Project 10: Market Basket Insights

**Introduction:**

This project focuses on implementing Market Basket Analysis using the Apriori algorithm in Python, a widely-used method for discovering associations between items in a transaction dataset. By analyzing historical sales data, this project aims to uncover hidden relationships and patterns among products, ultimately helping businesses make data-driven decisions.

1. **Data Preprocessing:** The project begins with data preprocessing, where we clean and transform raw transaction data into a suitable format for analysis.
2. **Market Basket Analysis:** Using the Apriori algorithm, we identify frequent itemsets and association rules. These rules reveal which products tend to be purchased together, allowing businesses to make targeted recommendations and promotions.
3. **Visualization:** Data visualization techniques are employed to provide clear and insightful representations of the associations and patterns discovered.

Note: before I am going to check dataset I see some error. I change some chances in excel file like separate column to work on it.

**Loading & Preprocessing the dataset:**

**1. Importing:**

*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

output:

no error

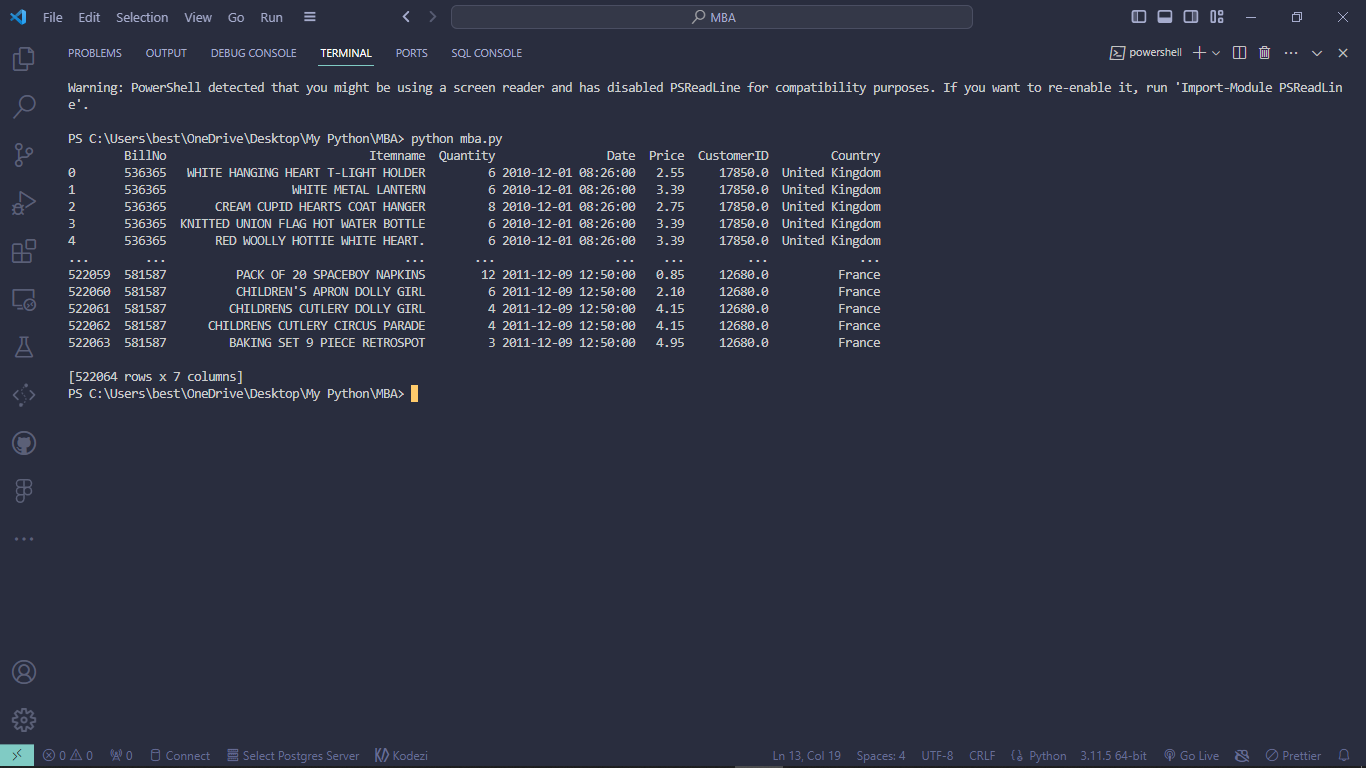
**2.Load & read the datasets:**

*# load & read the datasets*

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

print(mba\_data)

output:

