Project 10: Market Basket Insights

Indroduction:

**Market Basket Analysis** is a data mining technique used in retail and e-commerce to discover associations between products that are frequently purchased together by customers. It is based on the concept of finding patterns or relationships in transaction data, such as point-of-sale data, online shopping carts, or invoices. Market Basket Analysis helps businesses understand customer buying behavior, improve sales, and make data-driven decisions.

**Abstraction:**

* K-Means clustering to find patterns in your dataset
* Frequent 30 Items
* Frequent Items in Graph (Top 10)
* Visualize the association rules using a scatter plot
* Rules along with their evaluation metrics & sorting

**Analysis Program & Result:**

1. 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')

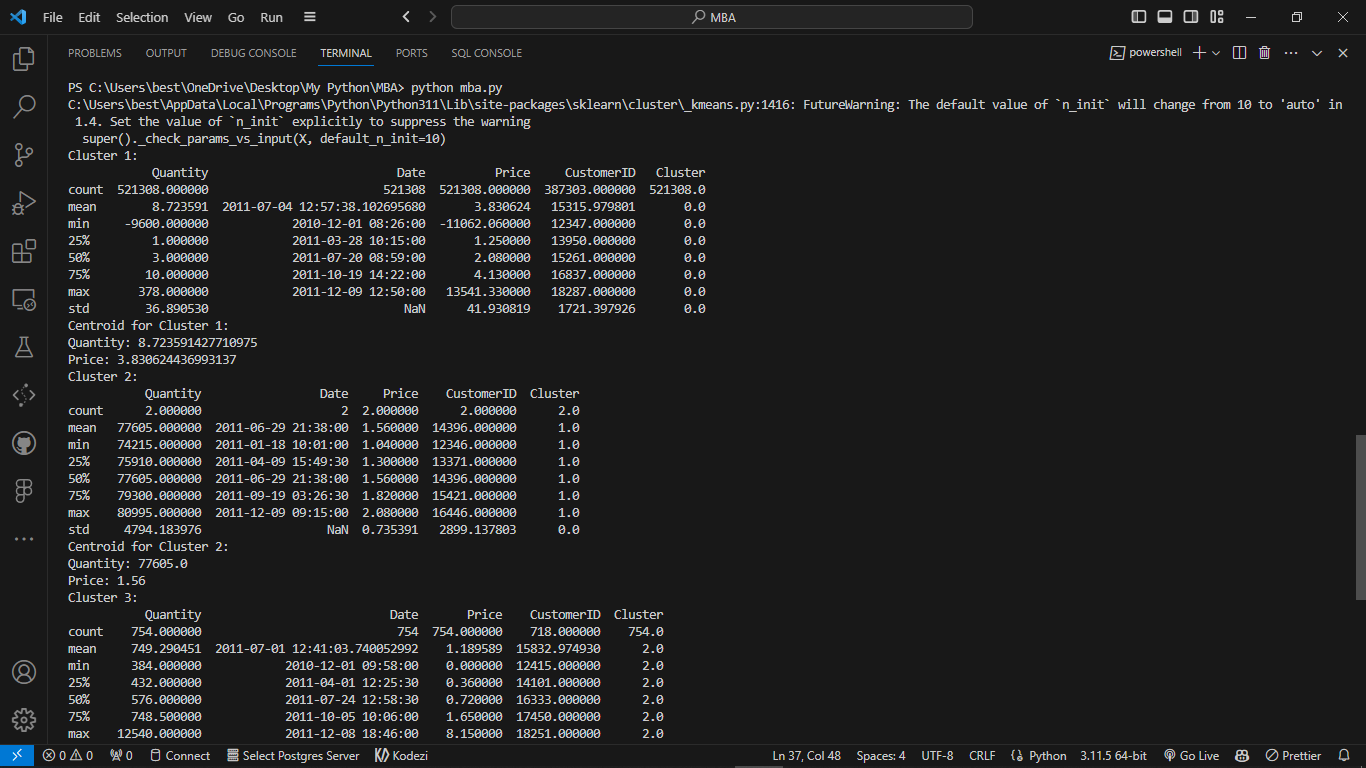
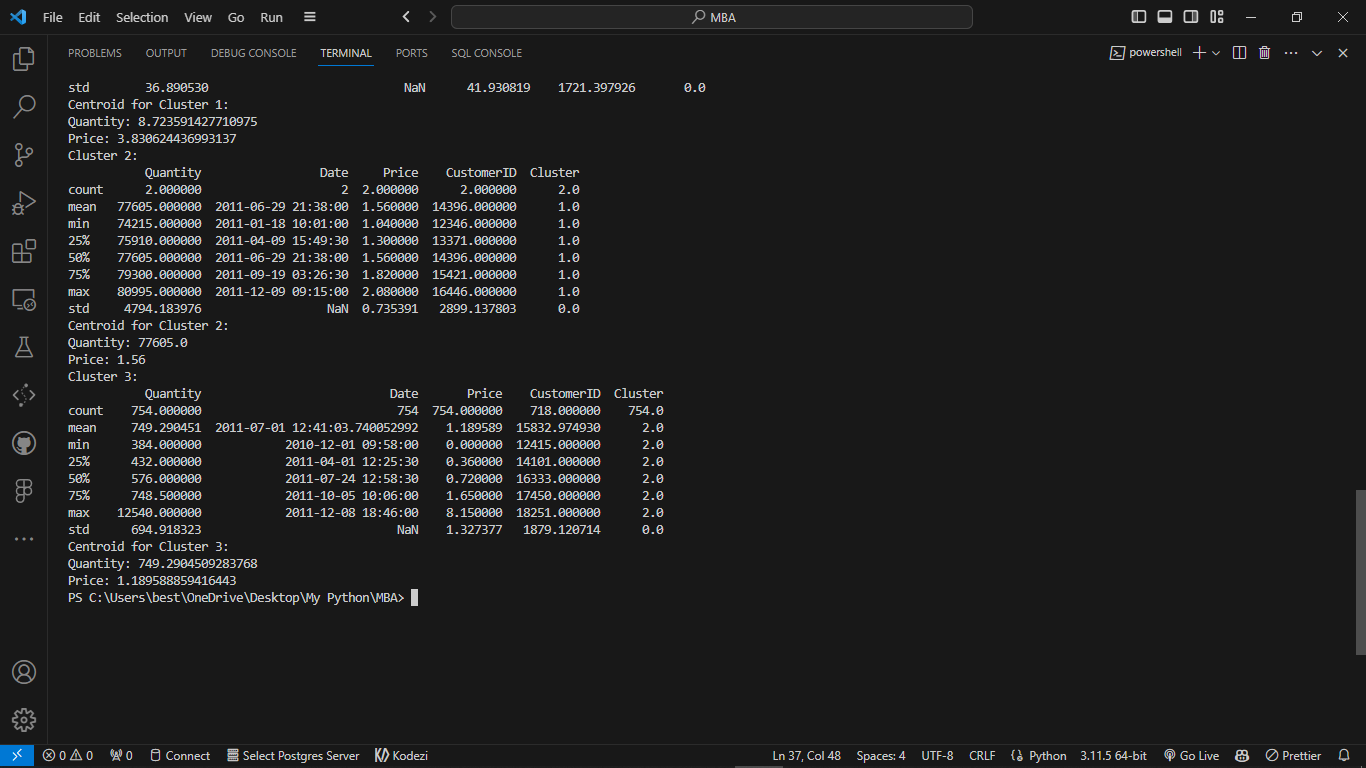
    plt.ylabel('Frequency')

