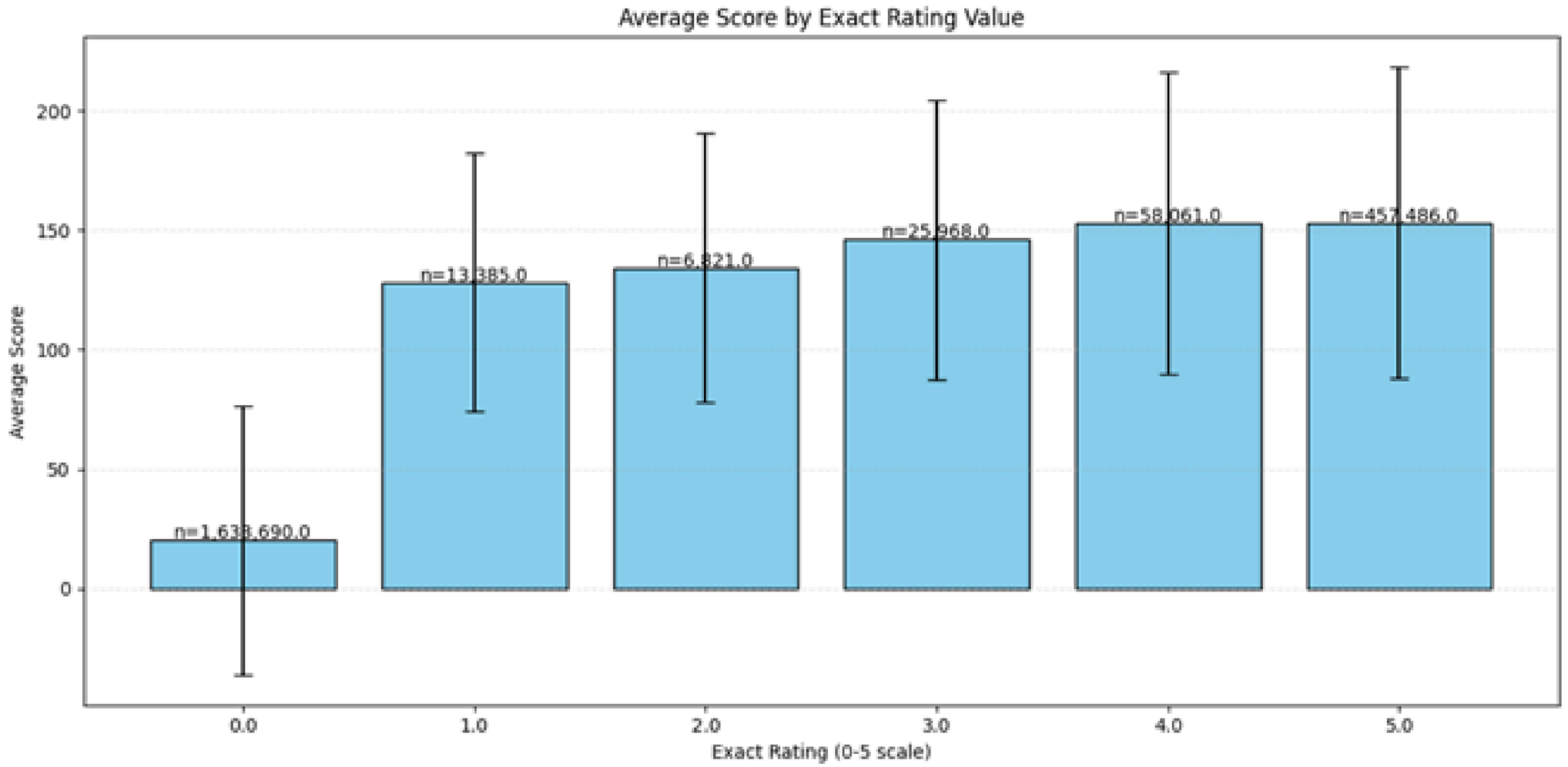
# SCORE VS SALES ANALYSIS

- Higher weekly sales correlate with better performance scores.



# SCORE VS RATING AVERAGE  ANALYSIS

- Better ratings strongly lead to higher scores.



# MEDIA VS HIGH SCORE ANALYSIS

- Having a picture and a video for the product achieves the highest scores

| has_photo | has_video | average_product_score | product_count |
|---|---|---|---|
| false | false | 18.22 | 524 |
| false | true | 13.88 | 6 |
| true | false | 68.5869 | 2134613 |
| true | true | 89.282 | 150089 |

# HOW TO HAVE A GOOD PRODUCT?

1. Select products from categories that consistently have high performance scores.
2. Apply moderate (not excessive) discounts to attract buyers without devaluing the product.
3. Offer variations (such as different sizes, colors, or styles) to appeal to a broader audience.
4. Prioritize maintaining high customer ratings by ensuring quality and excellent service.
5. Provide high-quality images and videos to showcase the product effectively.