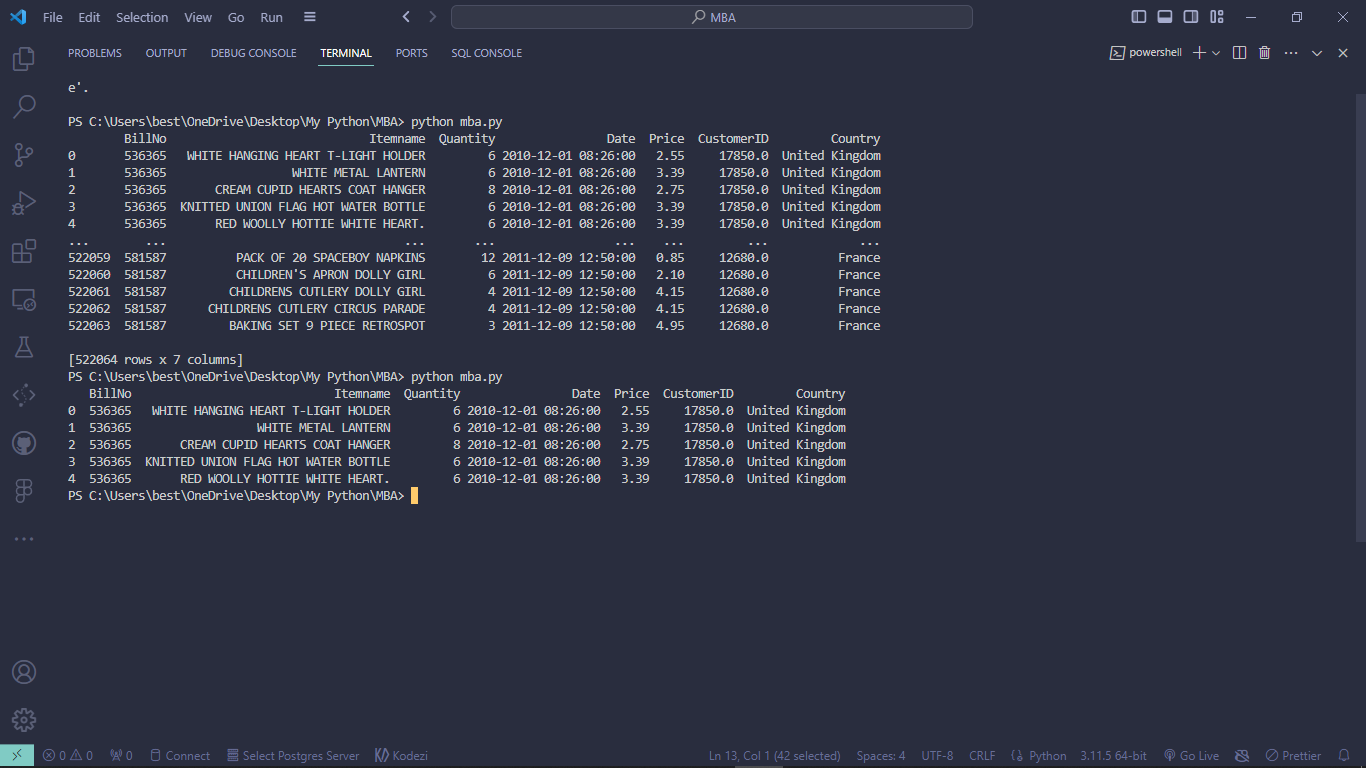


**3. To see the Header of dataset:**

*# head of datasets*

print(mba\_data.head())

output:



**4. To remove missing & duplicate values:**

*# Check for missing values in the "BillNo" column*

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

*# Check for duplicate values in the "BillNo" column*

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}")

*# Clean the dataset by removing rows with missing or duplicate values in the "BillNo" column*