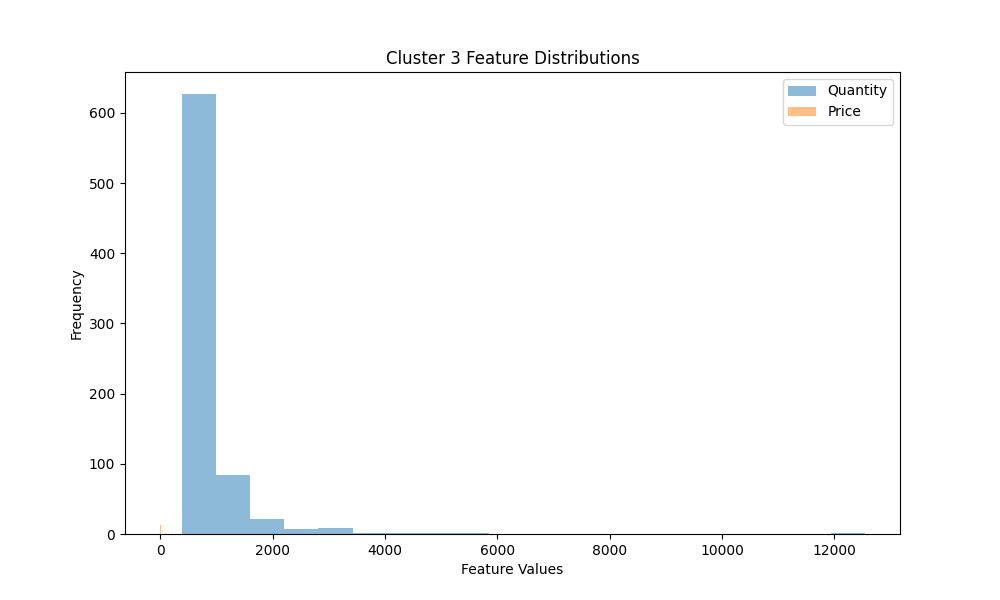
    plt.title(f'Cluster {cluster + 1} Feature Distributions')