# Technical Analysis

Data Preprocessing

Data Visualization

Model Training

# DATA PREPROCESSING

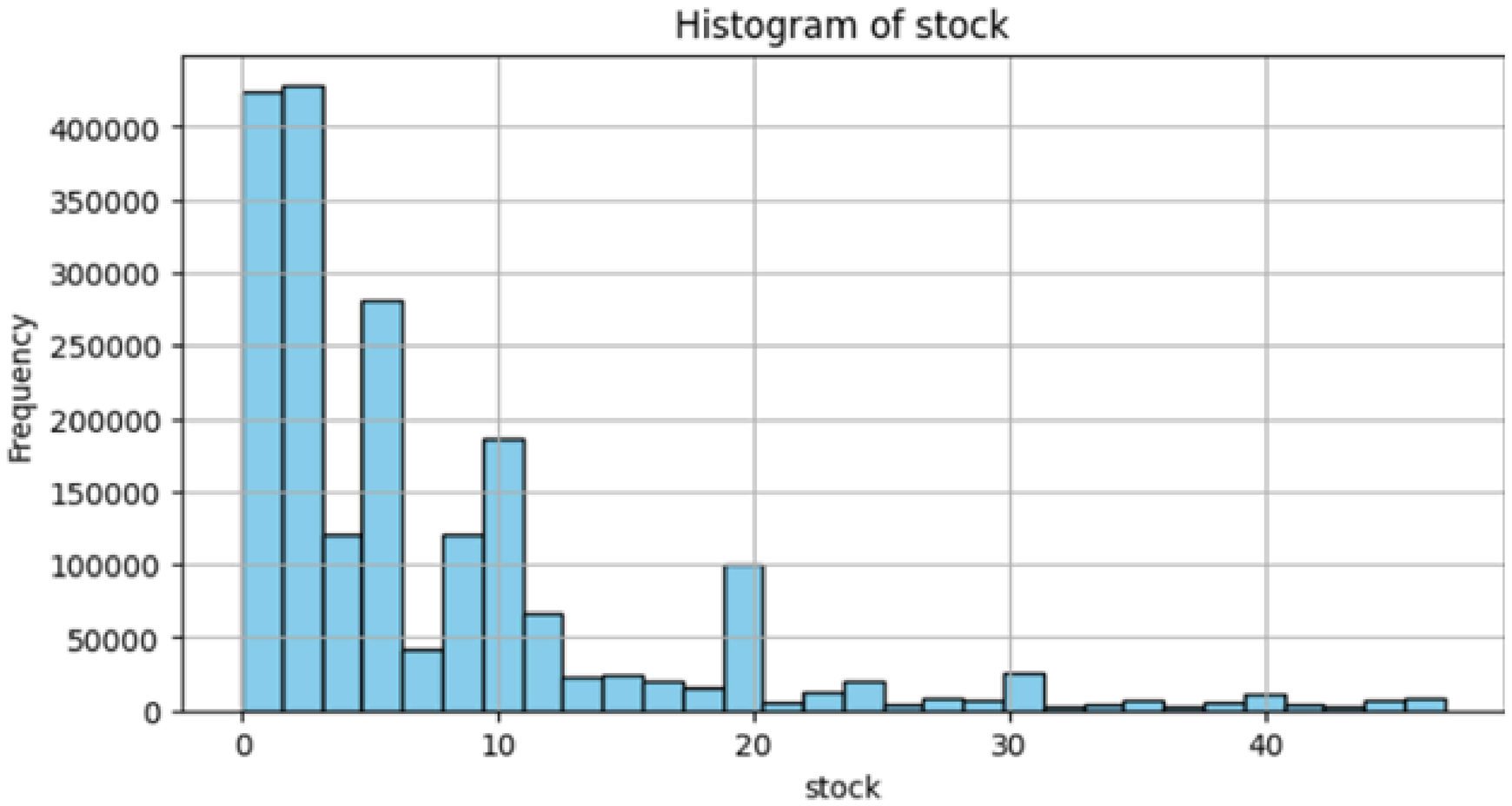
1. Dropping columns with high percent of null values (> 0.8)
2. Splitting data into train, test
3. Removing columns that won’t affect the model such as ids, photos URLs
   ['_id','photo_MEDIUM','vendor_identifier', 'photo_SMALL','vendor_id','vendor_provinceId', 'vendor_owner_id','vendor_status_id','vendor_cityId','vendor_statusId','categoryId','new_categoryId','status_id']
4. For association rules, we used dropna to drop all rows that have nulls after that we would have
   a. Original row count: 2411358
   b. Rows after removing nulls: 2262039
   c. Removed 149319 rows (6.19% of data)

