NAAN MUDHALVAN

PHASE 5-DOCUMENTATION AND SUBMISSION

NAME: Samyuktha S R

REGISTER NUMBER: 422521104032

DEPARTMENT: Computer Science and Engineering

NAAN MUDHALVAN ID: 58B1628D6DE2E9F8D314BA2A251ED289

DOMAIN: Artificial Intelligence(AI)

PROJECT TITLE: Market Basket Insights

COLLEGE NAME: University College of Engineering Villupuram

COLLEGE CODE: 4225

DOCUMENTATION

PROBLEM DEFINITION:

The problem is to perform market basket analysis on a provided dataset to unveil hidden patterns and associations between products. The goal is to understand customer purchasing behaviour and identify potential cross-selling opportunities for a retail business. This project involves using association analysis techniques, such as Apriori algorithm, to find frequently co-occurring products and generate insights for business optimization.

DESIGN THINKING:

1. **Data Source:** Choose a dataset containing transaction data, including lists of purchased products.
2. **Data Preprocessing:** Prepare the transaction data by transforming it into a suitable format for association analysis.
3. **Association Analysis:** Utilize the Apriori algorithm to identify frequent itemsets and generate association rules.
4. **Insights Generation:** Interpret the association rules to understand customer behaviour and cross-selling opportunities.
5. **Visualization:** Create visualizations to present the discovered associations and insights.
6. **Business Recommendations:** Provide actionable recommendations for the retail business based on the insights.

PHASES OF DEVELOPMENT:

Phase 2: Innovation

We explore innovative techniques such as ensemble methods and deep learning architectures to improve the prediction system's accuracy and robustness. Then we consider exploring advanced association analysis techniques or using visualization tools for enhanced insights presentation.

Phase 3: Development Part-I

In this part webuilt our project by loading and preprocessing the dataset for association analysis.

Phase 4: Development Part-II

In this part we built our "market basket insights" project by:

- Performing association analysis
- Generating insights.

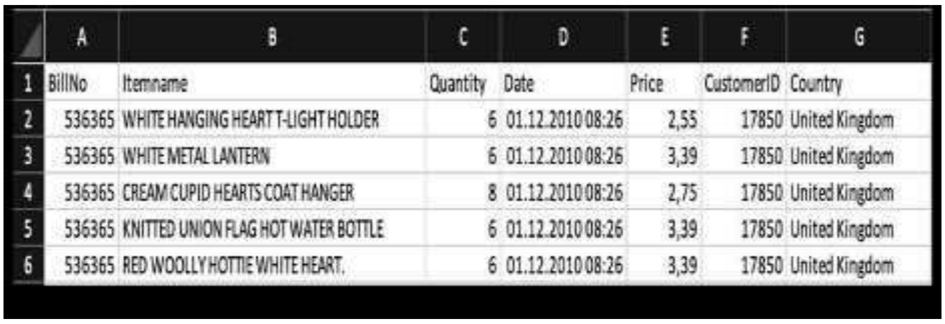
DATASET

Dataset Link:

https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis

Dataset Description:

- *File name:* Assignment-1_Data
- *List name:*retaildata
- *File format:* . xlsx
- *Number of Row:* 522065
- *Number of Attributes:* 7
- *Bill No:* 6-digit number assigned to each transaction. Nominal.
- *Item name:* Product name. Nominal.
- *Quantity:* The quantities of each product per transaction. Numeric.
- *Date:* The day and time when each transaction was generated.
- *Numeric Price:* Product price. Numeric.
- *Customer ID:* 5-digit number assigned to each customer. Nominal.
- *Country:* Name of the country where each customer resides. Nominal



	A	B	C	D	E	F	G
1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850	United Kingdom
3	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850	United Kingdom
4	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850	United Kingdom
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850	United Kingdom
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850	United Kingdom

PROJECT DEVELOPMENT

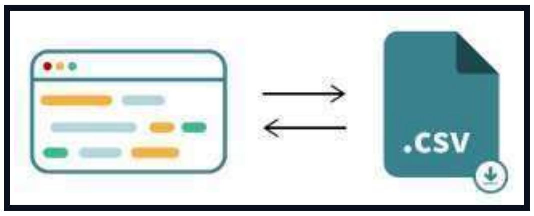
NECESSARY PACKAGES

#Importing the necessary packages

```
import pandas as pd

fromsklearn.preprocessingimportMinMaxScaler

importnumpyas np
fromsklearn.decompositionimport PCA
fromsklearn.preprocessingimportStandardScaler
frommlxtend.frequent_patternsimportapriori
frommlxtend.frequent_patternsimportassociation_rules
importmatplotlib.pyplotasplt
import seaborn assns
```

DATA LOADING:

Data loading refers to the process of importing data from one or more sources into a database, data warehouse, or other data storage system.