    plt.legend()

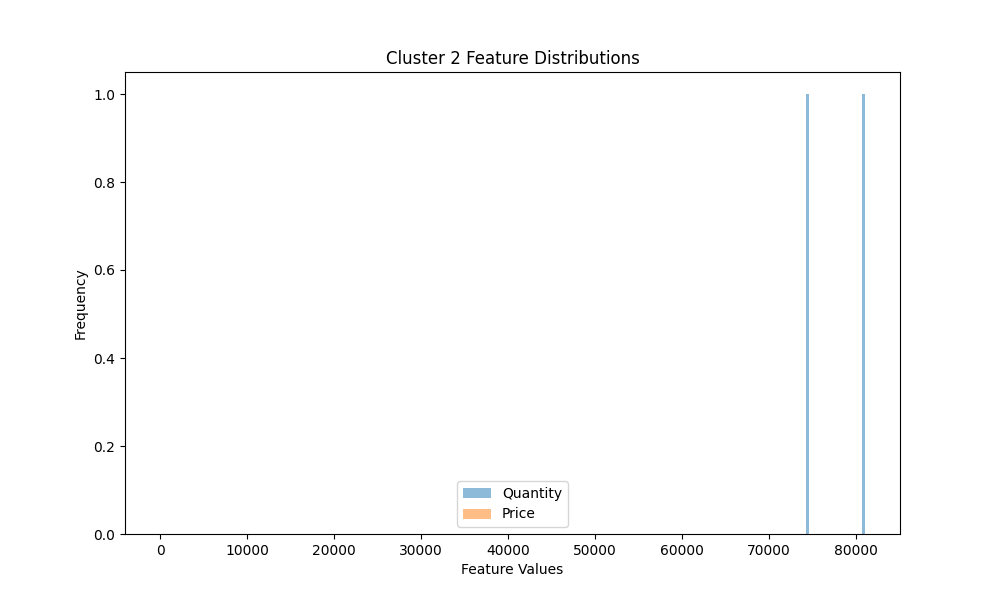
    plt.show()

