IBM PHASE 4 PROJECT SUBMISSION

NAME: Samyuktha S R

ROLL NO: 422521104032

TITLE: Market Basket Insights

DOMAIN: Artificial Intelligence

COLLEGE NAME: University College of engineering Villupuram

COLLEGE CODE:4225

PHASE 4 – DEVELOPMENT PART 2

1.PERFORMING 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)
```

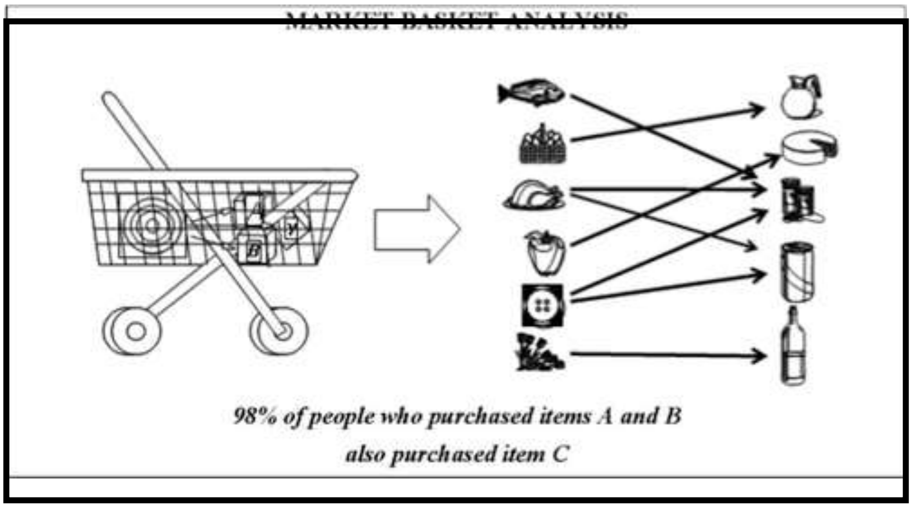
4.Generate association rules- Association rules are rules that show the relationship between two or more items. For example, an association rule might be "If a customer purchases milk, they are 80% likely to also purchase bread." You can use a variety of algorithms to generate association rules, such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

2.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.



Specific examples on this,

- Product placements
- Product bundles
- Personalised shopping

CODING PART:

(1)Performing the association rules

#importing the necessary packages

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import numpy as 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
23       (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
13    (0, 2)         (1)            0.994569            1.000000  0.994569
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
```

5	1.000000	1.0	0.0	inf	0.0
8	1.000000	1.0	0.0	inf	0.0
9	1.000000	1.0	0.0	inf	0.0
18	1.000000	1.0	0.0	inf	0.0
19	1.000000	1.0	0.0	inf	0.0
20	1.000000	1.0	0.0	inf	0.0
21	1.000000	1.0	0.0	inf	0.0
22	1.000000	1.0	0.0	inf	0.0

(2)Generating the Insights

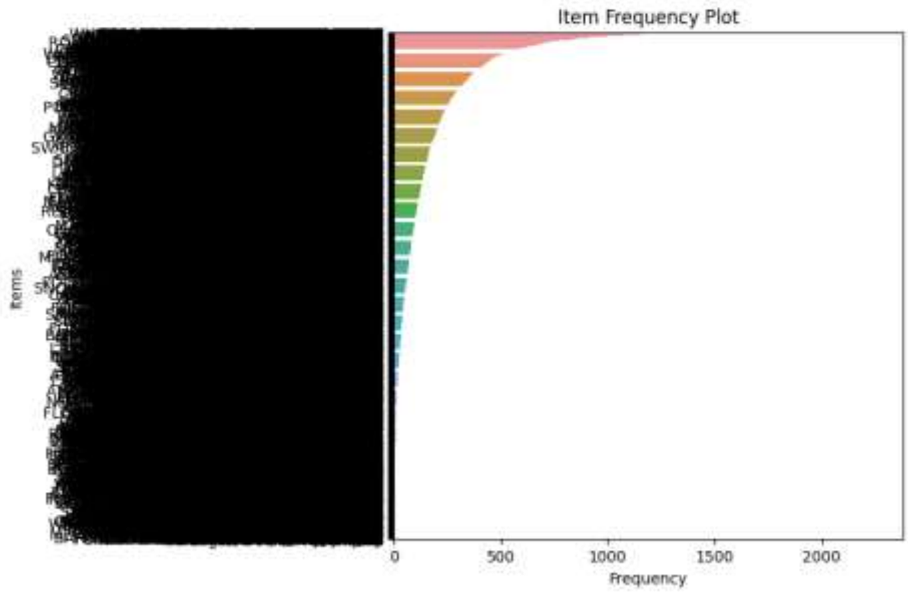
#Importing the packages for Visualisation

```
import matplotlib.pyplot as plt
import seaborn as sns
```