#Loading the dataset

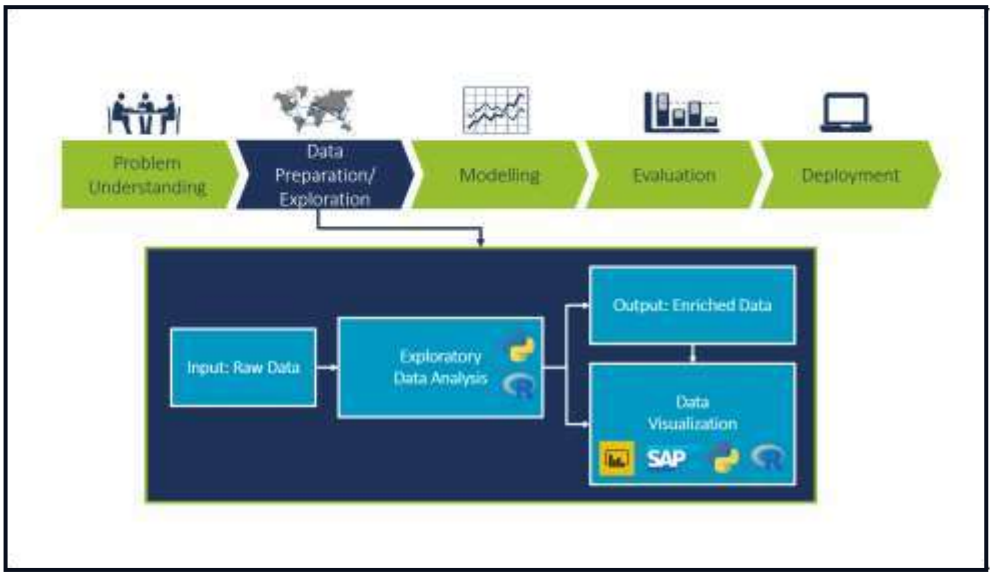
```
data=pd.read_csv('/content/Assignment-1_DataN.csv')
data.head()   #viewing the data
```

Output

```
BillNoQuantity Price CustomerID
0 536365 6 2.55 17850.0
1 536365 63.39 17850.0
2 536365 8 2.75 17850.0
3 536365 6 3.39 17850.0
```

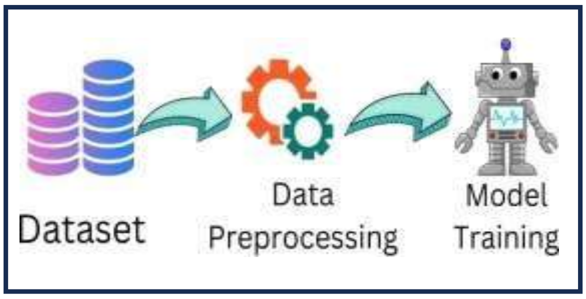
EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is the process of visually and statistically examining a dataset to summarize its main characteristics, often with the help of graphs and descriptive statistics. The primary goal of EDA is to gain insights, discover patterns, identify anomalies, and generate hypotheses that can guide further data analysis or research. It's a crucial initial step in data analysis to better understand the data's structure and potential relationships between variables.

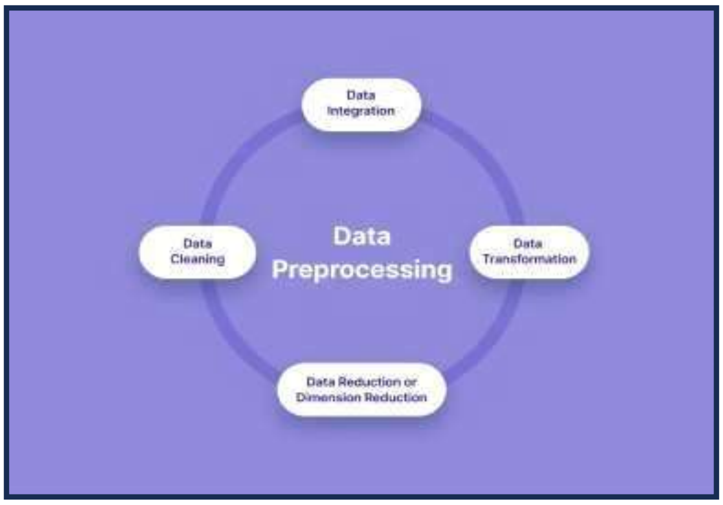


DATA PRE-PROCESSING

Data preprocessing can be defined as the process of transforming raw data intoa form that can be easily understood and analyzed by a machine learning algorithm. Data preprocessing involves various steps such as removing irrelevant data, dealing with missing values, dealing with outliers, scaling the data, and encoding categorical variables.



The following are the types of data processing



(i) Data cleaning:

The process of detecting and correcting (or removing) invalid or irrelevant records from the dataset.

- ✓ Removal of Unwanted Observations.
- ✓ Managing Unwanted Outliers.
- ✓ Fixing Structural Error
- ✓ Handling Structural Data.

Cleaning the data:

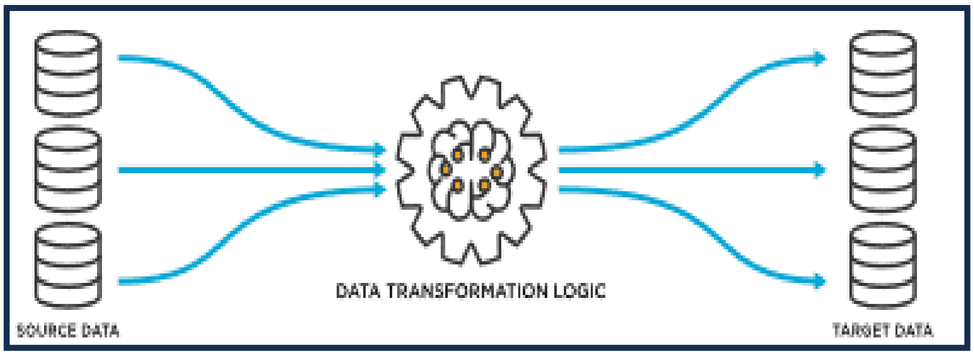
○Identify the data quality problems

○Prioritize the data quality problems.

