**Project 10: Market Basket Analysis**

**Introduction:**

Market Basket Analysis is a data-driven technique used to uncover patterns and relationships within large transactional datasets, particularly in retail and e-commerce. It helps businesses understand which products or items are often purchased together, providing insights for optimizing product placement, marketing strategies, and promotions.

**Why MBA is Important?**

Market Basket Analysis is a valuable tool for businesses seeking to optimize their product offerings, increase cross-selling opportunities, and improve marketing strategies. It can lead to higher revenue, enhanced customer satisfaction, and overall business success.

Below is the process you can follow for the task of Market Basket Analysis as a DS professional:

1. Gather transactional data, including purchase history, shopping carts, or invoices.
2. Analyze product sales and trends.
3. Use algorithms like Apriori or FP-growth to discover frequent item sets and generate association rules.
4. Interpret the discovered association rules to gain actionable insights.
5. Develop strategies based on the insights gained from the analysis.

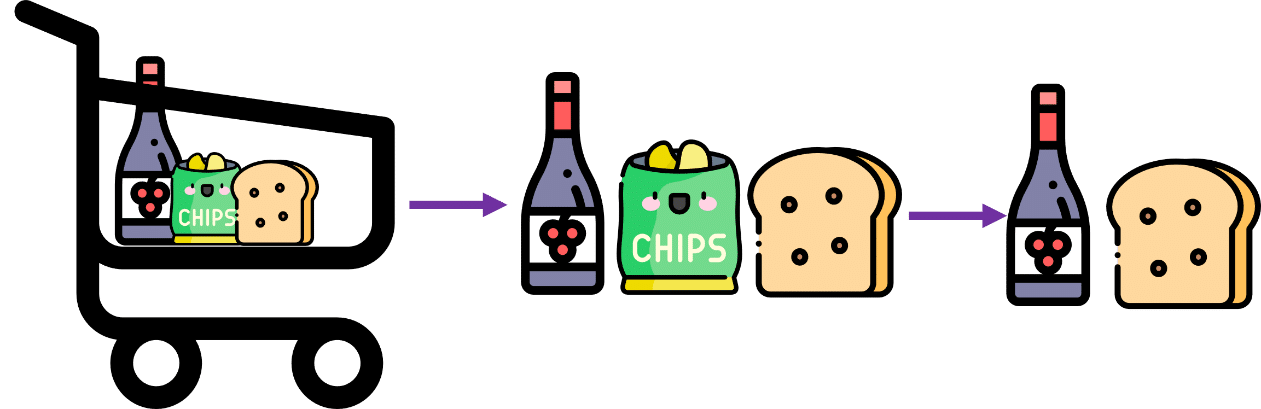
**Types Of Market Basket Analysis**

### **● Predictive Market Basket Analysis**

This kind employs [supervised learning](https://www.simplilearn.com/tutorials/machine-learning-tutorial/supervised-and-unsupervised-learning) methods like regression and classification. In essence, it seeks to imitate the market to examine what factors influence events. In essence, it determines cross-selling by taking into account things bought in a particular order.

### **● Differential Market Basket Analysis**

For competition analysis, this kind of analysis is useful. To identify intriguing patterns in consumer behavior, it compares purchase histories across brands, periods, seasons, days of the week, etc.



# **Apriori Algorithm:**

The Apriori algorithm generates association rules for a given data set. An association rule implies that if an item A occurs, then item B also occurs with a certain probability. There are three components in APRIORI ALGORITHM:

* SUPPORT
* CONFIDENCE
* LIFT

**Example:**

Let’s say there are 10 transactions for books and 8 transactions for pencils and 6 transactions are made for both products.

**1. Support**: It is the total number of transactions made for a particular product divided by the total number of transactions made. Zero represents no support while one represents the highest support. Higher the value of support, the greater the importance of the itemset in the data.

support (A⇒ B) =P (A ∪ B)

Support (Books) = Freq (Books)/Total transactions made

Support (Books) = 6/100 = 0.06%

**2. Confidence:** It is the ratio of combined transactions to individual transactions.

confidence (A⇒ B) =P(B|A)

Confidence (Books) = Combined transactions/Individual transaction

Confidence (Books) = 0.06/0.08 = 0.75

**3. Lift:** It is the ratio of the confidence percent to the support percent.

Lift = 0.75/0.10 = 7.5

* If the value of lift < 1, the combination is not bought by consumers frequently.
* If the value of lift >1, the combination is brought frequently by the consumers.
* If the value of lift = 1, then the purchase of antecedent makes no difference on the consequent.

Market basket analysis is used to search for the rules that result in a lift value greater than 1.

**Market Basket Analysis Overview:**

1. Data Collection: Gather transaction data, including purchased items, transaction timestamps, and relevant details.
2. Data Preprocessing: Clean and format the data, removing irrelevant information and addressing missing values.
3. Association Rule Mining: Use algorithms like Apriori or FP-Growth to find frequent item sets (items often bought together).
4. Support and Confidence Calculation: Compute support (frequency of item sets) and confidence (likelihood of one item following another).
5. Generate Association Rules: Create rules in the form "IF [antecedent], THEN [consequent]" based on support and confidence.
6. Interpret Results: Analyze findings to understand item associations and customer behavior patterns.
7. Business Applications: Apply insights to enhance product recommendations, optimize store layouts, and tailor marketing campaigns.

**Key Definitions:**

* **Antecedent**: The left side of an association rule, representing items often purchased together (e.g., "bread" in "IF bread, THEN butter").
* **Consequent**: The right side of an association rule, showing items associated with the antecedent (e.g., "butter" in "IF bread, THEN butter").

This process helps businesses leverage transaction data to make data-driven decisions and improve customer experiences.

**Market Basket Analysis using Python:**

Importing libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from apyori import apriori