**Output :**

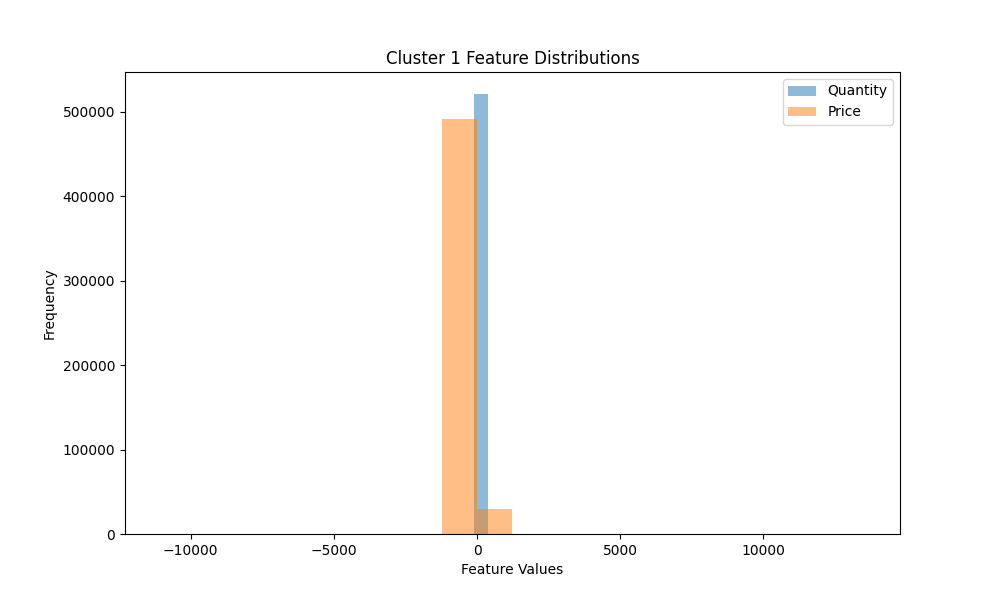
 

Cluster 3

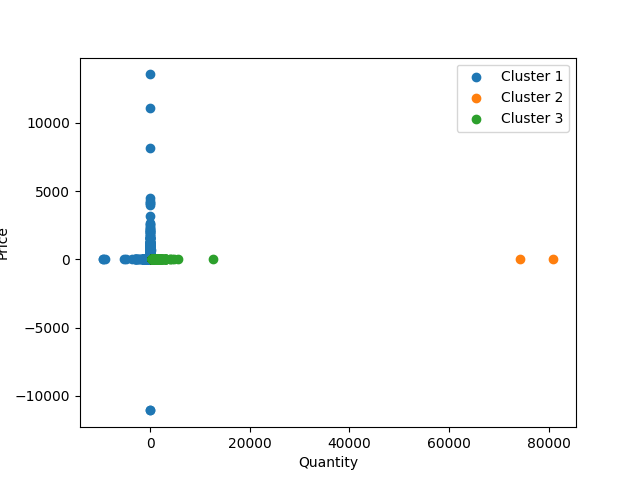
Cluster 2



Cluster 1



All 3 Cluster:



2. 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))

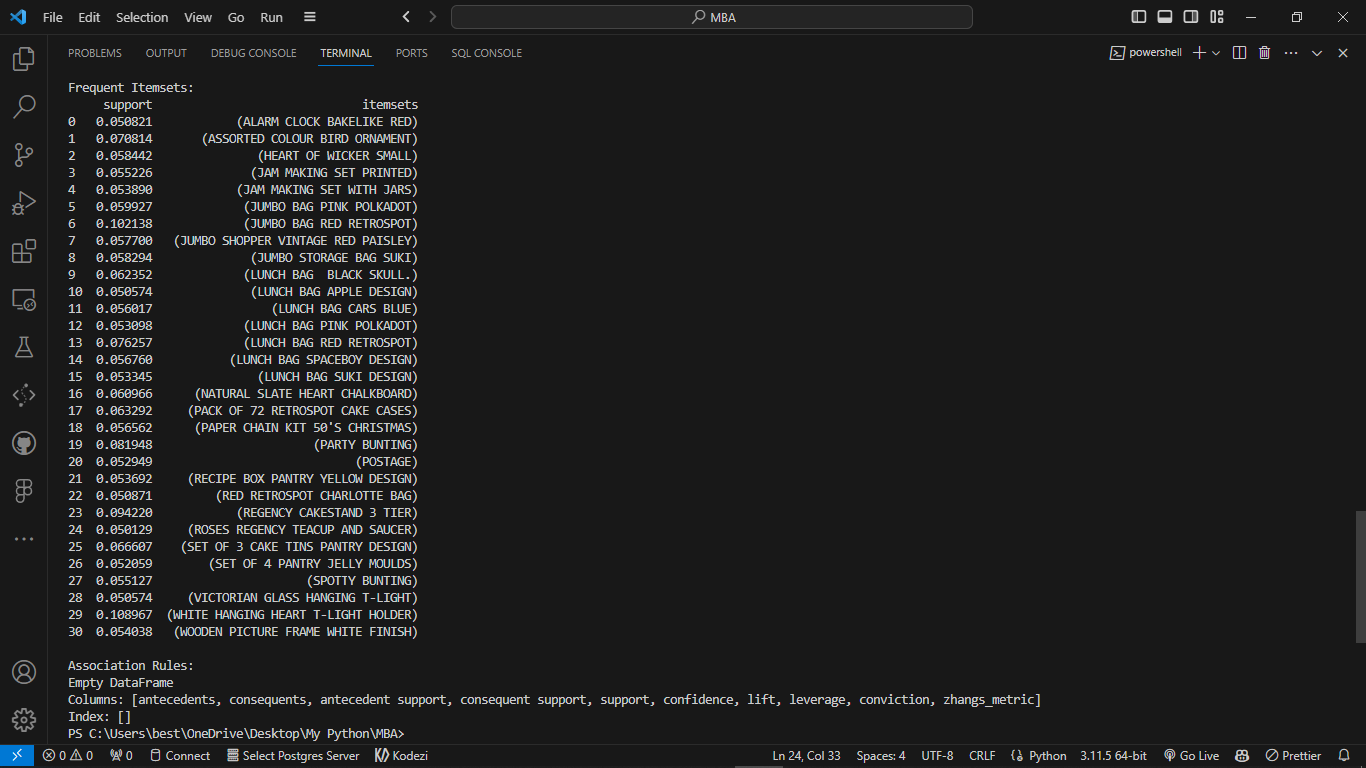
frequent\_itemsets = apriori(basket\_sets, min\_support=0.01, use\_colnames=True)