#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()
```

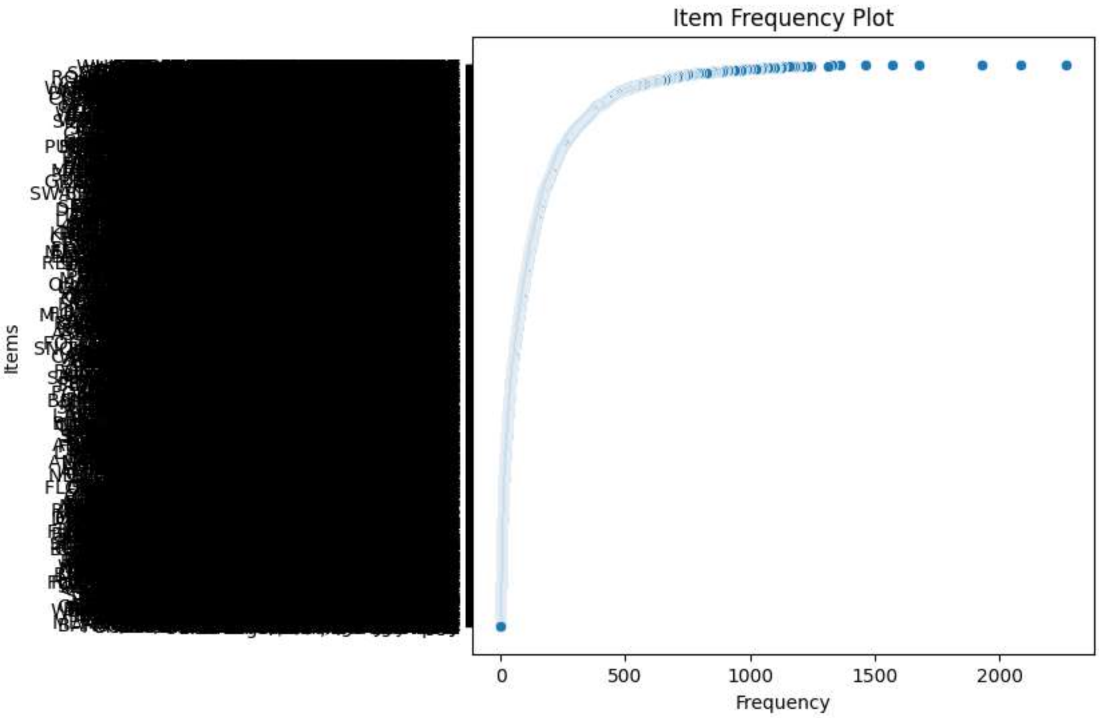
Output



#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

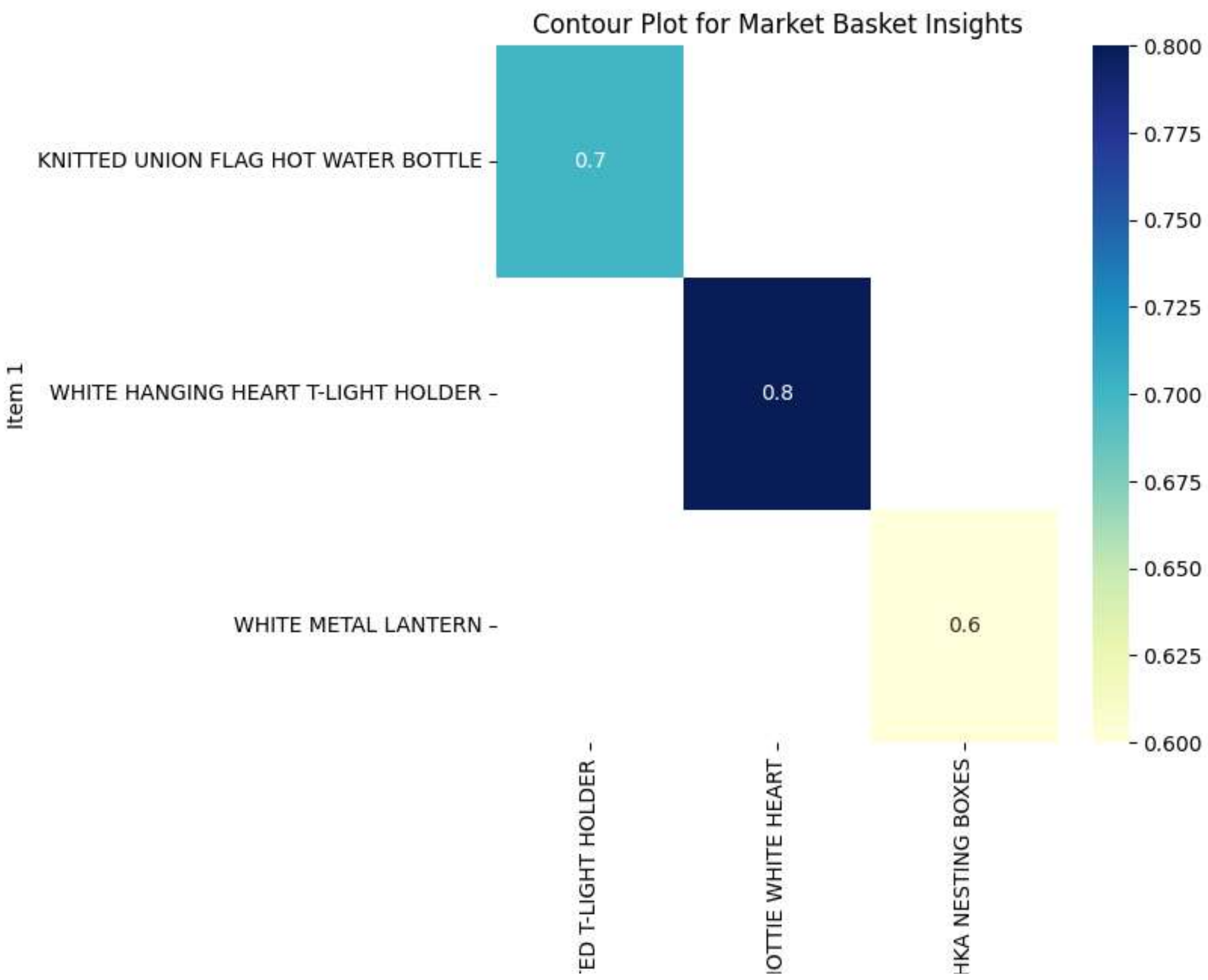


#Visualising through contour plots