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

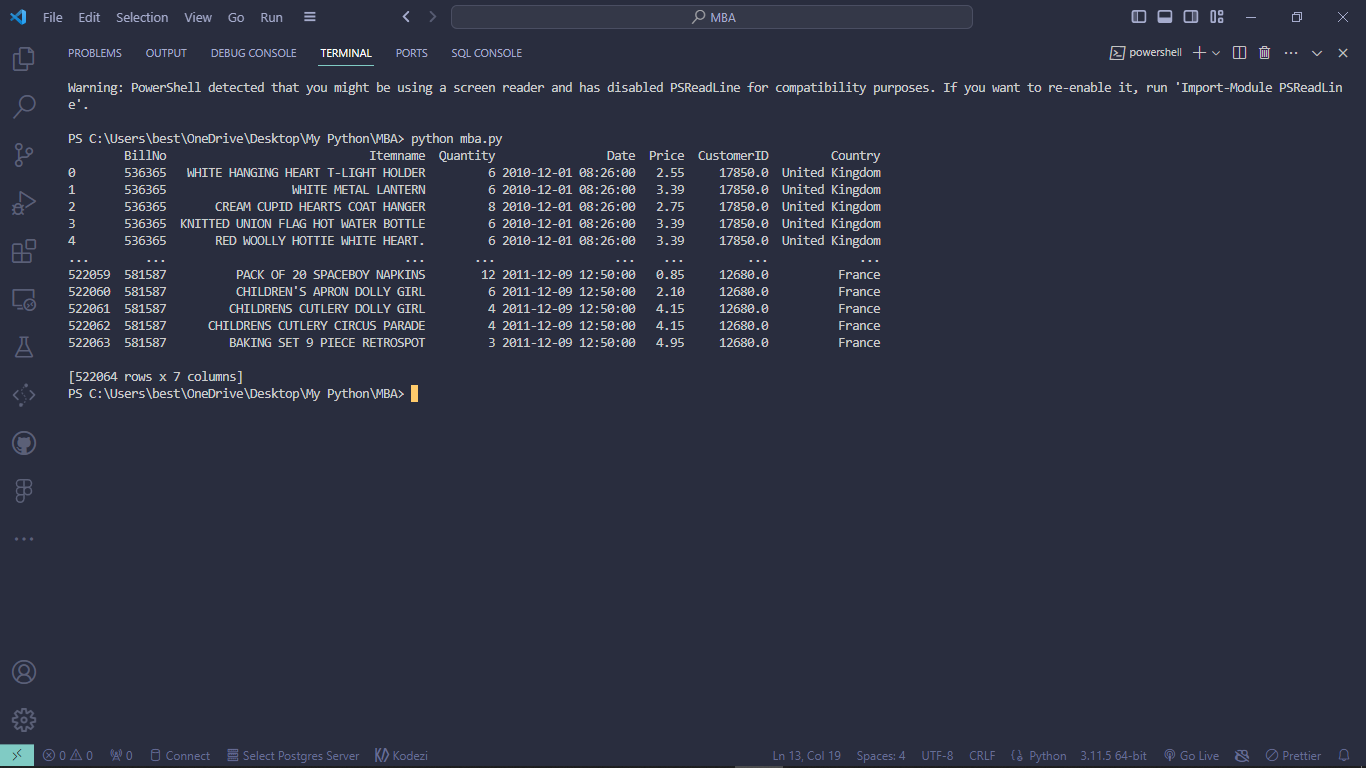
Data Preprocessing

# load & read the datasets

data = pd.read\_excel("mba.xlsx")

print(data)

OUTPUT:

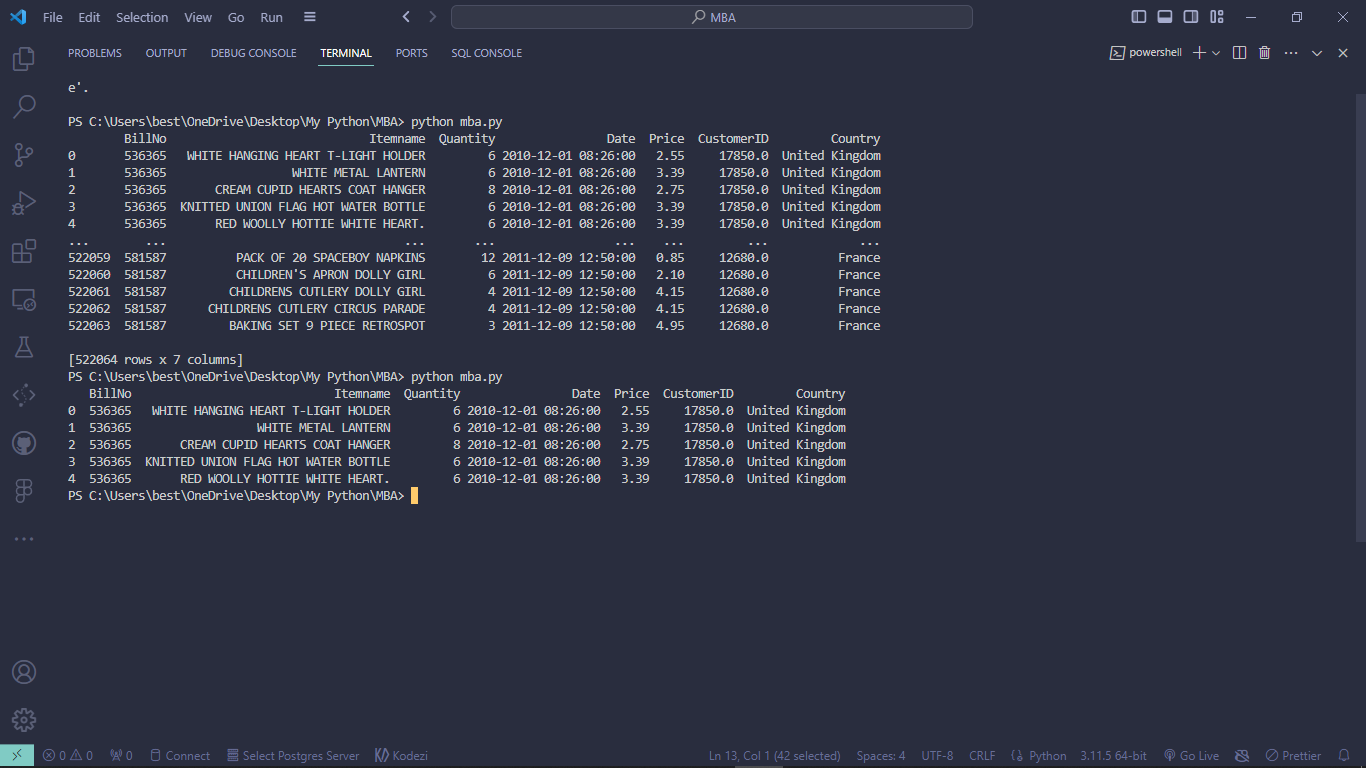


To see the Header of dataset:

*# head of datasets*

print(mba\_data.head())

OUTPUT:



To remove missing & duplicate values:

missing\_values = mba\_data['BillNo'].isnull().sum()

duplicate\_values = mba\_data['BillNo'].duplicated().sum()

print(f"Missing values in 'BillNo' column: {missing\_values}")

