

## Phase-2

### Data Preprocessing

#### Market Basket Analysis:

Date	08 Oct 2023
Team ID	Proj-212168-Team-1
Project Name	Market Basket Insights
Maximum Mark	

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining tasks.

#### Program:

#### #Import package:

## Explanation:

- 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.

```
In [20]: 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
```

0.002 seconds

•

```
In [21]: dataset=pd.read_csv('insights.csv')
Out[21]: /tmp/ipykernel_376/4147191086.py:1: DtypeWarning: Columns (0) have mixed types. Specify dtype
option on import or set low_memory=False.
dataset=pd.read_csv('insights.csv')
```

1.098 seconds Explain... Format Copy 5

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

- 

≡

In [22]:

dataset.head()

0.004 s

Out[22]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

The code `dataset.head()` is calling the `head()` function on the dataset object. The `head()` function is used to display the first few rows of a data set.

≡

In [26]:

dataset.isnull().sum()

Out[26]:

BillNo	0
Itemname	1455
Quantity	0
Date	0
Price	0
CustomerID	134041
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 used to count the number of missing values.



```
In [24]: dataset.info()
```

```
Out[24]: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   BillNo          522064 non-null object  
1   Itemname        520609 non-null object  
2   Quantity        522064 non-null int64   
3   Date            522064 non-null object  
4   Price           522064 non-null float64  
5   CustomerID      388023 non-null float64  
6   Country         522064 non-null object  
dtypes: float64(2), int64(1), object(4)
memory usage: 27.9+ MB
```

- The code `dataset.info()` is a method call in python to display the information about data set. The `.info()` method provides such as number of columns and rows datatypes of columns and memory usage of the dataset.

```
In [25]: df=dataset.fillna({'Itemname':'abcd'})
df
```

Out[25]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	12/9/2011 12:50	0.85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	12/9/2011 12:50	2.10	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	12/9/2011 12:50	4.15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	12/9/2011 12:50	4.15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	12/9/2011 12:50	4.95	12680.0	France

522064 rows x 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.

```
In [27]: df1=dataset.fillna(value=dataset['CustomerID'].mean())
df1
```

```
Out[27]:
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	12/9/2011 12:50	0.85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	12/9/2011 12:50	2.10	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	12/9/2011 12:50	4.15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	12/9/2011 12:50	4.15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	12/9/2011 12:50	4.95	12680.0	France

522064 rows x 7 columns

This code is fills the missing values in a dataframe calles dataset,using the mean of the "CustomerID" column.The filled dataframe than assigned variable df1and displayed.

```
In [28]: df1.isnull().sum()
```

```
Out[28]: 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.

- 

```
In [29]: print("Highest allowed",df1['Price'].mean()+3*df1['Price'].std())|  
         print("Lowest allowed",df1['Price'].mean()-3*df1['Price'].std())
```

0.01 seco

```
Out[29]: Highest allowed 129.52859810696216  
         Lowest allowed -121.87499535327679
```

This code is printing the highest and lowest allowed values based on statistical calculation. It calculates the mean and standard deviation of column called "price" in dataframe called df1.

•

```
In [30]: df1[(df1['Price']>129.52)|(df1['Price']<-121.87)]
```

Out[30]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
	237 536392	RUSTIC SEVENTEEN DRAWER SIDEBBOARD	1	12/1/2010 10:29	165.00	13705.00000	United Kingdom
	1781 536544	DOTCOM POSTAGE	1	12/1/2010 14:32	569.77	15316.93171	United Kingdom
	2994 536592	DOTCOM POSTAGE	1	12/1/2010 17:06	607.49	15316.93171	United Kingdom
	4897 536835	VINTAGE RED KITCHEN CABINET	1	12/2/2010 18:06	295.00	13145.00000	United Kingdom
	5348 536862	DOTCOM POSTAGE	1	12/3/2010 11:13	254.43	15316.93171	United Kingdom
	...	...	...	...	...	...	...
	517135 581219	DOTCOM POSTAGE	1	12/8/2011 9:28	1008.96	15316.93171	United Kingdom
	517534 581238	DOTCOM POSTAGE	1	12/8/2011 10:53	1683.75	15316.93171	United Kingdom
	519549 581439	DOTCOM POSTAGE	1	12/8/2011 16:30	938.59	15316.93171	United Kingdom
	521067 581492	DOTCOM POSTAGE	1	12/9/2011 10:03	933.17	15316.93171	United Kingdom
	521699 581498	DOTCOM POSTAGE	1	12/9/2011 10:26	1714.17	15316.93171	United Kingdom

668 rows x 7 columns

The code is filtering a dataframe df1 based on a condition.



```

In [31]: Q1=df1['Quantity'].quantile(0.25)
          Q3=df1['Price'].quantile(0.75)
          IQR=Q3-Q1
          lower_bound=Q1-1.5*IQR
          upper_bound=Q3+1.5*IQR
          outliers=df1[(df1['Quantity']<lower_bound)|(df1['Price']>upper_bound)]
          print(outliers)

Out[31]:
   BillNo  Itemname  Quantity  Date \
16  536367  BOX OF VINTAGE ALPHABET BLOCKS    2  12/1/2010 8:34
45  536370          POSTAGE    3  12/1/2010 8:45
65  536374  VICTORIAN SEWING BOX LARGE    32  12/1/2010 9:09
150  536382  3 TIER CAKE TIN GREEN AND CREAM    2  12/1/2010 9:45
151  536382  3 TIER CAKE TIN RED AND CREAM    2  12/1/2010 9:45
...      ...      ...      ...
521922  581574          POSTAGE    2  12/9/2011 12:09
521923  581578          POSTAGE    3  12/9/2011 12:16
521941  581578  BOX OF VINTAGE ALPHABET BLOCKS    6  12/9/2011 12:16
522004  581580  TABLECLOTH RED APPLES DESIGN    2  12/9/2011 12:20
522047  581586  RED RETROSPOT ROUND CAKE TINS   24  12/9/2011 12:49

   Price  CustomerID  Country
16    9.95    13047.0  United Kingdom
45   18.00    12583.0      France
65   10.95    15100.0  United Kingdom