# DATA PREPROCESSING

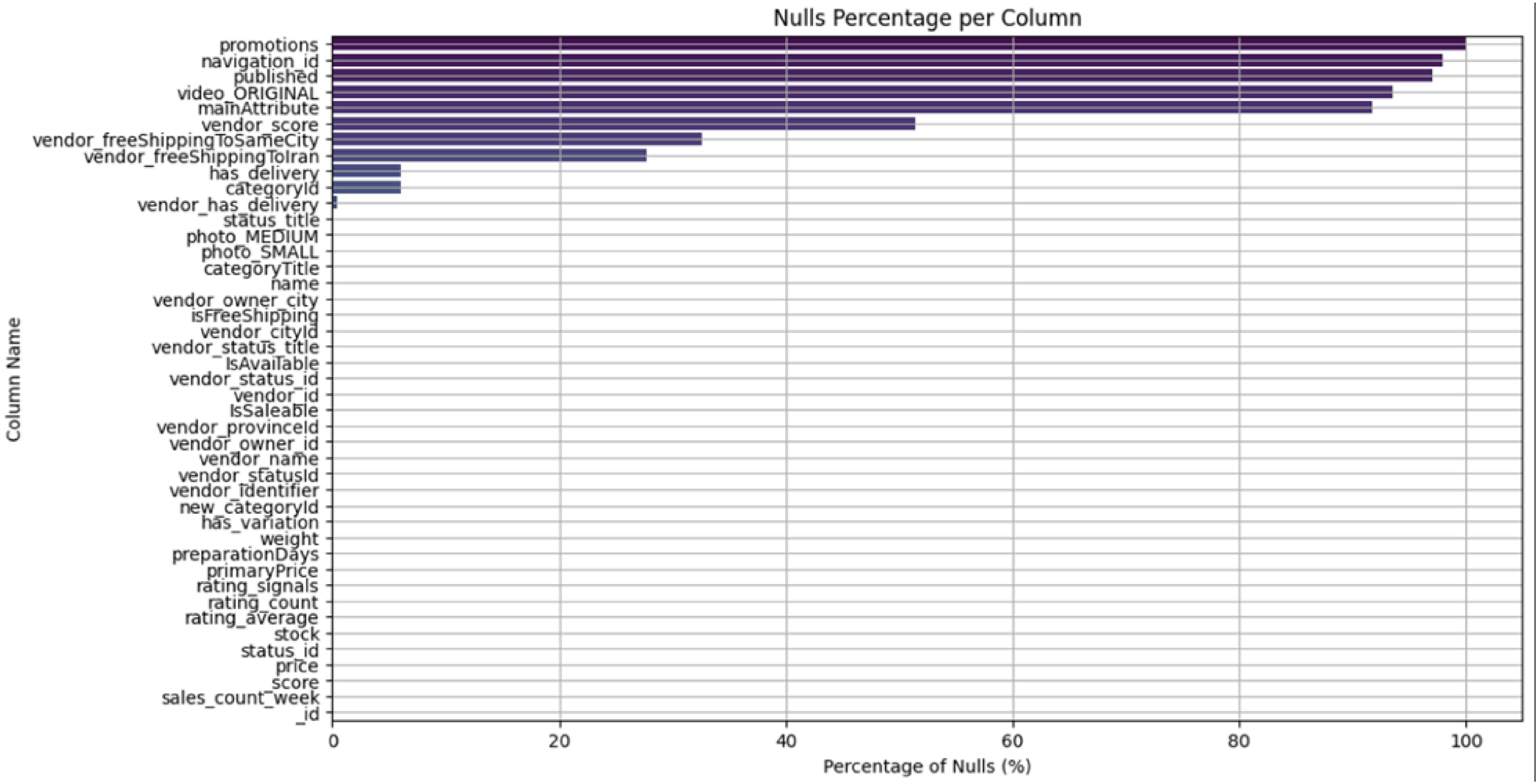
5. But for the predictive models, we found that the remaining columns have a low percent of nulls (0.2) thus we decided to replace these nulls with a median for numerical columns and mode for categorical, binary columns
6. Dropping categorical columns with very high cardinality before one hot encoding
7. One hot encoding categorical columns
8. Drop very high correlated columns
9. Now all columns are numerical, we Calculate the correlation with each column and target, and select the most correlated columns with the target