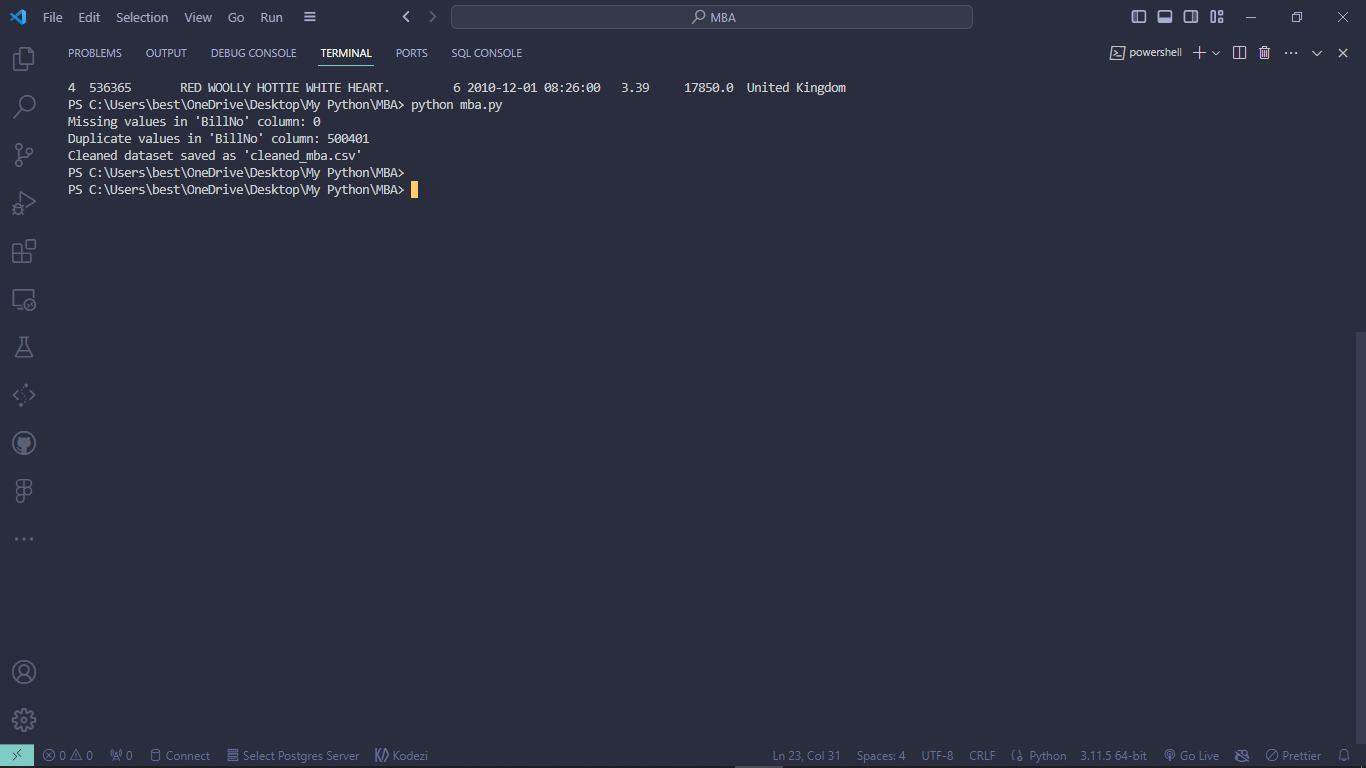
print(f"Duplicate values in 'BillNo' column: {duplicate\_values}")

mba\_data\_cleaned=mba\_data.dropna(subset=['BillNo']).drop\_duplicates(subset=['BillNo']);

mba\_data\_cleaned.to\_csv("cleaned\_mba.csv", index=False)

print("Cleaned dataset saved as 'cleaned\_mba.csv'")

OUTPUT:



**Combine All Sets of Data to Determine the Date with the Highest Sales**:

date\_sales = mba\_data.groupby('Date')['Price'].sum()

date\_with\_highest\_sales = date\_sales.idxmax()

print(f"The date with the highest sales is {date\_with\_highest\_sales} with a total sales of {date\_sales.max()}")

OUTPUT:

The date with the highest sales is **2010-12-07 15:08:00** with a total sale of **13541.33**

**Determine Which Country has the Highest Sales:**

country\_sales = mba\_data.groupby('Country')['Price'].sum()

country\_with\_highest\_sales = country\_sales.idxmax()

print(f"The country with the highest sales is {country\_with\_highest\_sales} with a total sales of {country\_sales.max()}")

OUTPUT:

The country with the highest sales is **United Kingdom** with a total sale of **1845443.914**

K-Means clustering to find patterns in your dataset

# Import necessary libraries

import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

# Load your dataset

data = pd.read\_excel('mba.xlsx')

selected\_features = data[['Quantity', 'Price']]

num\_clusters = 3

kmeans = KMeans(n\_clusters=num\_clusters)

data['Cluster'] = kmeans.fit\_predict(selected\_features)

# Visualize the clusters (you may need to adjust this based on the number of features)

for cluster in range(num\_clusters):

    cluster\_data = data[data['Cluster'] == cluster]

    plt.scatter(cluster\_data['Quantity'], cluster\_data['Price'], label=f'Cluster {cluster + 1}')

plt.xlabel('Quantity')

plt.ylabel('Price')

plt.legend()

plt.show()

# Analyze the clusters and patterns

for cluster in range(num\_clusters):

    cluster\_data = data[data['Cluster'] == cluster]

    print(f'Cluster {cluster + 1}:')

    print(cluster\_data.describe())

    # Further analysis within each cluster

    centroid = kmeans.cluster\_centers\_[cluster]

    print(f'Centroid for Cluster {cluster + 1}:')

    print(f'Quantity: {centroid[0]}')

    print(f'Price: {centroid[1]}')

    # Plot histograms for each feature within the cluster

    plt.figure(figsize=(10, 6))

    plt.hist(cluster\_data['Quantity'], bins=20, alpha=0.5, label='Quantity')

    plt.hist(cluster\_data['Price'], bins=20, alpha=0.5, label='Price')

    plt.xlabel('Feature Values')

