Phase-3

Data Visualization

Market Basket Analysis:

Date	17 Oct 2023
Team ID	Proj-212168-Team-1
Project Name	Market Basket Insights
Maximum Mark	

Data visualization:

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Program:

#Import package:

Explaination:

- Numpy :(import numpy as np) a library for mathematical operations and handling arrays.
- pandas :(import pandas as pd) a library for data manipulation and analysis.

- Matplotlib.pyplot: (import as plt) a library for creating visualization.
- Seaborn :as a library for creating additional data visualization.
- mlxtend.frquent_patterns: a module for performing frequent itemset mining and association rule leaening.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

#Load the Dataset:

```
dataset=pd.read_csv('insights.csv')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async`
  and should_run_async(code)
```

- This code reads contents of a csv file called "insights.csv" and saves it a variable called"dataset".The "pd" modul is already imported.

#Replace Missing Value:

```
df=dataset.fillna({'Itemname':'abc'})
df
df1=dataset.fillna(value=dataset['CustomerID'].mean())
df1
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell`
  and should_run_async(code)
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6.000000	12/1/2010 8:26	2.550000	17850.000000	United Kingdom
1	536365	WHITE METAL LANTERN	6.000000	12/1/2010 8:26	3.390000	17850.000000	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8.000000	12/1/2010 8:26	2.750000	17850.000000	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6.000000	12/1/2010 8:26	3.390000	17850.000000	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6.000000	12/1/2010 8:26	3.390000	17850.000000	United Kingdom
...	...	...	...	...	...	...	...
26994	538566	RED RETROSPOT APRON	1.000000	12/13/2010 11:21	12.720000	15602.731236	United Kingdom
26995	538566	PARTY INVITES WOODLAND	2.000000	12/13/2010 11:21	1.660000	15602.731236	United Kingdom
26996	538566	VINTAGE RED TEATIME MUG	1.000000	12/13/2010 11:21	2.510000	15602.731236	United Kingdom
26997	538566	CHRISTMAS TOILET ROLL	7.000000	12/13/2010 11:21	2.510000	15602.731236	United Kingdom
26998	538566	CREAM SLICE FLANN	15602.731236	15602.731236	15602.731236	15602.731236	15602.731236

26999 rows × 7 columns

- This code is filling the missing values in the columns "itemname" of the dataframe "dataset" with the value "abcd".The filled dataframe is then displayed.

#Find the Missing Value:

```
[20] df1.isnull().sum()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecatio
  and should_run_async(code)
BillNo        0
Itemname      0
Quantity      0
Date          0
Price         0
CustomerID    0
Country       0
dtype: int64
```

- The given code is used to find the number of missing values in column of a dataset.The sum() function is count the number of missing values.

#DataFrame:

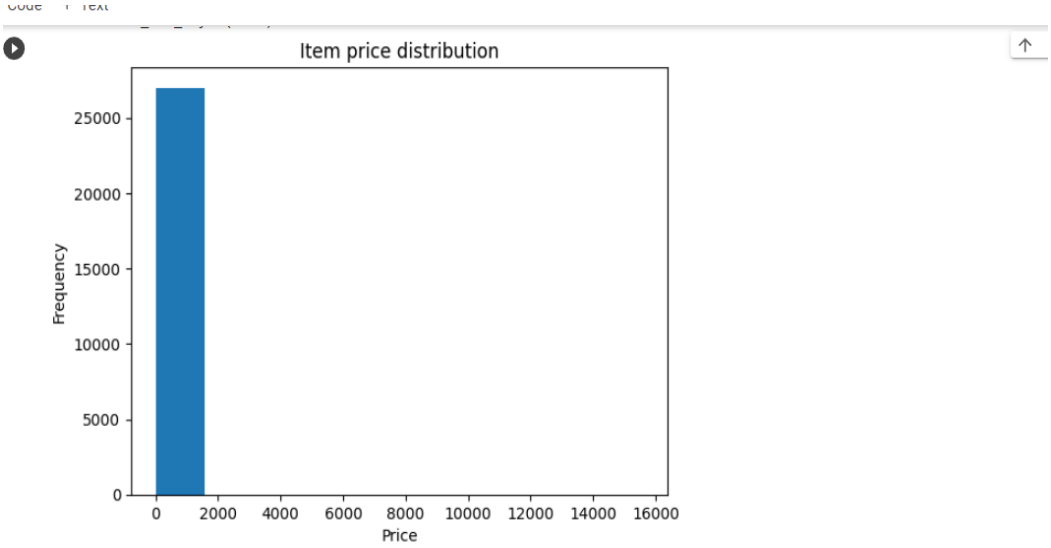
```
df2 = pd.DataFrame(df1)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWa
  and should_run_async(code)
```

- The code creates a new DataFrame called df2 by copying the contents of an existing DataFrame df1. The pd.DataFrame() function is used from the `pandas library to create the new DataFrame.

#Histogram:

```
item_prices = df2['Price']
item_prices.plot(kind='hist', bins=10)
plt.title('Item price distribution')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```



- This Python code reads the 'Price' column from a DataFrame called df2. It then creates a histogram plot of the item prices using the plot() function with the kind='hist' parameter and bins=10 to specify the number of bins in the histogram. The code sets the title of the plot to 'Item price distribution' and labels the x-axis as 'Price' and the y-axis as 'Frequency'. Finally, the plot is displayed using plt.show().