# DATA VISUALIZATION

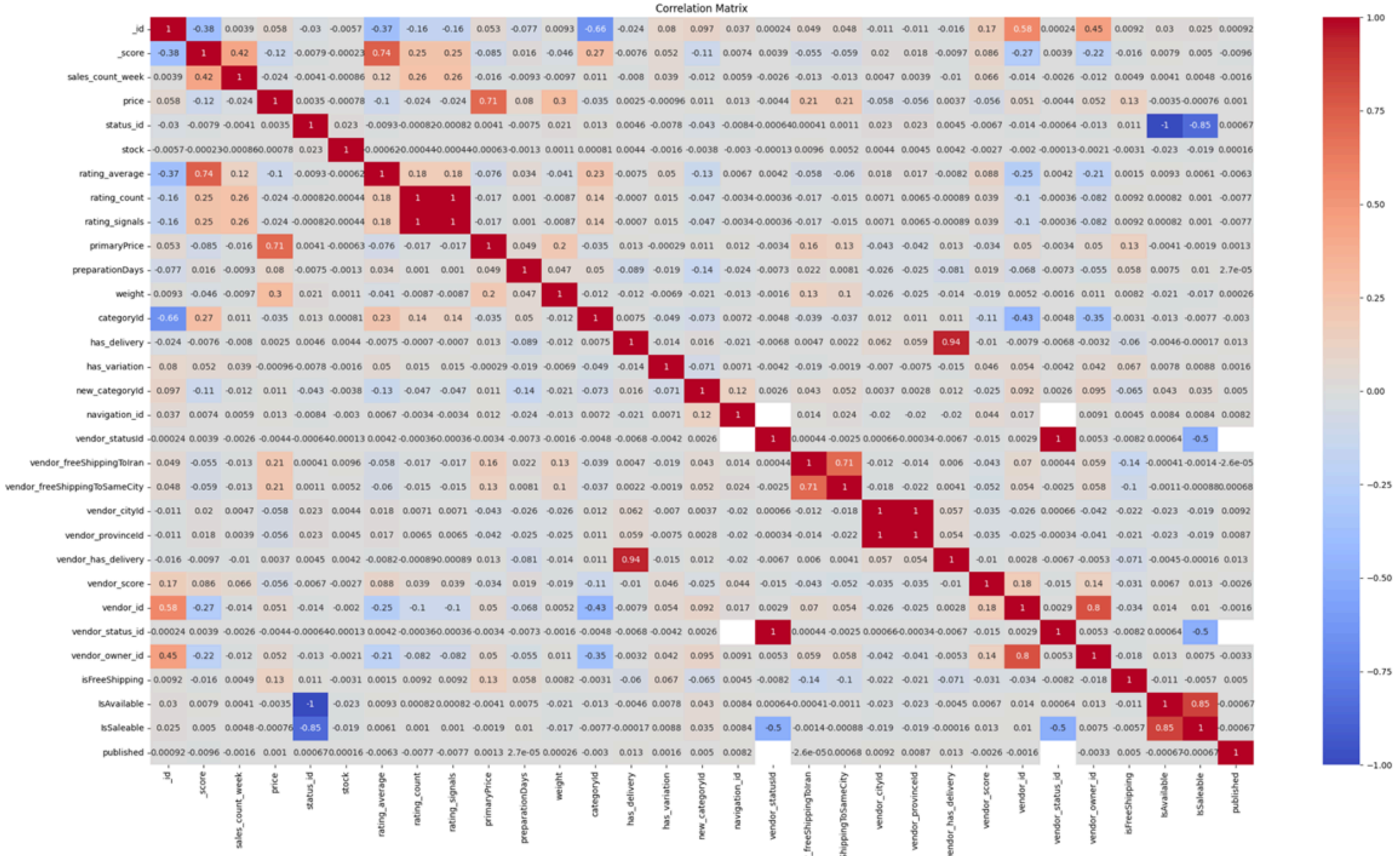
- High stock is infrequent



# DATA VISUALIZATION

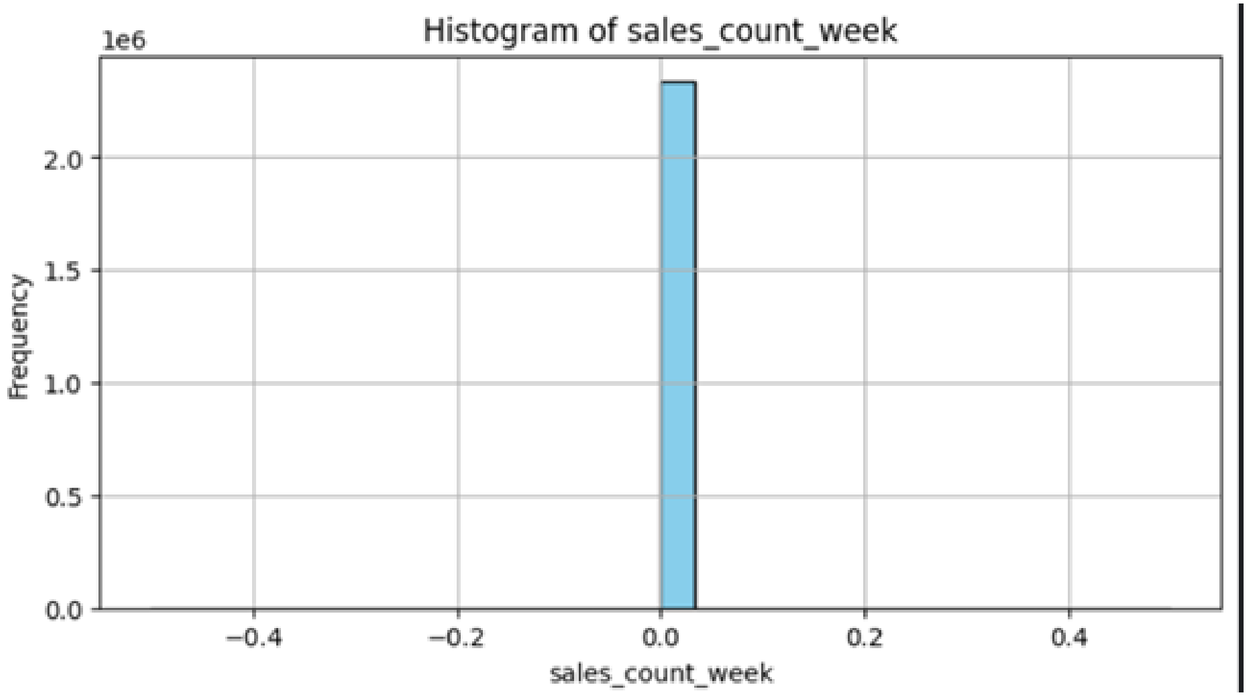


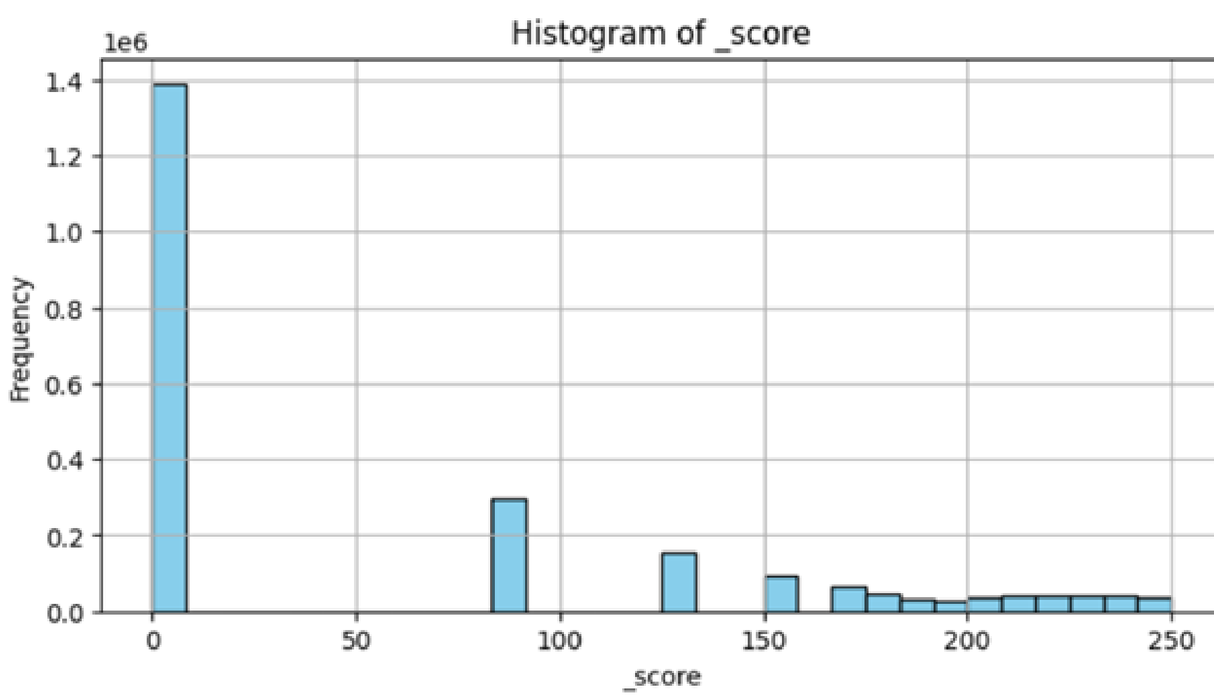
# DATA VISUALIZATION



# DATA VISUALIZATION

- As we can notice the distribution and values of score in the data are better than sales_count_week



# PREDICTIVE ANALYTICS- REGRESSION

| | Train Metrics | | Test Metrics | |
|---|---|---|---|---|
| | **RMSE** | **R2** | **RMSE** | **R2** |
| Linear Regression | 54.39 | 0.666 | 54.53 | 0.664 |
| Random Forest Regressor | 46.606 | 0.754 | 46.548 | 0.756 |
| Decision Tree Regressor | 41.52 | 0.805 | 41.37 | 0.807 |
| Generalized Linear Regression | 54.39 | 0.666 | 54.53 | 0.6649 |
| GBT Regressor | 40.90 | .811 | 40.76 | .8127 |
| Isotonic Regression | 79.708 | 0.283 | 79.76 | 0.283 |