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

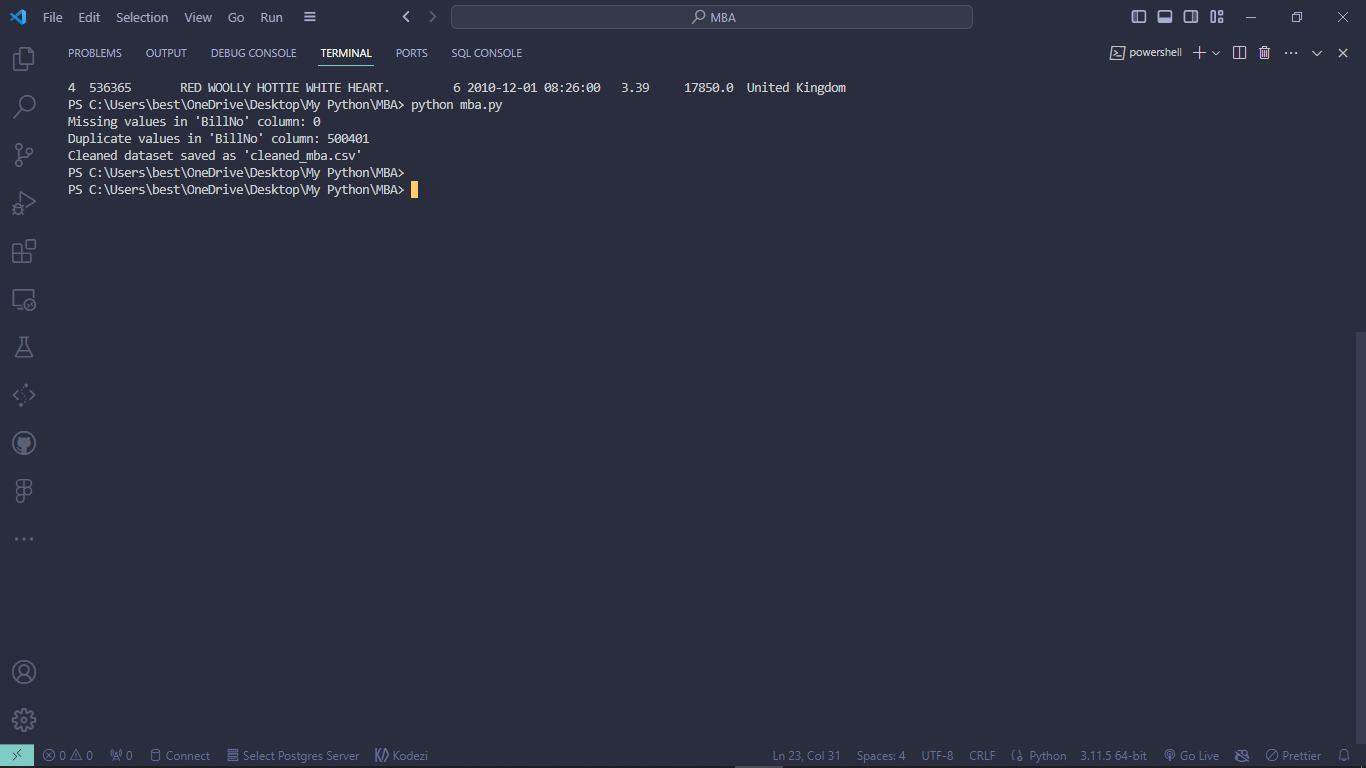
*# Save the cleaned dataset to a new CSV file*

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

*# Confirm the cleaning and saving*

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

output:



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

*# Group the data by Date and calculate the total sales for each date*

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

*# Find the date with the highest total sales*

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 sales of **13541.33**

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

*# Group the data by Country and calculate the total sales for each country*

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

*# Find the country with the highest total sales*

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 sales of **1845443.914**

**7. Display Which Item has the Highest Sale using the Quantity Column:**

*# Group the data by Itemname and calculate the total quantity sold for each item*

item\_sales = mba\_data.groupby('Itemname')['Quantity'].sum()

*# Sort the items by total quantity in descending order*

sorted\_items = item\_sales.sort\_values(ascending=False)

*# Plot a bar chart to visualize the top-selling items*

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

plt.bar(sorted\_items.index, sorted\_items)

plt.xlabel('Item Names')