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

print("Association Rules:")

print(rules)

**Output :**



3. 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')

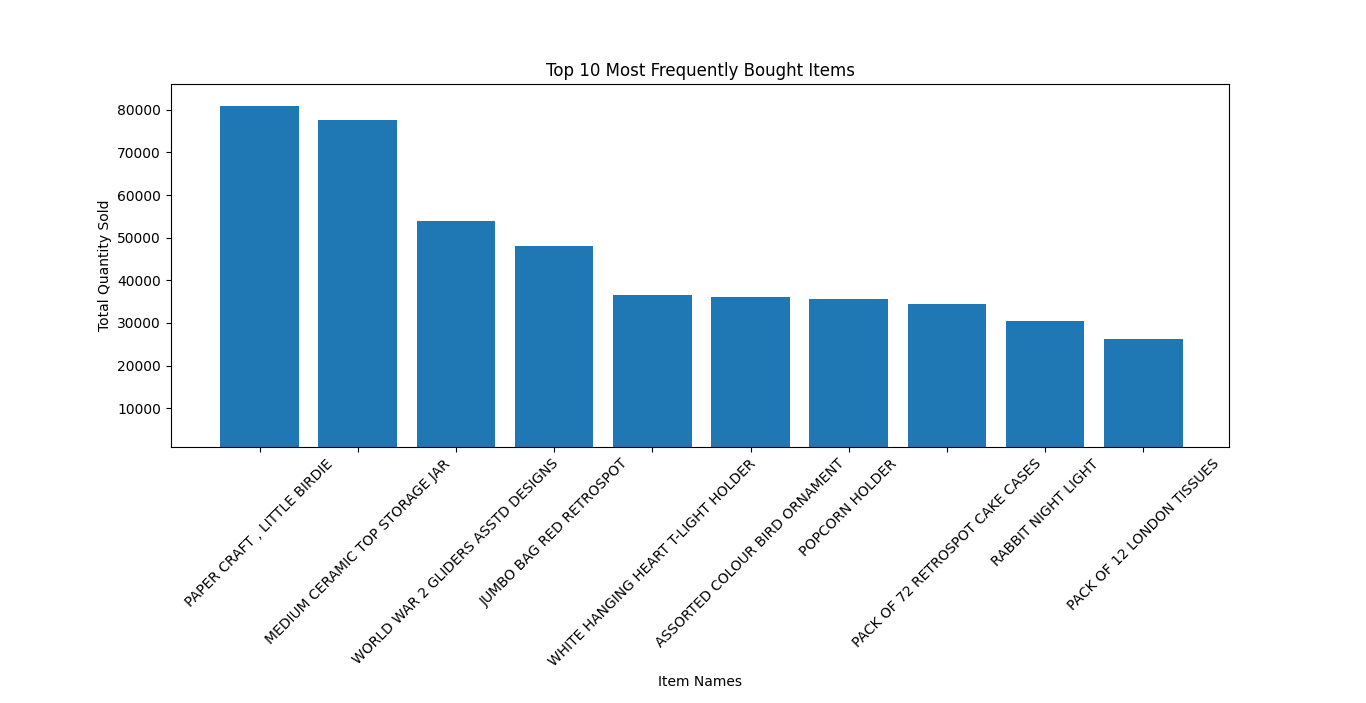
plt.ylabel('Total Quantity Sold')

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

plt.xticks(rotation=45)

plt.show()

**Output :**



4. Visualize the association rules using a scatter plot

import pandas as pd

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

import matplotlib.pyplot as plt

# 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 in a table

print("Association Rules:")

print(rules)