○Validate the data.

(ii) Data integration:

Merging multiple datasets into one for analysis.



(iii) Data transformation:

The process of converting data from one form to another

(iv) Data reduction:

- The process of reducing the amount of data by aggregating it or selecting a subset of relevant features.
- The process of converting continuous variables into categorical variables by dividing them into intervals.
- The process of scaling the features or attributes of a dataset to the same range to avoid the dominance of any particular feature.
- Data preprocessing is essential to ensure that the data is accurate, complete, and suitable for machine learning algorithms to produce accurate and reliable results.

PERFORMIG ASSOCIATION ANALYSIS

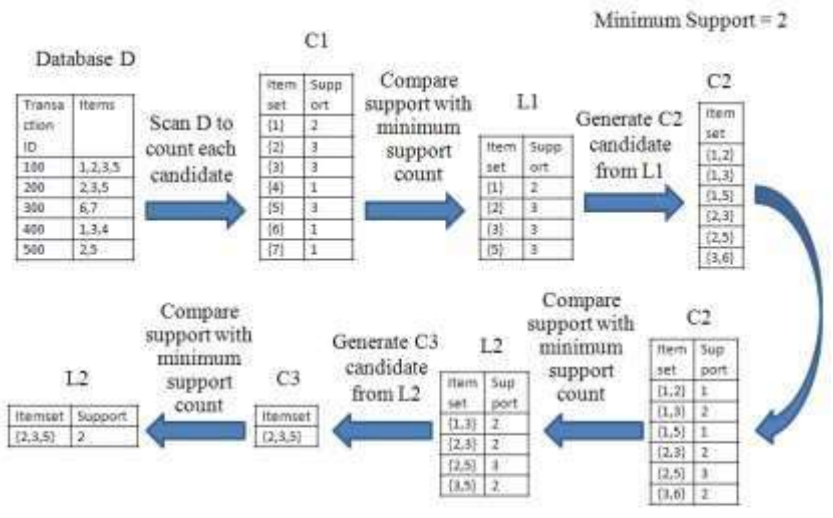
To perform association analysis, you can follow these steps:

1. ***Collect market basket data***- This data should include all of the items that customers purchase together in each transaction. You can collect this data from your point-of-sale system, loyalty program, or other customer data sources.

2. ***Clean and prepare the data***- This may involve removing incomplete or inaccurate data and formatting the data in a consistent way.

3. ***Identify frequent item sets***- This is the process of finding sets of items that are frequently purchased together. You can use a variety of algorithms to do this, such as the Apriori algorithm or the FP Growth algorithm

```
frequent_itemsets = apriori(binary_matrix_df, min_support=min_support, use_colnames=True
```


Minimum Support = 2
Database D
Scan D to count each candidate
C1
Compare support with minimum support count
L1
Generate C2 candidate from L1
C2
C2
Compare support with minimum support count
L2
Generate C3 candidate from L2
C3
Compare support with minimum support count
L2

Transaction ID	Items
100	1,2,3,5
200	2,3,5
300	6,7
400	1,3,4
500	2,5

Item set	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3
{6}	1
{7}	1

Item set	Support
{1}	2
{2}	3
{3}	3
{5}	3

Item set
{1,2}
{1,3}
{1,5}
{2,3}
{2,5}
{3,6}

Item set	Sup port
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,6}	2

Item set	Sup port
{1,3}	2
{2,3}	2
{2,5}	3
{3,5}	2

Itemset
{2,3,5}

Itemset	Support
{2,3,5}	2

4. ***Generate association rules***- Association rules are rules that show the relationship between two or more items. For example, an association rule might be "If a customer purchases milk, they are 80% likely to also purchase bread." You can use a variety of algorithms to generate association rules, such as the confidence-lift algorithm.

5. ***Identify strong association rules***- Strong association rules are rules that have a high confidence score and a high support count. Confidence score measures how likely it is purchase baby wipes and formula. This insight could be used to develop targeted promotions or product recommendations.

So,the customer will purchase one item if they purchase another item. Support count measures how often two items are purchased together.

GENERATING INSIGHTS

To generate insights in market basket, you can follow these steps:

1. Identify the strong association rules. Strong association rules are rules that have a high confidence score and a high support count. Confidence score measures how likely it is that a customer will purchase one item if they purchase another item. Support count measures how often two items are purchased together.

2. Analyse the strong association rules to identify patterns in customer behaviour. For example, you might find that customers who purchase diapers are also likely to purchase baby wipes and formula. This insight could be used to develop targeted promotions or product recommendations.

3. Use the insights to improve your business. For example, you could use the insights to improve your product placement, create new product bundles, or develop personalized marketing campaigns.