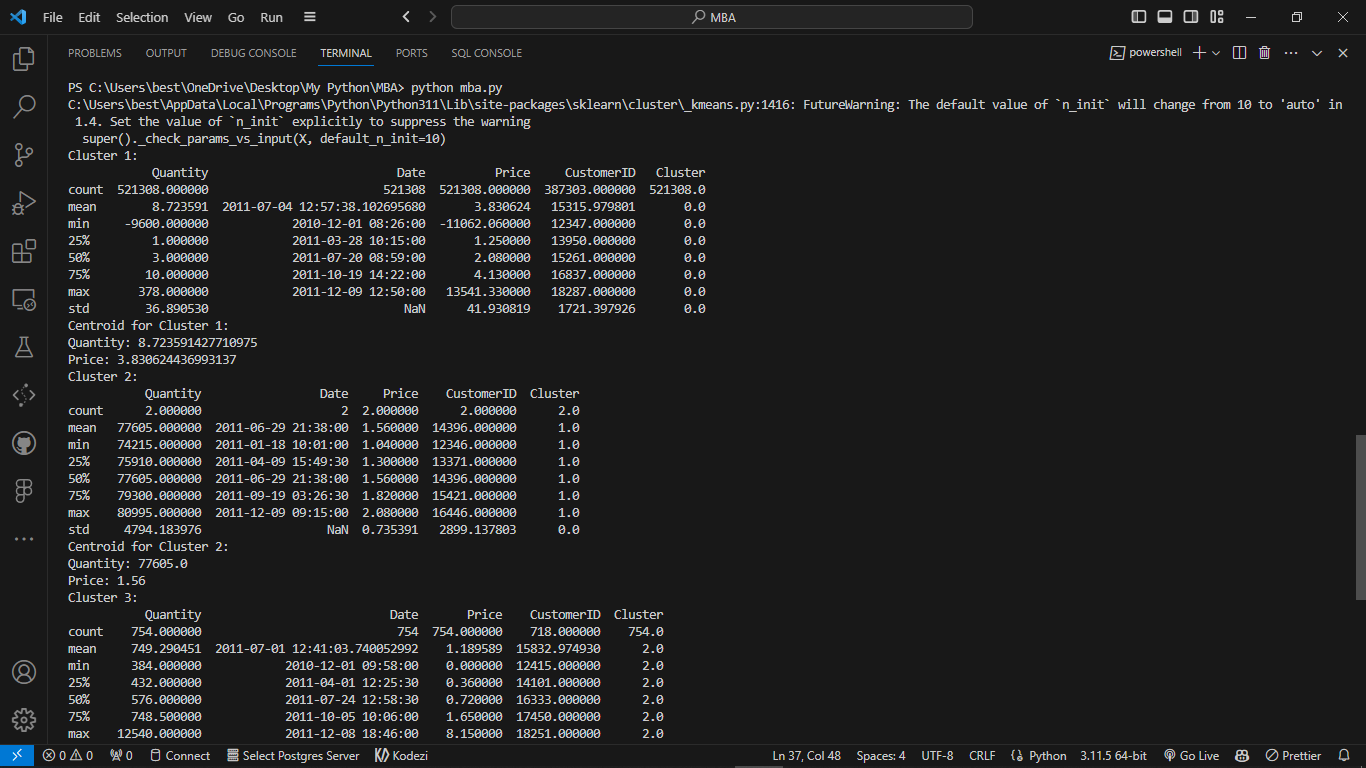
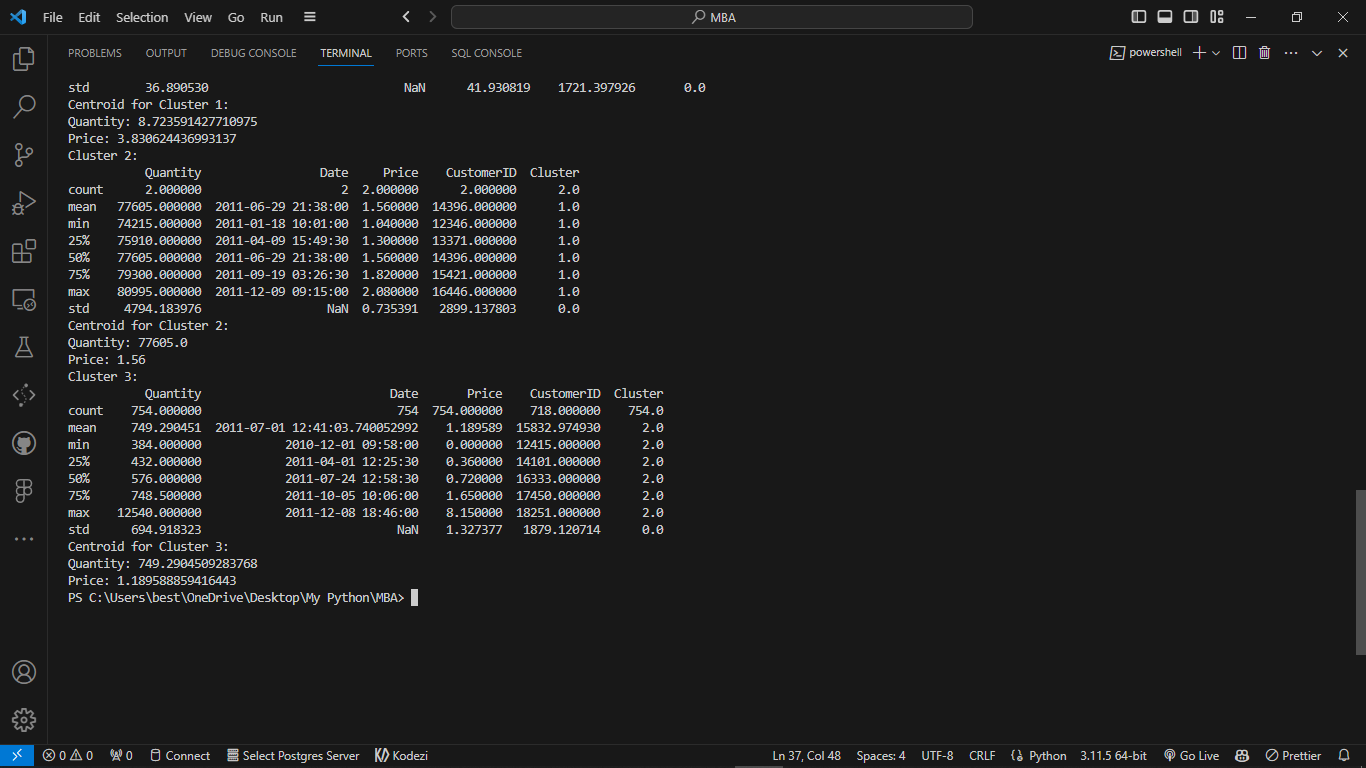
    plt.ylabel('Frequency')

    plt.title(f'Cluster {cluster + 1} Feature Distributions')