# Visualize the association rules using a scatter plot

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

plt.scatter(rules['support'], rules['confidence'], alpha=0.5)

plt.xlabel('Support')

plt.ylabel('Confidence')

plt.title('Association Rules')

plt.show()

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:")

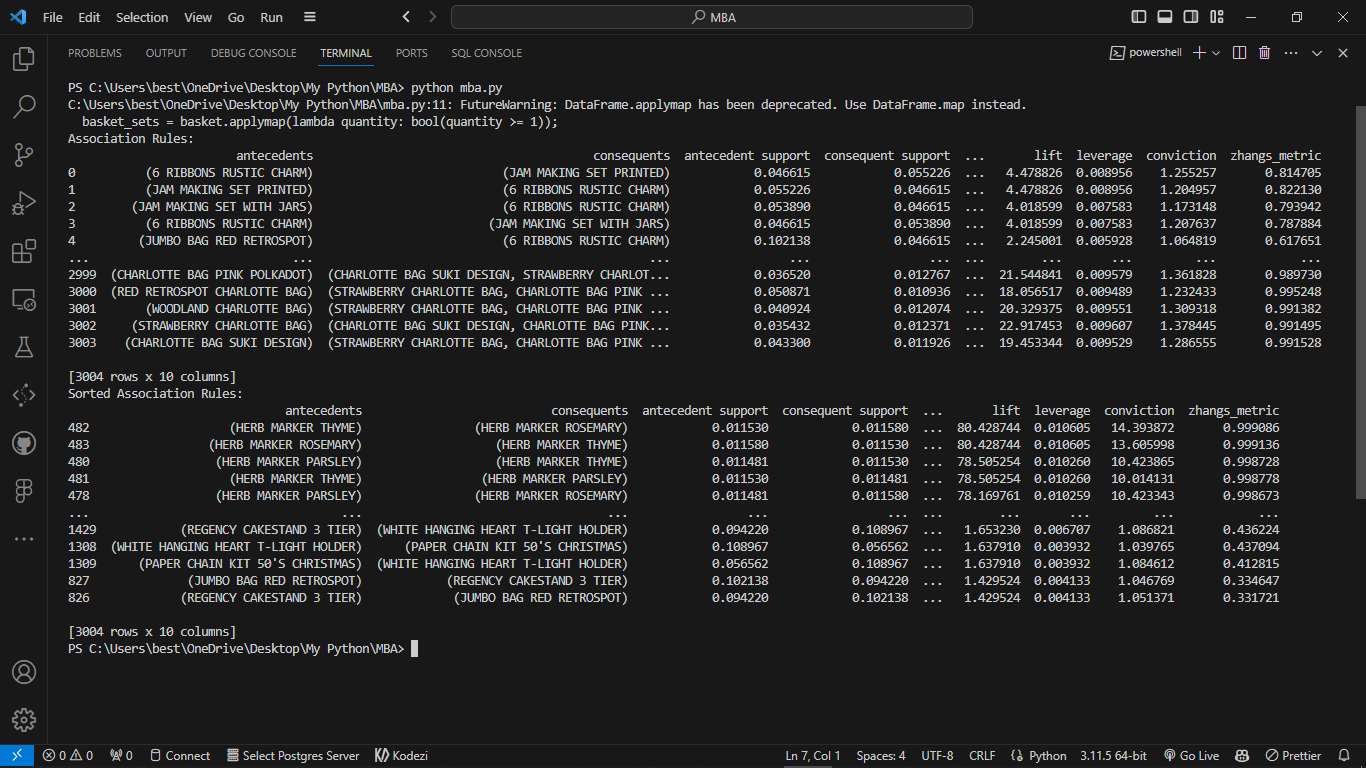
print(rules)

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

print("Sorted Association Rules:")

print(sorted\_rules)

**Output :**



Conclusion:  
In conclusion, association rule mining, particularly through the Apriori algorithm, is a powerful technique for discovering patterns, relationships, and associations within transactional data. Here are the key takeaways:

1. **Association Rules**
2. **Apriori Algorithm**
3. **Data Preprocessing**
4. **Evaluation Metrics**
5. **Applications**
6. **Rule Evaluation**
7. **Visualization**
8. **Memory Optimization**

Overall, association rule mining is a valuable technique for uncovering hidden patterns in transactional data and making data-driven decisions in various industries. Its practical applications and adaptability to different domains make it a fundamental tool for data analysts and researchers.