MARKET BASKET ANALYSIS
A
B
98% of people who purchased items A and B
also purchased item C

CODING PART

#importing the necessary libraries

```
import pandas aspd
fromsklearn.preprocessingimportMinMaxScaler
importnumpyasnp
fromsklearn.decompositionimport PCA
fromsklearn.preprocessingimportStandardScaler
```

#Loading the dataset

```
data=pd.read_csv('/content/Assignment-1_DataN.csv')
data.head()   #viewing the data
```

output

```
<ipython-input-7-3fa16f8c979c>:1: DtypeWarning: Columns (0) have mixed types. Specify
dtype option on import or set low_memory=False.
  data=pd.read_csv('/content/Assignment-1_DataN.csv')
```

```
BillNoQuantity Price CustomerID
     0   536365          6    2.55       17850.0
     1   536365          6    3.39       17850.0
     2   536365          8    2.75       17850.0
     3   536365          6    3.39       17850.0
     4   536365          6    3.39       17850.0
```

```
data.tail()    #Viewing the end of the dataset
```

output

```
               BillNo   Quantity   Price   CustomerID
     522059    581587         12    0.85      12680.0
     522060    581587          6    2.10      12680.0
     522061    581587          4    4.15      12680.0
     522062    581587          4    4.15      12680.0
     522063    581587          3    4.95      12680.0
```

#information about dataset

```
data.info()
```

output

```
<class
'pandas.core.frame.DataFrame'>RangeIn
dex: 522064 entries, 0 to 522063 Data
columns (total 4 columns):
```

```
columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
--  -------     ---------   ----
0   BillNo      522064 non-null  object
1   Quantity    522064 non-null int64
2   Price       522064 non-null float64 3 CustomerID 388023 non-null float64
dtypes: float64(2), int64(1), object(1)
memory usage: 15.9+ MB
CodeText
```

#Counting the number of Data

```
data.count()
```

output

```
BillNo 522064

       Quantity 522064

       Price 522064

       CustomerID 388023

       dtype: int64
```

#Printing the attribute

```
data.BillNo
```

output

```
0 536365

1 536365

2 536365

3 536365

4 536365

   ...

522059 581587

522060 581587

522061 581587

522062 581587

522063 581587

   Name: BillNo, Length: 522064, dtype: object
```

#type of the data

```
type(data)
```

output

```
pandas.core.frame.DataFrame
```

#printing the shape

```
data.shape
```

output

```
(419475, 4)
```

PRE-PROCESSING

(i) Cleaning

#Handling Missing Data.

```
data['Quantity'].fillna(data['Quantity'].mean(),inplace=True)

data['Price'].fillna(data['Price'].mean(),inplace=True)
```

#Removes the null value

```
print(data.isnull().sum())
```

output

```
BillNo          0
Quantity        0
Price           0
CustomerID  40749
dtype: int64
```

#Encoding the categorical data

```
data = pd.get_dummies(data, columns=['BillNo'], prefix=['BillNo'])
data = pd.get_dummies(data, columns=['Quantity'], prefix=['Quantity'])
```

#Handling the duplicates

```
data.drop_duplicates(inplace=True)
```

(ii)Data Integration

#split and load the data set

```
data=pd.read_csv('/content/Assignment-1_DataN.csv')
data1=pd.read_csv('/content/Assignment-1_DataM.csv')
```

```
<ipython-input-18-c6fb65c16250>:1: DtypeWarning: Columns (0) have mixed
types. Specify dtype option on import or set low_memory=False
data=pd.read_csv('/content/Assignment-1_DataN.csv')
```

#convert the datasets to data frame

```
data = pd.DataFrame(data)
 data1= pd.DataFrame(data1)
```

#Merging the dataset

```
merged_data = pd.merge(data, data1, on='BillNo')
```

#Printing the merged dataset

```
print(merged_data)
```

output

```
BillNo Quantity Price CustomerID  \
0        536365        6   2.55     17850.0
1        536365        6   2.55     17850.0
2        536365        6   2.55     17850.0
3        536365        6   2.55     17850.0
4        536365        6   2.55     17850.0
...       ...      ...   ...       ...
14256971    545334        2   1.55     15750.0
14256972    545334        2   1.55     15750.0
14256973    545334        2   1.55     15750.0
14256974    545334        2   1.55     15750.0
14256975    545334        2   1.55     15750.0

    Itemname       Date        Country
    0              WHITE HANGING HEART T-LIGHT HOLDER 01-12-2010 United Kingdom
    1              WHITE METAL LANTERN 01-12-2010 United Kingdom
    2              CREAM CUPID HEARTS COAT HANGER 01-12-2010 United Kingdom
    3              KNITTED UNION FLAG HOT WATER BOTTLE 01-12-2010 United
                   Kingdom
    4              RED WOOLLY HOTTIE WHITE HEART. 01-12-2010 United Kingdom
    ...                          ...        ...            ...
```

```
Itemname       Date        Country
5                 WHITE HANGING HEART T-LIGHT HOLDER 01-12-2010 United Kingdom
6                 WHITE METAL LANTERN 01-12-2010 United Kingdom
7                 CREAM CUPID HEARTS COAT HANGER 01-12-2010 United Kingdom
8                 KNITTED UNION FLAG HOT WATER BOTTLE 01-12-2010 United
                  Kingdom
9                 RED WOOLLY HOTTIE WHITE HEART. 01-12-2010 United Kingdom
...                        ...      ...        ...
14256971            PACK OF 6 SANDCASTLE FLAGS ASSORTED 01-03-2011  United
                    Kingdom
14256972            EASTER CRAFT 4 CHICKS 01-03-2011 United Kingdom
14256973            FELTCRAFT BUTTERFLY HEARTS 01-03-2011 United Kingdom
14256974            3 STRIPEY MICE FELTCRAFT 01-03-2011 United Kingdom
14256975            BROWN  PIRATE TREASURE CHEST  01-03-2011       United K

14256976            rows x 7 columns]
```