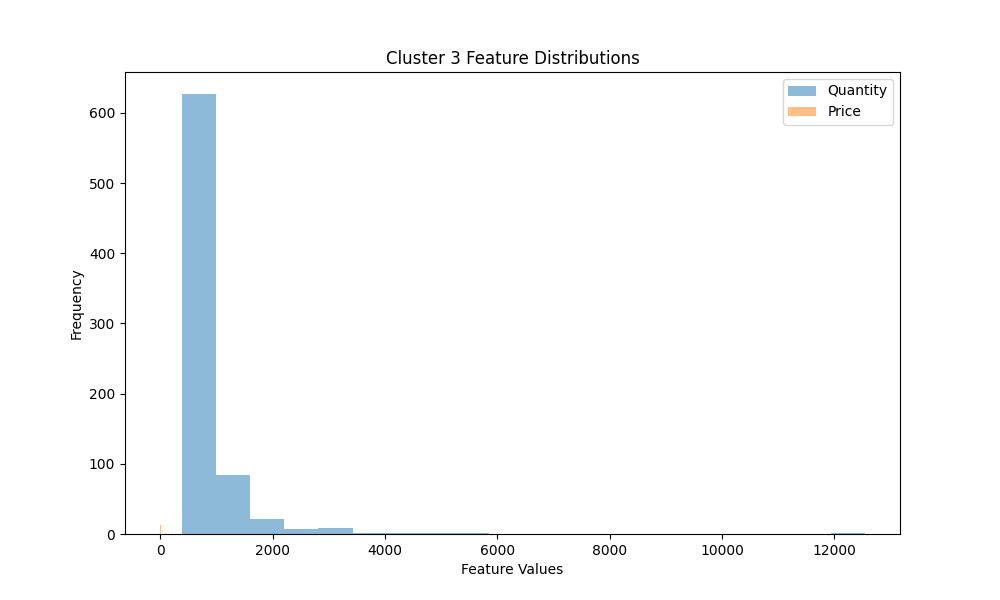
    plt.legend()

    plt.show()

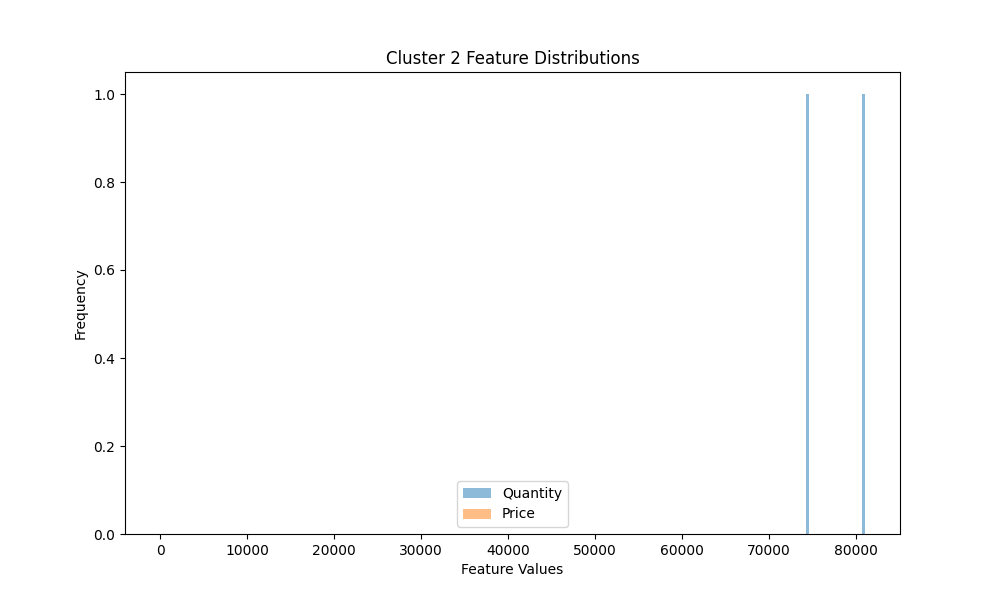
OUTPUT:

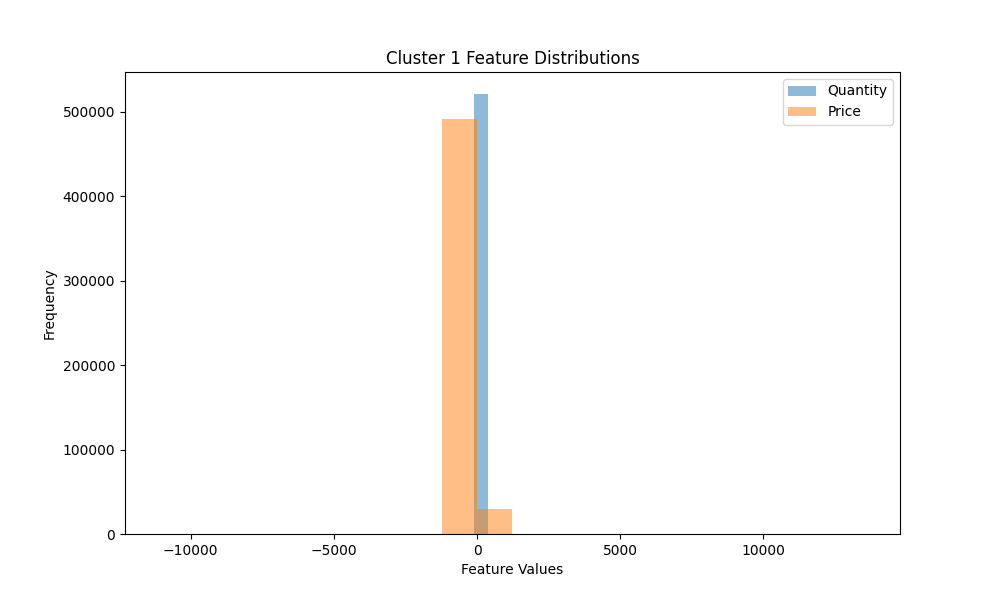
Cluster 3



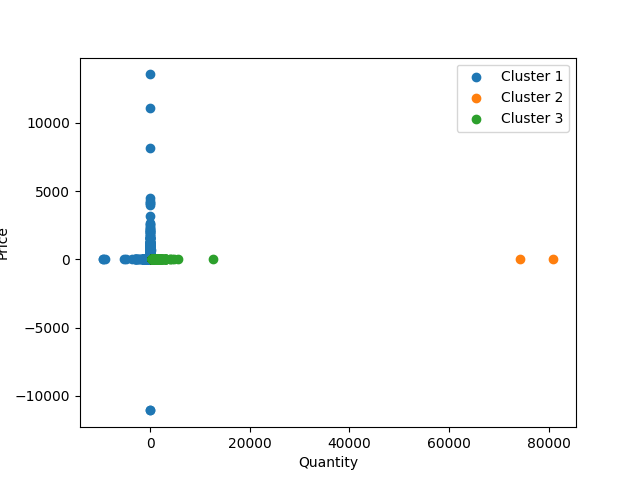
Cluster 2



Cluster 1



All 3 Cluster:



Frequent itemset

import pandas as pd

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

# Load your dataset

data = pd.read\_excel('mba.xlsx')

basket = (data.groupby(['BillNo', 'Itemname'])['Quantity']

          .sum().unstack().reset\_index().fillna(0)

          .set\_index('BillNo'));

basket\_sets = basket.applymap(lambda quantity: bool(quantity >= 1))