```

- Q1 and Q3 are the first and third quartiles of the 'Quantity' and 'Price' columns, respectively.
- IQR is the interquartile range, calculated as the difference between Q3 and Q1.
- lower\_bound and upper\_bound are the lower and upper bounds, respectively, for identifying outliers. They are calculated as  $Q1 - 1.5 * IQR$  and  $Q3 + 1.5 * IQR$ .

- outliers is a DataFrame containing the rows from df1 where either the 'Quantity' is less than lower\_bound or the 'Price' is greater than upper\_bound.
- Finally, the code prints out the outliers DataFrame.

```
In [32]: df1=df1.drop('Country',axis=1)
print(df1)
```

```
Out[32]:
```

	BillNo	Itemname	Quantity
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	WHITE METAL LANTERN	6
2	536365	CREAM CUPID HEARTS COAT HANGER	8
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6
...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12
522060	581587	CHILDREN'S APRON DOLLY GIRL	6
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4
522063	581587	BAKING SET 9 PIECE RETROSPOT	3

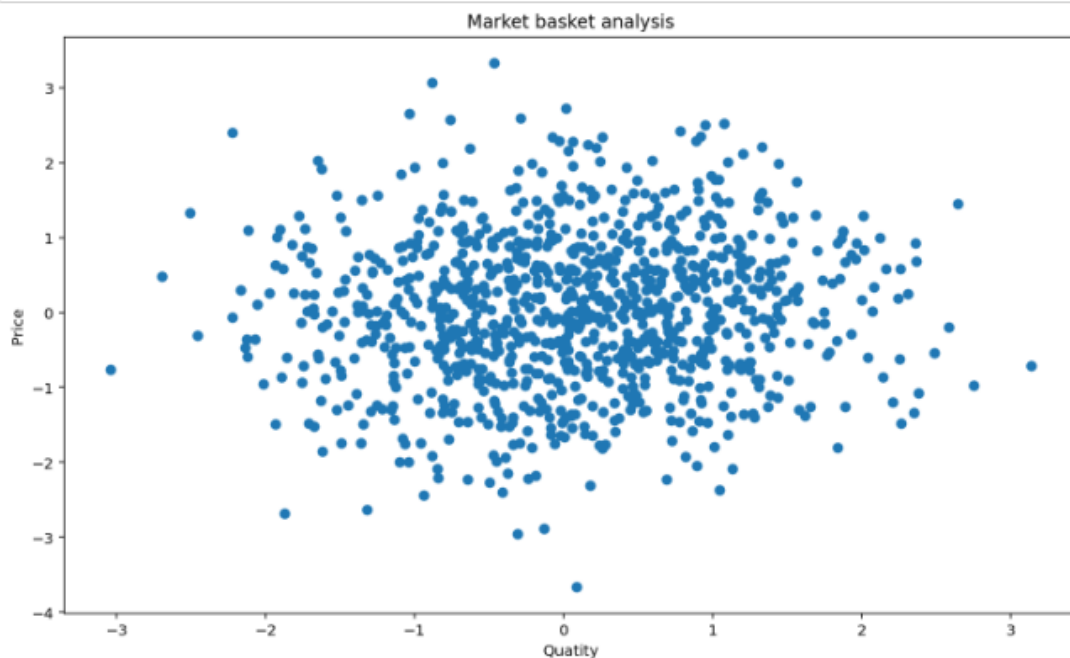
  

	Date	Price	CustomerID
0	12/1/2010 8:26	2.55	17850.0
1	12/1/2010 8:26	3.39	17850.0
2	12/1/2010 8:26	2.75	17850.0
3	12/1/2010 8:26	3.39	17850.0
4	12/1/2010 8:26	3.39	17850.0
...	...	...	...
522059	12/9/2011 12:50	0.85	12680.0
522060	12/9/2011 12:50	2.10	12680.0
522061	12/9/2011 12:50	4.15	12680.0
522062	12/9/2011 12:50	4.15	12680.0
522063	12/9/2011 12:50	4.95	12680.0

[522064 rows x 6 columns]

- The code is using the pandas library in Python to drop the 'Country' column from a DataFrame called df1. The 'axis=1' parameter specifies that the column is being dropped.
- After dropping the column, the code then prints the updated DataFrame.

```
In [33]: x=np.random.normal(0,1,1000)
y=np.random.normal(0,1,1000)
plt.scatter(x,y)
plt.xlabel('Quatity')
plt.ylabel('Price')
plt.title('Market basket analysis')
plt.show()
```



- This code generates two arrays of random numbers with a normal distribution, assigns them to the variables x and y, plots them as a scatterplot using the scatter() function from the pyplot module of the matplotlib library, adds labels to the x-axis and y-axis, sets a title the plot, and displays the plot.