# PREDICTIVE ANALYTICS- CLASSIFICATION WITH SVM

Validation accuracy for each parameter combination:
MaxIter: 10, RegParam: 0.01, Validation Accuracy: 0.8677
MaxIter: 10, RegParam: 0.1, Validation Accuracy: 0.8677
MaxIter: 10, RegParam: 1.0, Validation Accuracy: 0.8677
MaxIter: 50, RegParam: 0.01, Validation Accuracy: 0.8677
MaxIter: 50, RegParam: 0.1, Validation Accuracy: 0.8677
MaxIter: 50, RegParam: 1.0, Validation Accuracy: 0.8677
MaxIter: 100, RegParam: 0.01, Validation Accuracy: 0.8677
MaxIter: 100, RegParam: 0.1, Validation Accuracy: 0.8677
MaxIter: 100, RegParam: 1.0, Validation Accuracy: 0.8677
Best parameter combination:
MaxIter: 10, RegParam: 0.01, Validation Accuracy: 0.8677
**Training Accuracy: 0.8673096628188802**
**Validation Accuracy: 0.8675115536215187**
**Test Accuracy: 0.8684010181241871**

# PREDICTIVE ANALYTICS- CLASSIFICATION MAP REDUCE

- **Map PHASE**
  - Each partition finds its top-most similar neighbors.

- **Reduce PHASE**
  - Combine neighbors from all partitions.
  - Sort combined neighbors based on similarity scores and select the top-k global neighbors.

- **Weighted Voting Prediction**
  - Each neighbor votes for its class label.
  - Votes are weighted by similarity scores and inverse class frequency to reduce bias toward frequent classes.
  - Predict the label with the highest total weighted vote.

Confusion Matrix:

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 28 | 2 | 0 |
| 1 | 81 | 353 | 0 |
| 2 | 0 | 0 | 36 |

| | |
|---|---|
| Accuracy | 0.834 |
| Macro Precision | 0.750416 |
| Micro Precision | 0.834 |
| Macro Recall | 0.915566 |
| Micro Recall | 0.834 |
| Macro F1 Score | 0.824805 |
| Micro F1 Score | 0.834 |

# DESCRIPTIVE ANALYTICS- ASSOCIATION RULES

- **Step 1.** Convert numerical columns to categorical
- **Step 2.** Prepare dataframe: Append column name at the beginning of each item in the column to help differentiate values after creating the itemsets
- **Step 3.** Concatenate columns to create item sets
- **Step 4.** Create the Association Rules from the item sets

Association Rules sorted by Support:

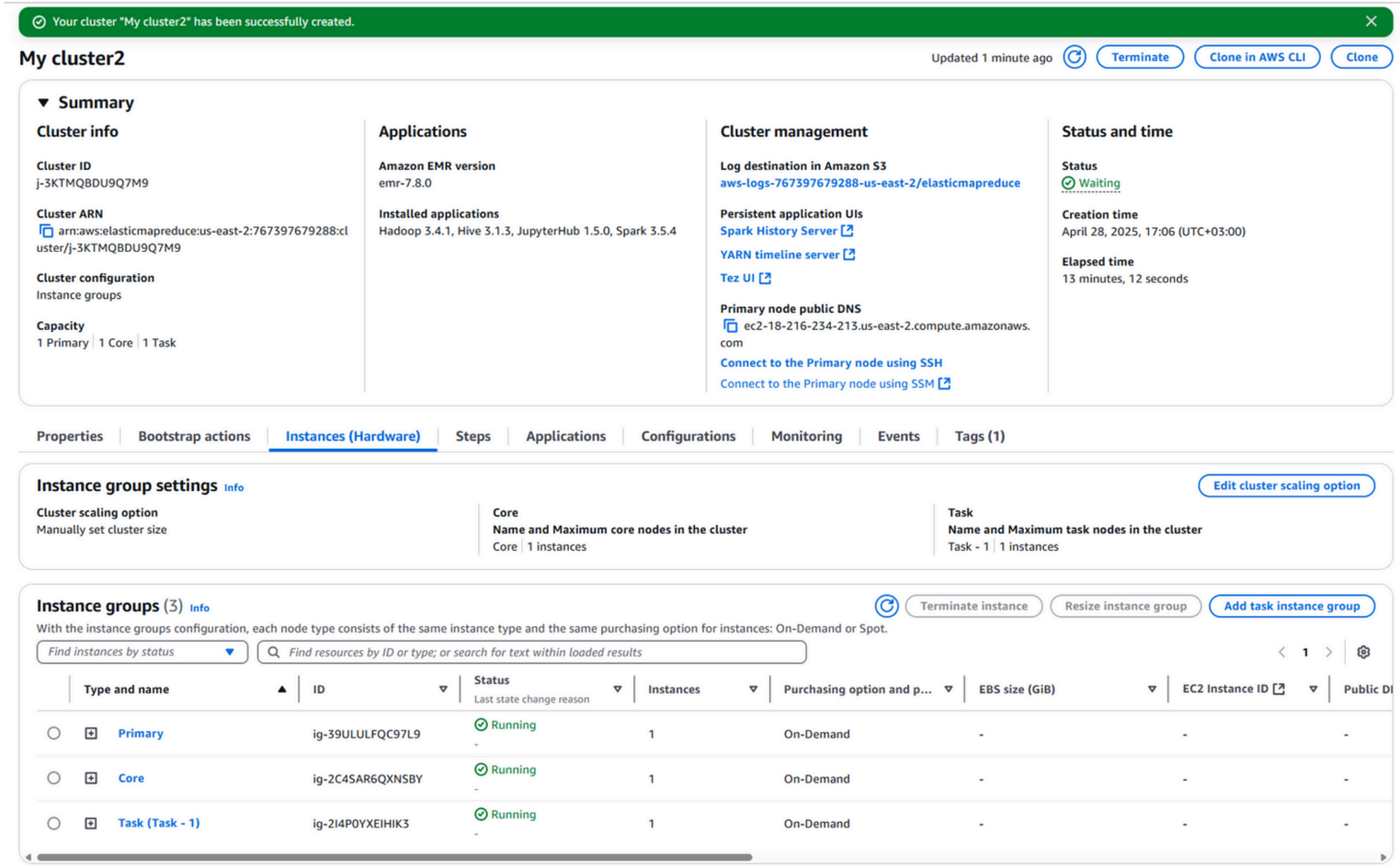
| antecedent | consequent | confidence | lift | support |
|---|---|---|---|---|
| [stock_category_Very Low] | [rating_count_category_Very Low] | 0.9999593277046414 | 0.9999999989750774 | 0.9999341302249872 |
| [rating_count_category_Very Low] | [stock_category_Very Low] | 0.9999748004705681 | 0.9999999989750773 | 0.9999341302249872 |
| [sales_count_week_category_Very Low] | [stock_category_Very Low] | 0.9999748000137937 | 0.9999999985182915 | 0.9999160049848831 |
| [stock_category_Very Low] | [sales_count_week_category_Very Low] | 0.9999412020077967 | 0.9999999985182914 | 0.9999160049848831 |
| [sales_count_week_category_Very Low] | [rating_count_category_Very Low] | 0.999964631598307 | 1.0000053030844678 | 0.9999058371672637 |
| [rating_count_category_Very Low] | [sales_count_week_category_Very Low] | 0.999946506262083 | 1.0000053030844678 | 0.9999058371672637 |
| [sales_count_week_category_Very Low, rating_count_category_Very Low] | [stock_category_Very Low] | 0.9999747991224789 | 0.9999999976269541 | 0.9998806386627286 |
| [sales_count_week_category_Very Low, stock_category_Very Low] | [rating_count_category_Very Low] | 0.9999646307070013 | 1.0000053021931259 | 0.9998806386627286 |
| [rating_count_category_Very Low, stock_category_Very Low] | [sales_count_week_category_Very Low] | 0.999946504914032 | 1.0000053017363375 | 0.9998806386627286 |
| [stock_category_Very Low] | [IsSaleable_1] | 0.9995490680297191 | 0.9999999886320627 | 0.9995238808879953 |
| [IsSaleable_1] | [stock_category_Very Low] | 0.9999747901278141 | 0.9999999886320627 | 0.9995238808879953 |
| [rating_count_category_Very Low] | [IsSaleable_1] | 0.9995490610522705 | 0.9999999816514664 | 0.9995084081220527 |
| [IsSaleable_1] | [rating_count_category_Very Low] | 0.9999593103817349 | 0.9999999816514663 | 0.9995084081220527 |
| [sales_count_week_category_Very Low] | [IsSaleable_1] | 0.9995490528784131 | 0.9999999734739216 | 0.9994902828819485 |
| [IsSaleable_1] | [sales_count_week_category_Very Low] | 0.9999411769648995 | 0.9999999734739217 | 0.9994902828819485 |
| [IsSaleable_1, stock_category_Very Low] | [rating_count_category_Very Low] | 0.999959309355929 | 0.9999999806256188 | 0.9994832096175177 |
| [rating_count_category_Very Low, stock_category_Very Low] | [IsSaleable_1] | 0.9995490496885349 | 0.9999999702826043 | 0.9994832096175177 |
| [IsSaleable_1, rating_count_category_Very Low] | [stock_category_Very Low] | 0.9999747891019922 | 0.999999987606215 | 0.9994832096175177 |
| [IsSaleable_1, sales_count_week_category_Very Low] | [stock_category_Very Low] | 0.9999747886448056 | 0.9999999871490168 | 0.9994650843774134 |
| [IsSaleable_1, stock_category_Very Low] | [sales_count_week_category_Very Low] | 0.9999411754819408 | 0.9999999719908759 | 0.9994650843774134 |