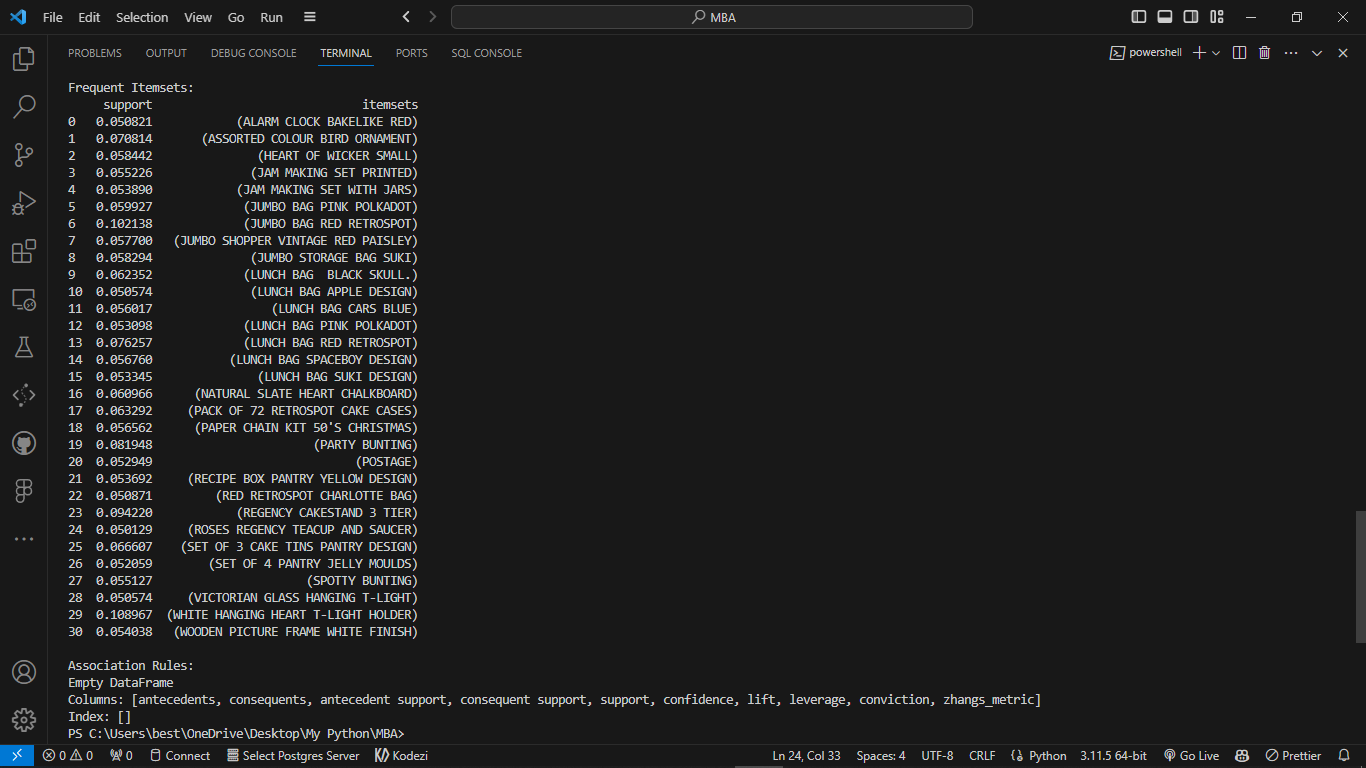
frequent\_itemsets = apriori(basket\_sets, min\_support=0.01, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0)

print("Association Rules:")

print(rules)

OUTPUT:



Frequent Items in Graph (Top 10)

import pandas as pd

import matplotlib.pyplot as plt

# Load your dataset

data = pd.read\_excel('mba.xlsx')

item\_sales = data.groupby('Itemname')['Quantity'].sum().sort\_values(ascending=False)

top\_n = 10;

# Create a bar chart to visualize the top N most frequently bought items

plt.figure(figsize=(10, 6))

plt.bar(item\_sales.index[:top\_n], item\_sales.values[:top\_n])

plt.xlabel('Item Names')

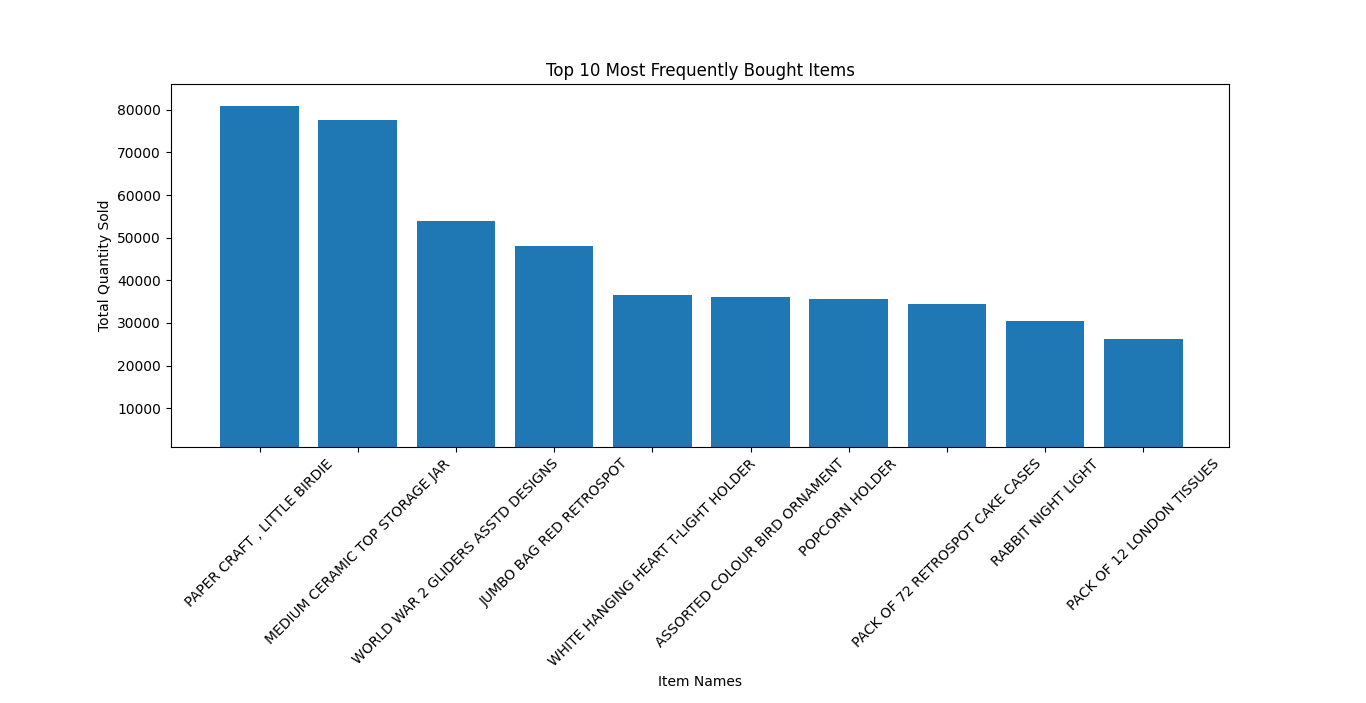
plt.ylabel('Total Quantity Sold')

plt.title(f'Top {top\_n} Most Frequently Bought Items')

plt.xticks(rotation=45)

plt.show()