(iv)Data Reduction

```
pca = PCA(n_components=2)
# Fit and transform your data
reduced_data =pca.fit_transform(data)
```

PERFORMING THE ASSOCIATION RULES

#importing the necessary packages

```
frommlxtend.frequent_patternsimportapriorifrom
mlxtend.frequent_patternsimportassociation_rule
simportnumpyas np
```

#Loading the dataset

```
datanew=pd.read_csv('/content/dataset_it.csv')
datanew.head()
```

output

```
/usr/local/lib/python3.10/dist -packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during        thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above. and
should_run_async(code)
```

<ipython-input-17-cd01a968ea79>:1: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set low_memory=False.
    datanew=pd.read_csv('/content/dataset_it.csv')

	BillNo	Itemname	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	United Kingdom
1	536365	WHITE METAL LANTERN	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	United Kingdom

```
# Convert the market basket dataset to a binary matrix
binary_matrix = data.to_numpy().astype(bool)
    binary_matrix_df = pd.DataFrame(binary_matrix)

#Findng the frequent itemset for analysis
    frequent_itemsets = apriori(binary_matrix_df, min_support=min_support,
    use_colnames=True)

#Finding the association rules for analysis

    association_rules = association_rules(frequent_itemsets, metric="confidence",
    min_threshold=min_confidence)


#Sort the association by decreasing values of confidence,support
    association_rules = association_rules.sort_values(by=['confidence', 'support'],
    ascending=False)
#Printing the association rules

    print(association_rules)
```

output

```
   antecedents consequents  antecedent support  consequent support   support \
0        (0)        (1)          1.000000           1.000000  1.000000
1        (1)        (0)          1.000000           1.000000  1.000000
4        (0)        (3)          1.000000           1.000000  1.000000
5        (3)        (0)          1.000000           1.000000  1.000000
8        (1)        (3)          1.000000           1.000000  1.000000
9        (3)        (1)          1.000000           1.000000  1.000000
```

```
 18        (0, 1)      (3)          1.000000          1.000000  1.000000
 19        (0, 3)      (1)          1.000000          1.000000  1.000000
 20        (1, 3)      (0)          1.000000          1.000000  1.000000
 21        (0)     (1, 3)        1.000000          1.000000  1.000000
 22        (1)     (0, 3)        1.000000          1.000000  1.000000
      (3)     (0, 1)        1.000000          1.000000  1.000000
 3         (2)        (0)         0.994569          1.000000  0.994569
7          (2)        (1)         0.994569          1.000000  0.994569
 10   (2)        (3)         0.994569          1.000000  0.994569
(0, 2)        (1)         0.994569          1.000000  0.994569

 13
 14     (1, 2)        (0)         0.994569          1.000000  0.994569
17   (2)     (0, 1)         0.994569          1.000000
    0.994569    24     (0, 2)        (3)         0.994569
    1.000000  0.994569

    26     (2, 3)        (0)         0.994569          1.000000  0.994569
   28   (2)     (0, 3)               0.994569          1.000000  0.994569


   …… ……………….

   confidence lift leverage conviction  zhangs_metric
    0     1.000000   1.0       0.0        inf           0.0
    1     1.000000   1.0       0.0        inf           0.0
    4     1.000000   1.0       0.0        inf           0.0
```

GENERATING THE INSIGHTS

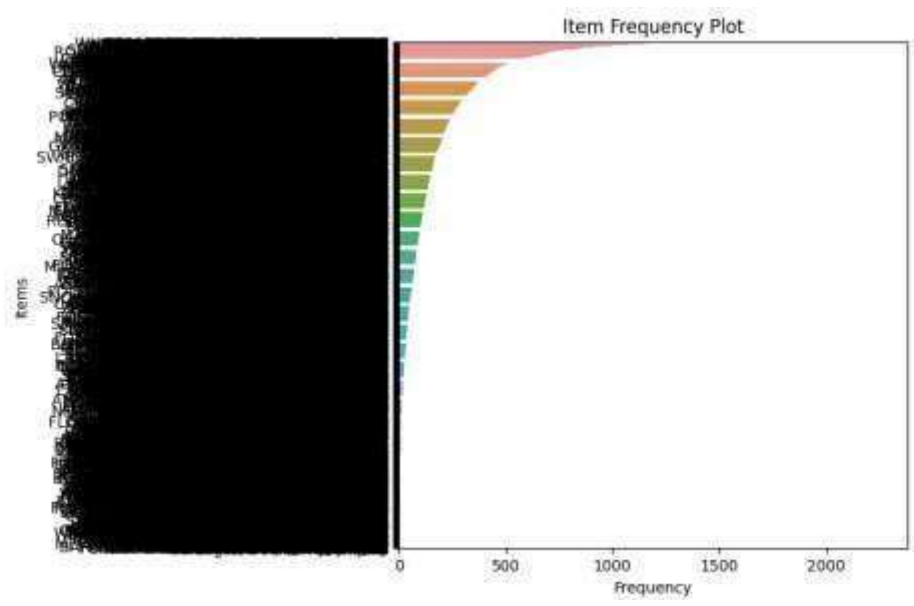
#Importing the packages for Visualisation

```
importmatplotlib.pyplotasplt
import seaborn assns
```