# DESCRIPTIVE ANALYTICS- ASSOCIATION RULES

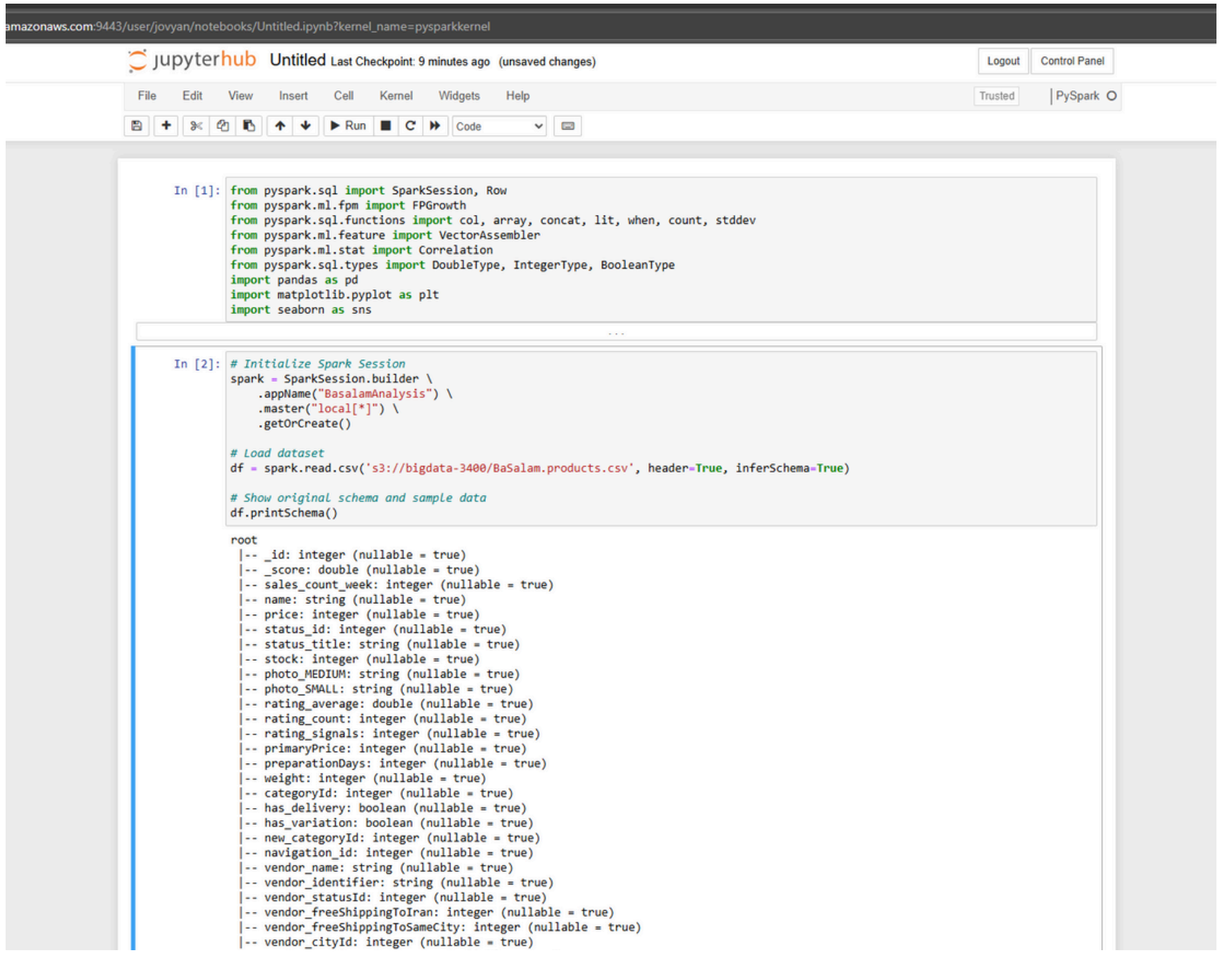
- **Interesting Rules:**
  - Products with no ratings tend to have low engagement metrics
    - Items with rating_average_0.0 almost always have very low sales counts, stock levels, and rating counts
    - This suggests new or unpopular products need specific attention to gain attraction
  - Extremely low rating counts and low sales are consistently linked.
    - Products that have low rating counts tend to have lower sales as the products don’t have feedback so people don’t trust them as much
  - Low-performing products share a profile of
    - Low Stock
    - Low Weekly Sales
  - Well-stocked products share a profile of
    - High-priced
    - Without free shipping
    - Without variation
    - Possibly because they're premium, standardized items.


# AWS EMR Deployment

1. Connected to the cluster and opened jupyter hub
2. Tested the code on a pyspark jupyter notebook



3. Uploaded the dataset into an S3 bucket
4. Created an Amazon EMR Cluster with the following nodes:
    - 1 Primary node
    - 1 Code node
    - 1 Task node




Thank You.