#Scatter Plot:

```
[23] plt.scatter(df1['Price'], df2['CustomerID'])
     plt.title('Item Price vs. Customer ID')
     plt.xlabel('Price')
     plt.ylabel('Customer ID')
     plt.show()
```

- The plt.scatter() function is used to create the scatter plot, where df1['Price'] represents the x-coordinates (Price) and df2['CustomerID'] represents the y-coordinates (Customer ID) of the data points.The plt.title() function sets the title of the plot to "Item Price vs. Customer ID", plt.xlabel() sets the label for the x-axis to "Price", and plt.ylabel() sets the label for the y-axis to "Customer ID".Finally, plt.show() is used to display the plot.

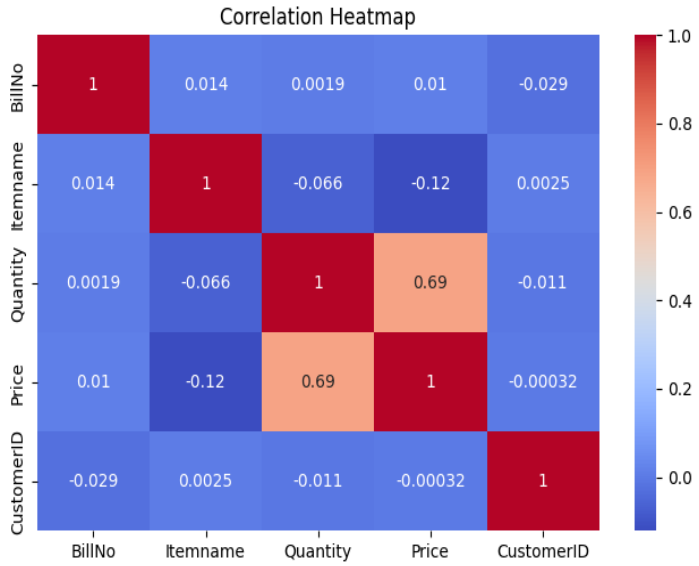
#Box plot:

```
df.boxplot(column='Price')
plt.title('Box Plot of Item Prices')
plt.show()
```

- The code is using the pandas library to create a box plot of the 'Price' column in a DataFrame called 'df'. It then sets the title of the plot to 'Box Plot of Item Prices' and displays the plot using the plt.show() function from the matplotlib library.

#Heatmap:

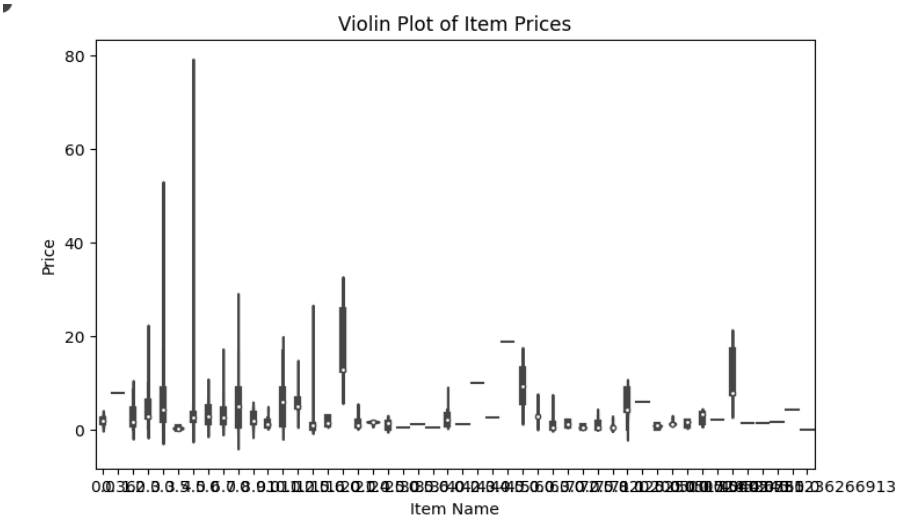
```
df2['Itemname'] = df2['Itemname'].astype(str).str.replace('[^0-9.]', '', regex=True)
df2['Itemname'] = pd.to_numeric(df2['Itemname'], errors='coerce')
plt.figure(figsize=(8, 5))
correlation_matrix = df2.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



- 1 It is using the .replace() method in thepandaslibrary to remove non-numeric characters from the 'Item' column of a DataFrame calleddf
- 2. regular expression [^0-9.]` is used to match any character that is not a digit or a dot. This operation converts the values in the 'Itemname' column to a string type.It is using the pd.to_numeric() function from the pandas library to convert the 'Itemname' column to numeric values. The errors='coerce' argument is used to replace non-numeric values with NaN (Not a Number).It is creating a new figure with a size of 8x5 using the plt.figure() function from the matplotlib.pyplot library.It is creating a correlation matrix of the DataFrame df2 using the corr() method.It is using the sns.heatmap() function from the seaborn library to create a heatmap of the correlation matrix. The argument annot=True is used to display the correlation values on the heatmap, and cmap='coolwarm' is used to set the color scheme of the heatmap.It is setting the title of the figure to 'Correlation Heatmap' using the plt.title() function.It is displaying the figure using the plt.show() function.

#Violin plot:

```
plt.figure(figsize=(8, 5))
sns.violinplot(x="Itemname", y="Price", data=df2)
plt.title('Violin Plot of Item Prices')
plt.xlabel('Item Name')
plt.ylabel('Price')
plt.show()
```

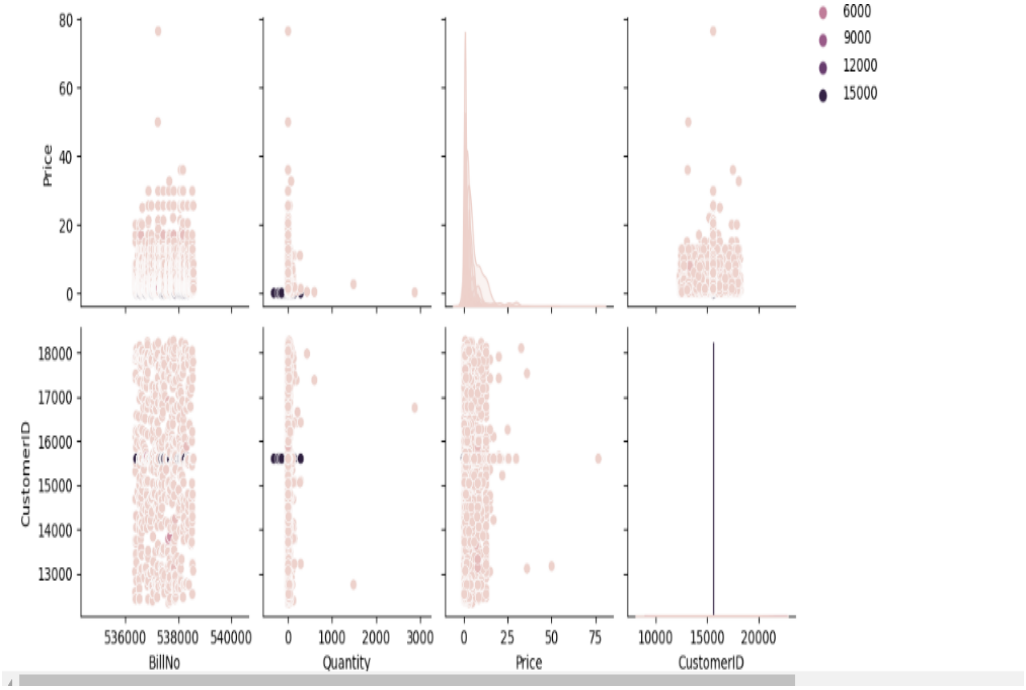


- The line sns.violinplot(x="Itemname", y="Price", data=df2) creates the actual violin plot. It uses the Itemname column as the x-axis variable and the Price column as the y-axis variable. The data used for the plot is stored in the variable df2.The line plt.title('Violin Plot of Item Prices') sets the title of the plot as "Violin Plot of Item Prices".The line plt.xlabel('Item Name') sets the label of the x-axis as "Item Name".The line plt.ylabel('Price') sets the label of the y-axis as "Price".The line plt.show() displays the plot.

#Pair Plot:

```
sns.pairplot(df2, hue="Itemname", diag_kind="kde")
plt.suptitle("Pair Plot of Marker Basket Data", y=1.02)
plt.show()
```

✓ 15s completed at 7:49PM



- The hue parameter specifies the column name in df2 that will be used to color the data points in the plot. In this case, it's "Itemname".The diag_kind parameter specifies the type of plot to use on the diagonal. In this case, it's a kernel density estimate (KDE) plot.After creating the pair plot, the code sets the plot title using plt.suptitle and shows the plot using plt.show().