#Interactive Visualisation with plot

```
item_counts = datanew.explode('Itemname')['Itemname'].value_counts()

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

sns.barplot(x=item_counts.values,
y=item_counts.index) plt.xlabel('Frequency')
plt.ylabel('Items')
plt.title('Item Frequency Plot') plt.show()
```

Output



#Visualising with scatter plot

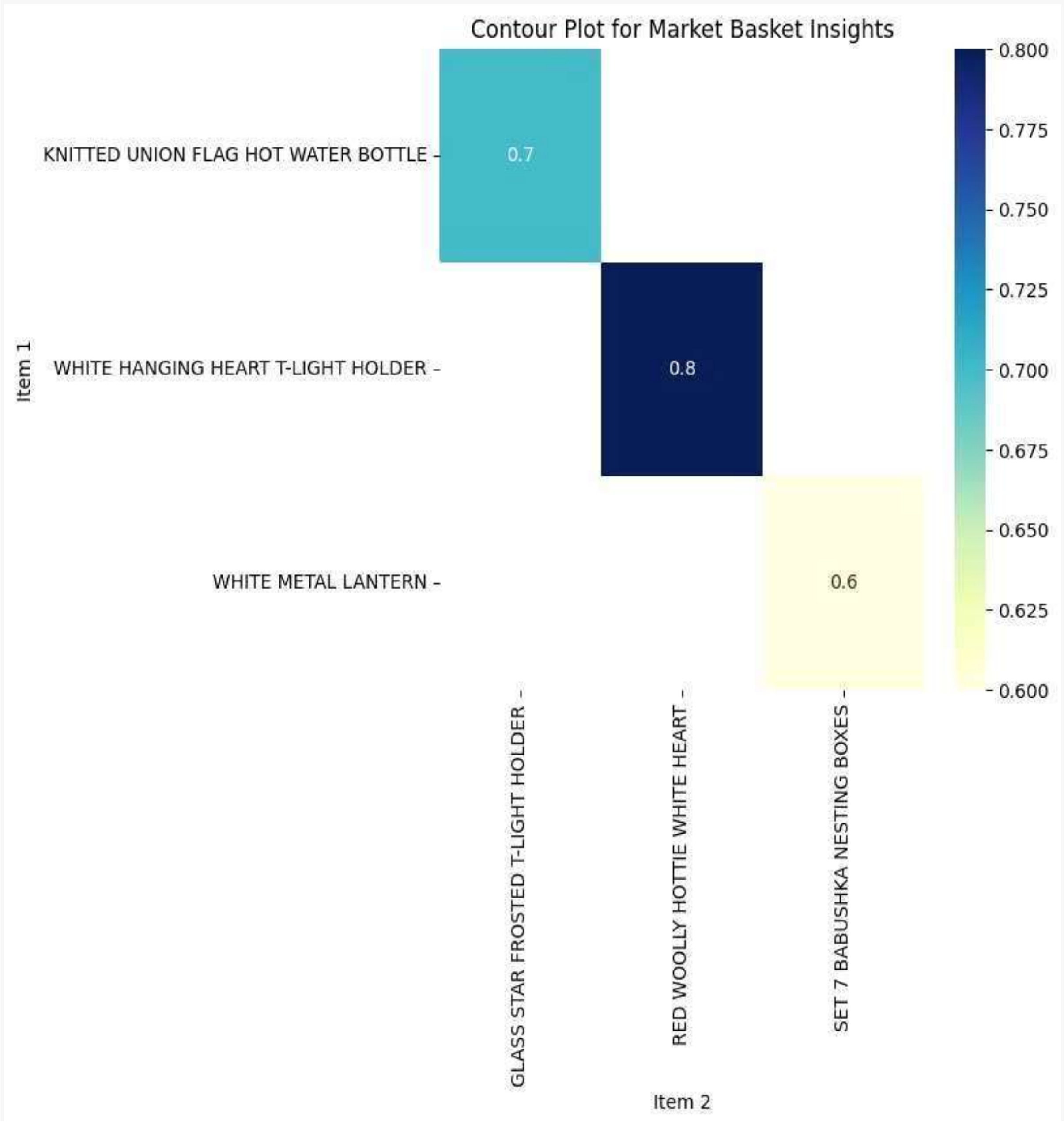
```
item_counts = datanew.explode('Itemname')['Itemname'].value_counts()
plt.figure(figsize=(6, 6)) sns.scatterplot(x=item_counts.values,
y=item_counts.index) plt.xlabel('Frequency') plt.ylabel('Items')
plt.title('Item Frequency Plot')
plt.show()
```

Output

#Visualising through contour plots

```
   # Pivot the data to create a matrix for the contour plot
pivot_data = data.pivot('Item1', 'Item2', 'Confidence')

# Create the contour plot using seaborn
plt.figure(figsize=(6, 6))
sns.heatmap(pivot_data, annot=True, cmap='YlGnBu')

# Customize labels and title
plt.xlabel('Item 2')
plt.ylabel('Item 1')
plt.title('Contour Plot for Market Basket Insights')

# Show the plot
        plt.show()
```