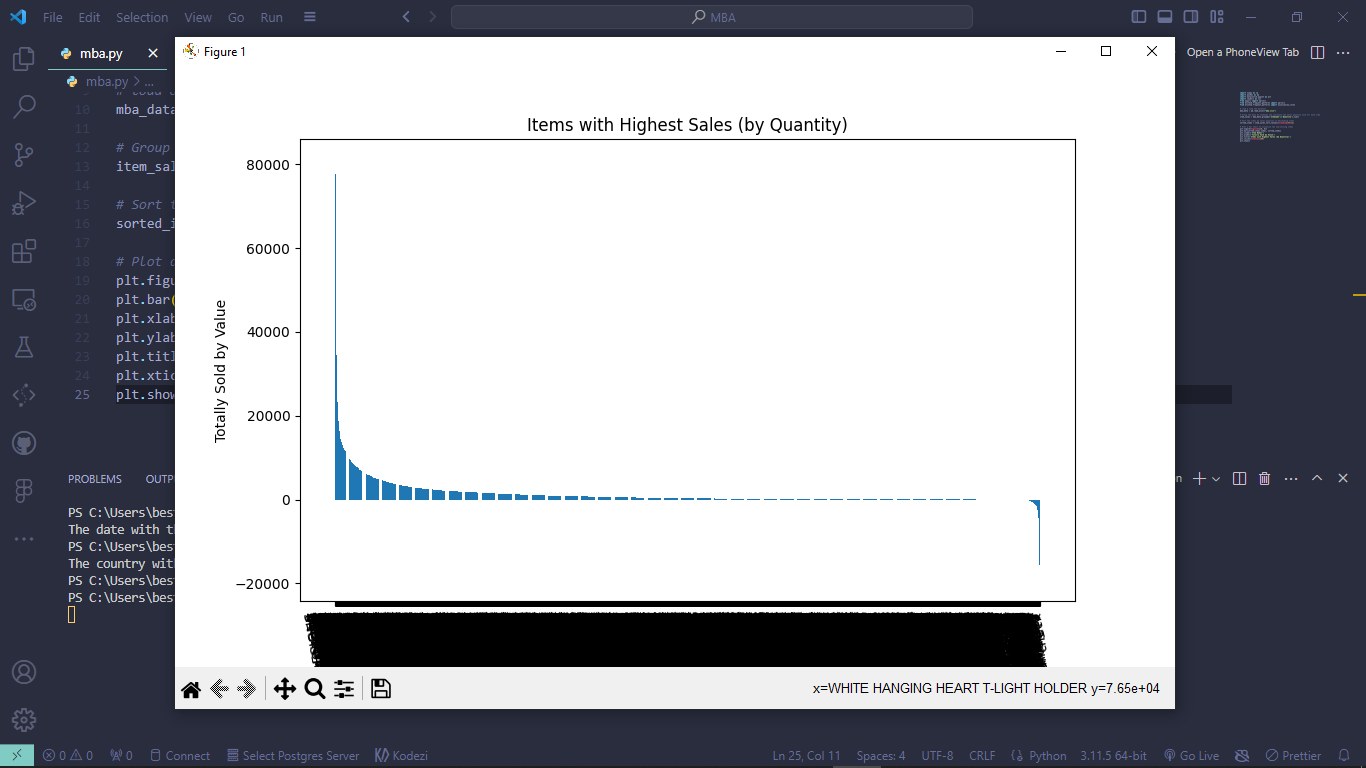
plt.ylabel('Totally Sold by Value')

plt.title('Items with Highest Sales (by Quantity)')

plt.xticks(rotation=100)

plt.show()

output:



**Conclusion:**

Market Basket Analysis is a valuable tool for businesses as it can provide insights into customer behavior and help optimize marketing strategies, inventory management, and more. Your project demonstrates how Python and the Apriori algorithm can be used to uncover meaningful associations in transaction data. The code provided helps you understand which products are selling well, identify patterns in your data, and visualize item associations.

Overall, Market Basket Analysis is a crucial aspect of data-driven decision-making, enabling businesses to make informed choices to improve sales, customer experience, and overall profitability.