OUTPUT:



5. Rules along with their evaluation metrics & sorting

import pandas as pd

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

# Load your dataset

data = pd.read\_excel('mba.xlsx')

basket = (data.groupby(['BillNo', 'Itemname'])['Quantity']

          .sum().unstack().reset\_index().fillna(0)

          .set\_index('BillNo'));

basket\_sets = basket.applymap(lambda quantity: bool(quantity >= 1));

frequent\_itemsets = apriori(basket\_sets, min\_support=0.01, use\_colnames=True);

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0);

# Display the association rules with evaluation metrics

print("Association Rules:")

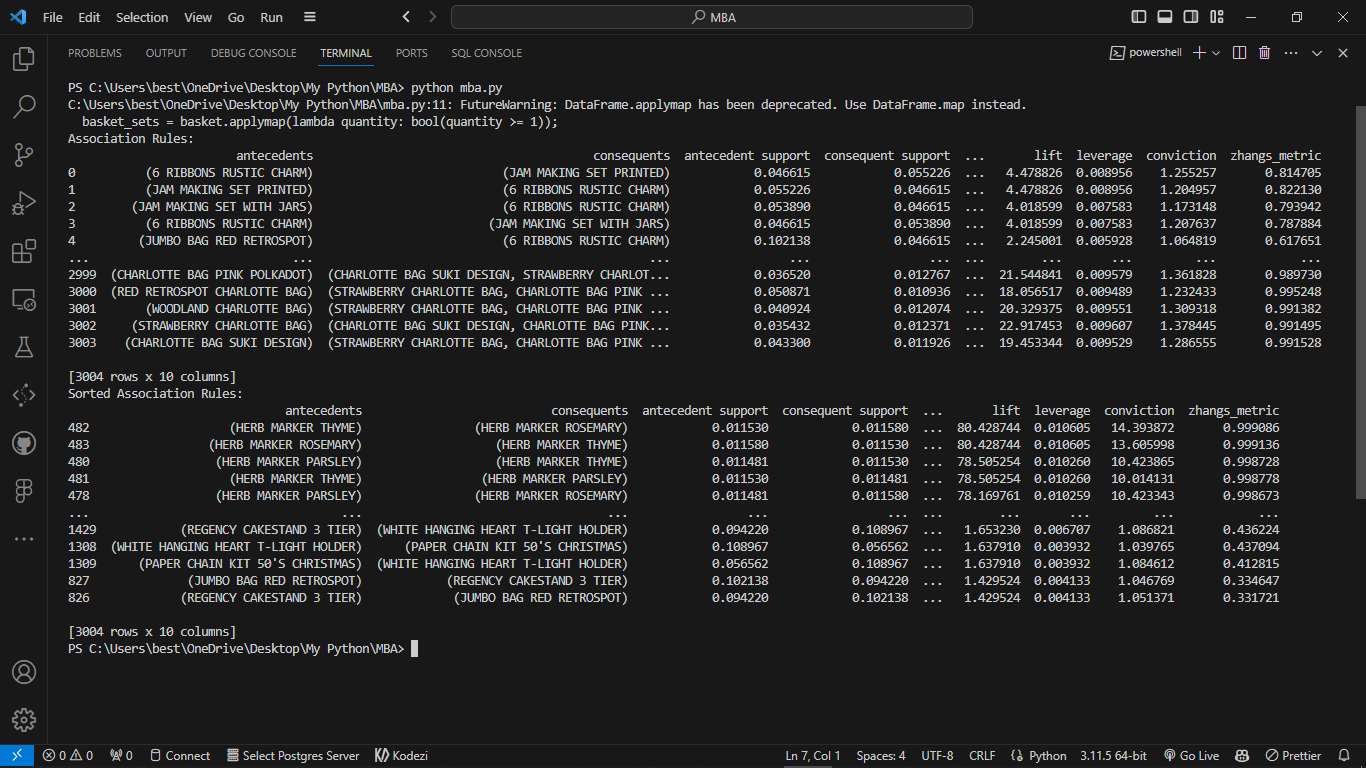
print(rules)

sorted\_rules = rules.sort\_values(by=['lift'], ascending=False)

print("Sorted Association Rules:")

print(sorted\_rules)

OUTPUT:



Top 10 Most Popular Items using Seaborn

item\_popularity = data['Itemname'].value\_counts().head(10)

# Top 10 Most Popular Items using Seaborn

plt.figure(figsize=(12, 6))