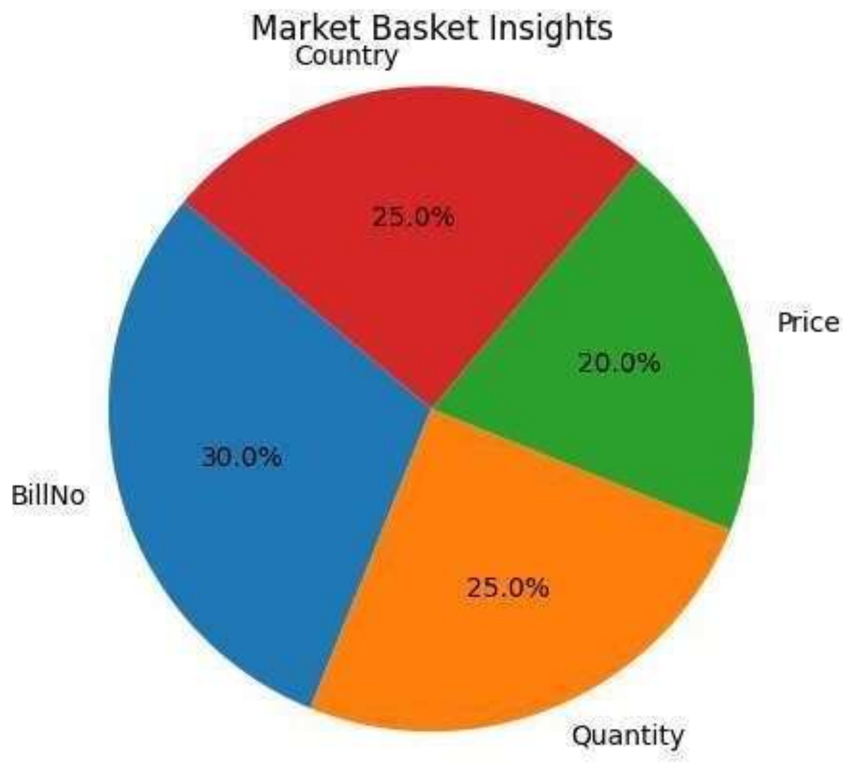
Output



#Visualising through pie-chart

```
# Sample data (replace with your market basket insights) labels
= ['BillNo', 'Quantity', 'Price', 'Country']
sizes = [30, 25, 20, 25] # Replace with your actual data

# Create a pie chart
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)

# Add a title
plt.title('Market Basket Insights')

# Display the pie chart
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a
circle.plt.show()
```

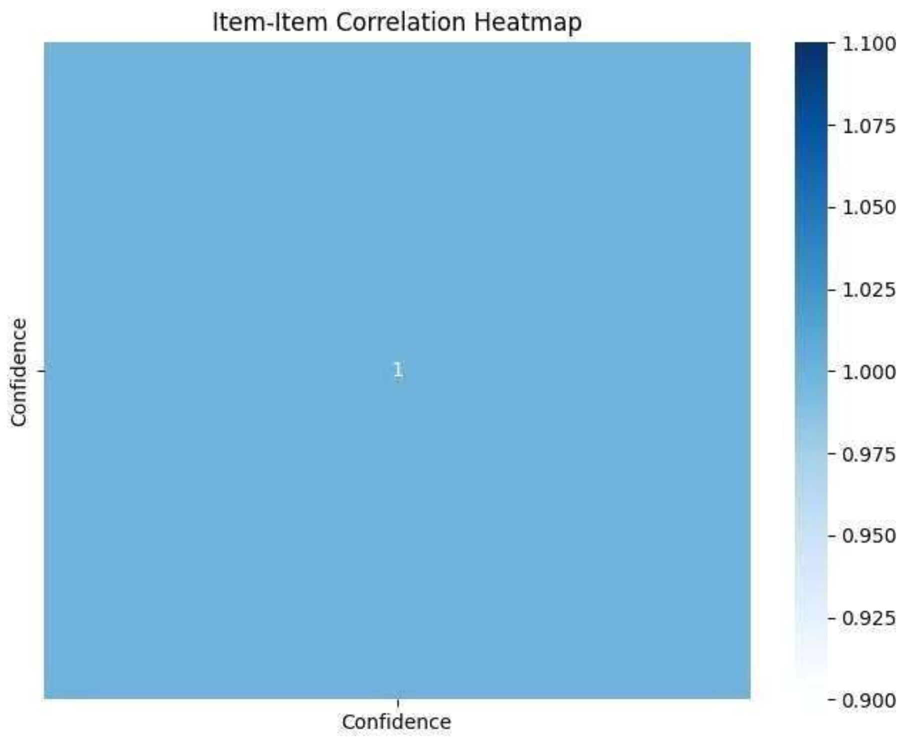
output



# Create a heatmap of item-item relationships

```
item_corr = data.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(item_corr, annot=True, cmap='Blues')
plt.title('Item-Item Correlation Heatmap')
plt.show()
```

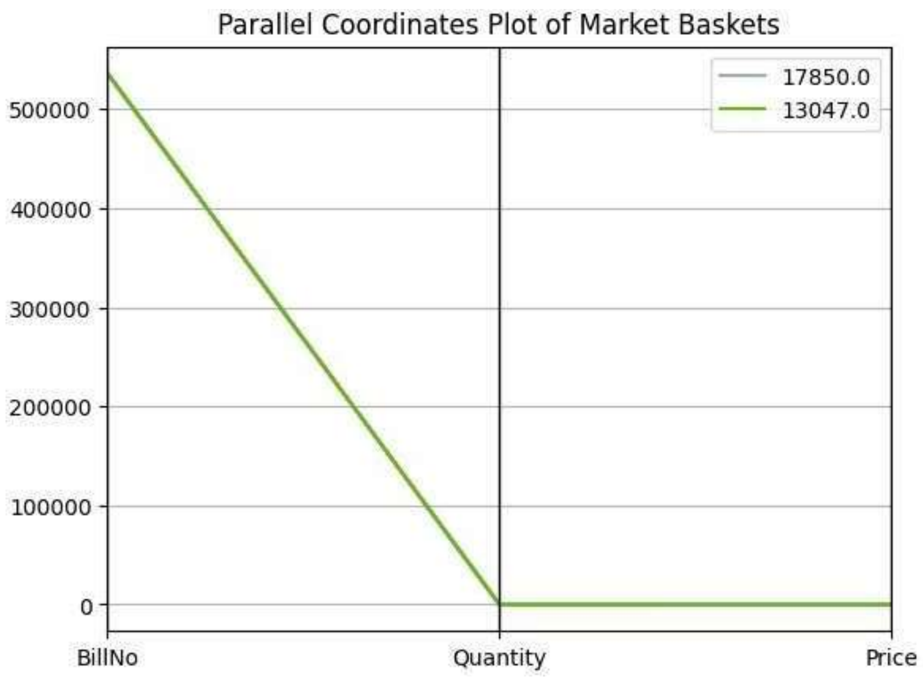
Output



#Plotting with Parellel Coordinators

```
frompandas.plottingimportparallel_coordinates

# Example: Plot the first 10 transactions
parallel_coordinates(data.head(10), 'CustomerID')
plt.title('Parallel Coordinates Plot of Market Baskets')
plt.show()
```

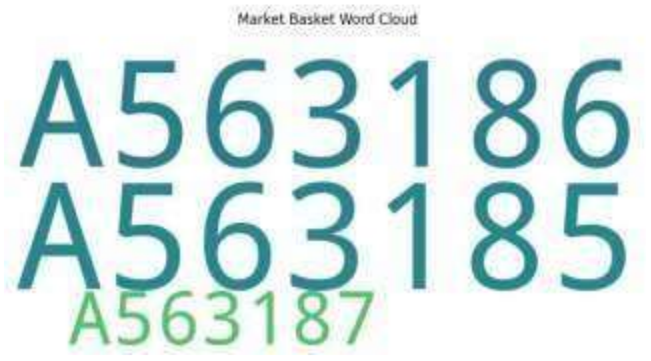
Output



#Visualise through WordCloud

```
from wordcloud import WordClouditem_frequency = '
'.join(data['BillNo'].explode().astype(str))

wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(item_frequency)

plt.figure(figsize=(10, 5))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.title('Market Basket Word Cloud')

plt.show()
```

Output



BUSINESS IMPLICATION :

The Apriori algorithm has several business implications in the context of market basket analysis:

1. Cross-Selling Opportunities: Apriori helps identify associations between products. Businesses can use this information to cross-sell related items to customers. For example, if customers who buy coffee also tend to buy creamer, a store can place these items near each other to increase sales.

2. Inventory Management: It aids in optimizing inventory by identifying which items are frequently bought together. This can help businesses stock their shelves more efficiently and reduce carrying costs.

3. Pricing Strategies: Businesses can adjust pricing strategies based on associations. For example, offering discounts on products that are often purchased together can boost sales.

4. Targeted Marketing: Apriori can assist in creating targeted marketing campaigns. If a customer buys a product that is typically part of a bundle, businesses can send promotions or recommendations for related items.

5. Store Layout: Retailers can use Apriori to optimize the layout of their stores. Products that are frequently purchased together can be placed closer to each other for convenience.

6. Customer Segmentation: It helps segment customers based on their purchase patterns. Businesses can tailor marketing and loyalty programs to specific customer segments.

7. Supply Chain Optimization: Understanding the associations between products can help in optimizing the supply chain. This can lead to cost savings and improved efficiency.

8. Fraud Detection: Apriori can also be used to detect unusual purchasing patterns. If a customer's basket contains items that are rarely purchased together, it might be a sign of fraudulent activity.

9. Personalization: E-commerce platforms can use Apriori results to offer personalized product recommendations, enhancing the customer experience.

10. Market Basket Insights: Overall, the Apriori algorithm enables businesses to gain valuable insights into consumer behavior and preferences, which can be leveraged for strategic decision-making, improving customer satisfaction, and increasing revenue.

Implementing the Apriori algorithm effectively requires understanding the data, interpreting results, and integrating findings into various aspects of the business, from marketing to operations.

CODE LINK

https://colab.research.google.com/drive/1jImYELLpsmnmpI-oE5uyVsSlUmvaGb3Q#scrollTo=ujyGmOOTqiJK