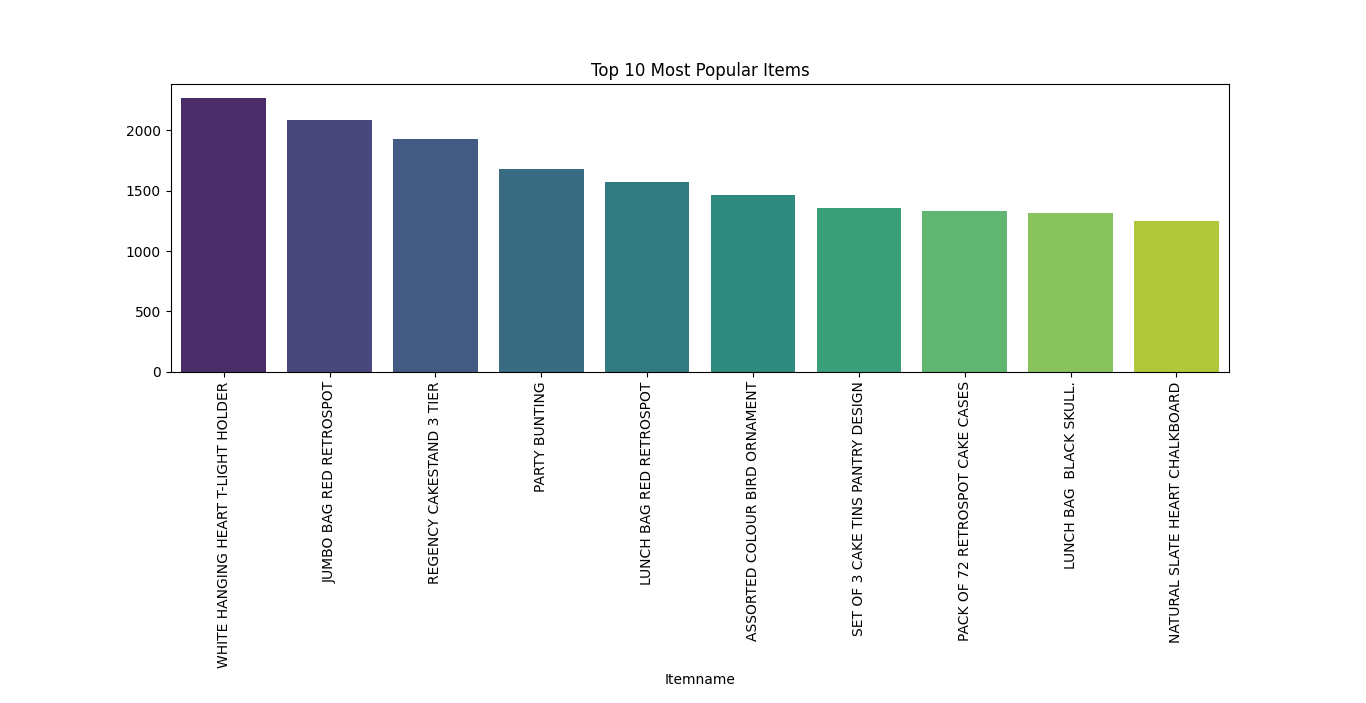
sns.barplot(x=item\_popularity.index, y=item\_popularity.values, palette='viridis')

plt.title(f'Top 10 Most Popular Items')

plt.xticks(rotation=90)

plt.show()

OUTPUT:

****

Calculate average quantity and spending per customer

customer\_behavior = data.groupby('CustomerID').agg({'Quantity': 'mean', 'Price': 'sum'}).reset\_index()

plt.figure(figsize=(12, 6))

plt.scatter(customer\_behavior['Quantity'], customer\_behavior['Price'], s=100, c='coral', label='Customers')

plt.title('Customer Behavior')

plt.xlabel('Average Quantity')

plt.ylabel('Total Spending')

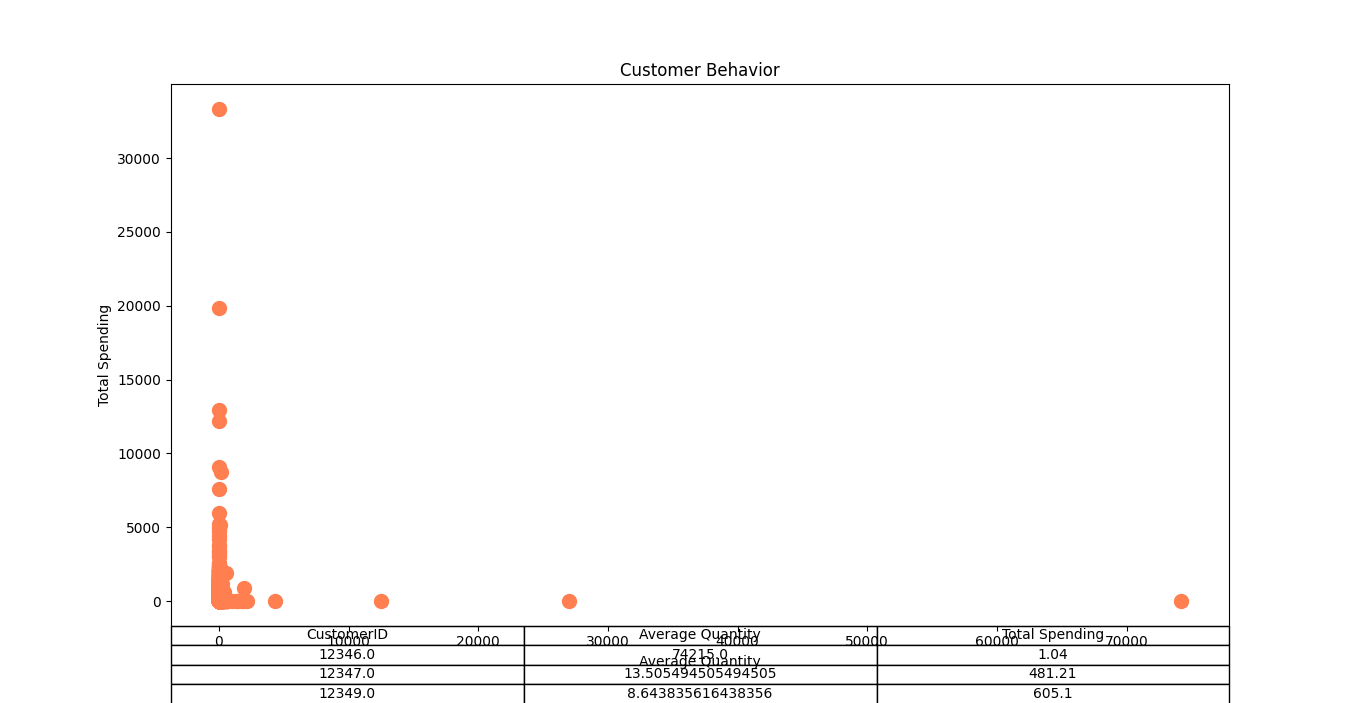
cell\_text = []

**for** row **in** customer\_behavior.itertuples(index=False):cell\_text.append([row.CustomerID, row.Quantity, row.Price])

plt.table(cellText=cell\_text, colLabels=['CustomerID', 'Average Quantity', 'Total Spending'], loc='bottom', cellLoc='center')

plt.show()

OUTPUT:

****

## **Benefits of market basket analysis**

The never-ending list of impeccable benefits that market basket analysis has to offer is widely being leveraged by organizations around the world. This is also the reason one can notice a spike in the hiring of [ML engineers](https://www.turing.com/hire/ml-engineers) in companies around the world.

* **Personalized recommendations**
* **Promotions and campaigns**
* **Customer behavior analysis**
* **In-store operations optimizations**
* **New marketing tactics**

**Advantages:**

* Easy to Implement
* Scalability
* Recommendation Systems
* Cross-Selling
* Improved Product Placement
* Data-Driven Decision-Making
* Cost Reduction
* Increased Revenue

**Disadvantages:**

* Computational Complexity
* Memory Usage
* Doesn't Consider Time
* Privacy Concerns

**Conclusion**

Cross-selling and upselling is the secret mantra of the retail industry that pushed the consumer to buy more. Organizations are using this technique wisely and making billions by playing with the mind of the customer.