```
# Example data - You should replace this with your actual data and analysis results
item1 = ['WHITE HANGING HEART T-LIGHT HOLDER', 'WHITE METAL
LANTERN', 'KNITTED UNION FLAG HOT WATER BOTTLE']

item2 = ['RED WOOLLY HOTTIE WHITE HEART', 'SET 7 BABUSHKA NESTING
BOXES', 'GLASS STAR FROSTED T-LIGHT HOLDER']
confidence = [0.8, 0.6, 0.7]

# Create a DataFrame with the results
data = pd.DataFrame({'Item1': item1, 'Item2': item2, 'Confidence': confidence})

# 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()
```

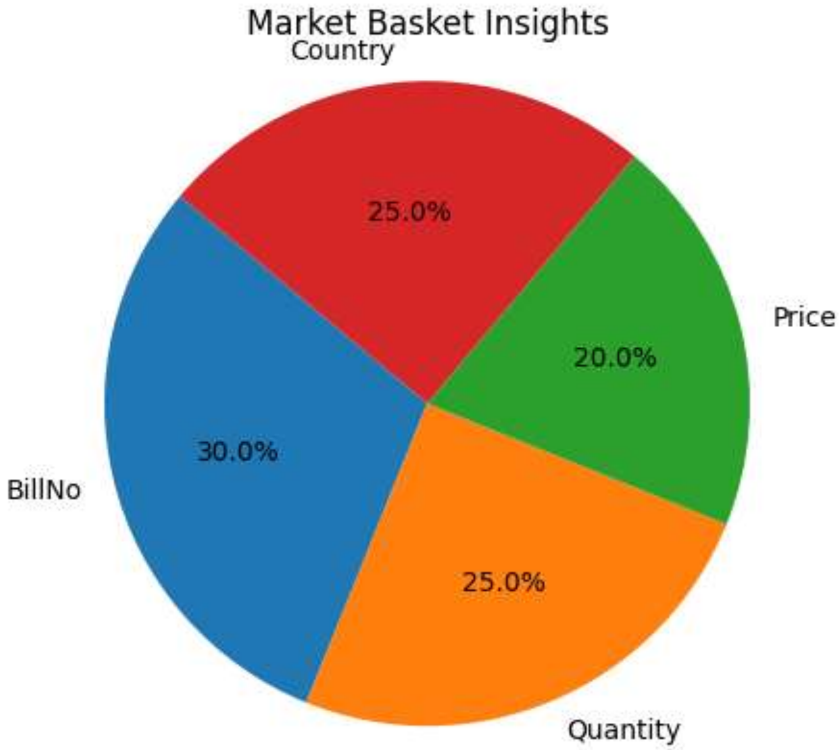
output



#Visualising through pie-chart

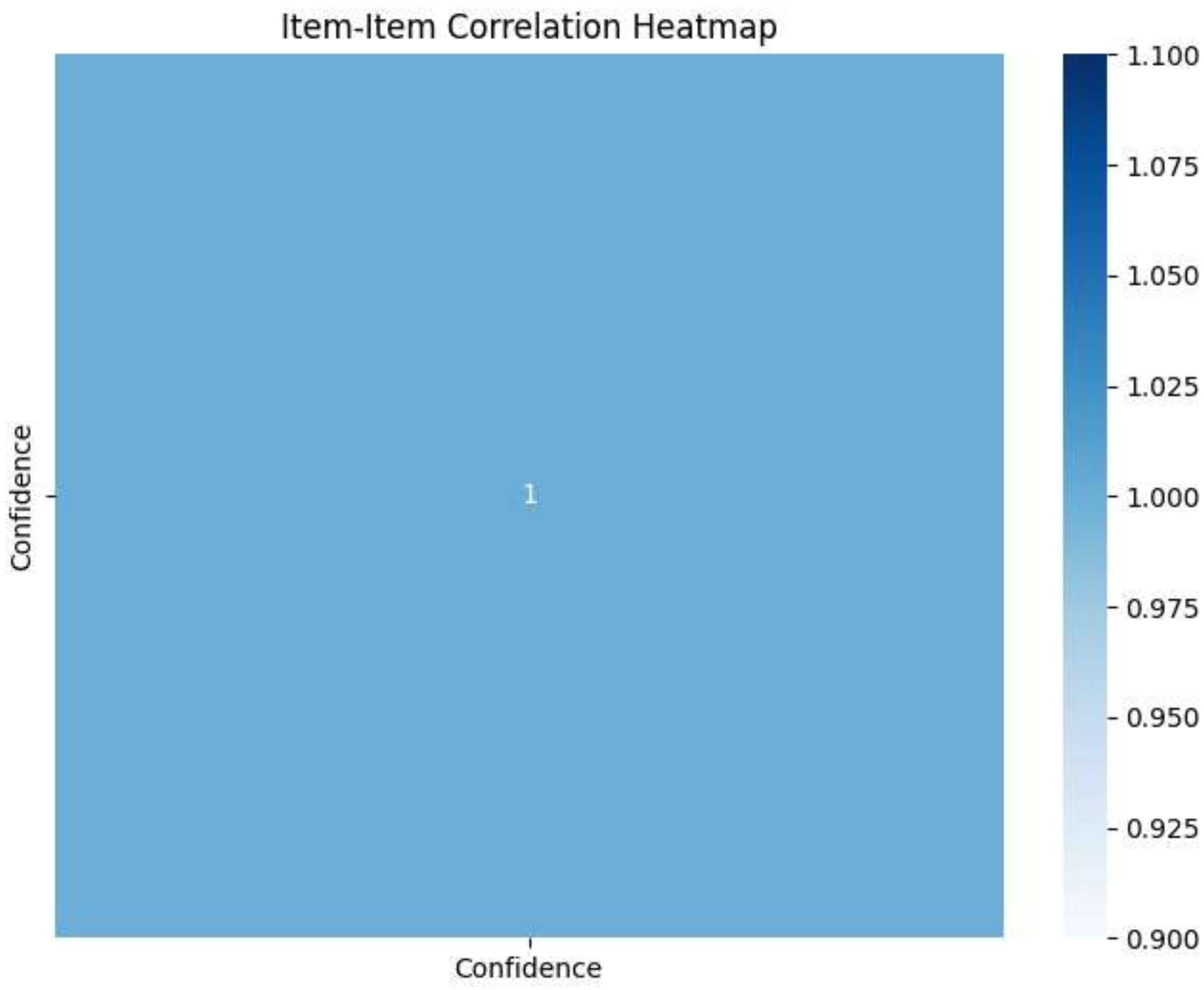
```
 # Sample data (replace with your market basket insights)
labels = ['BillNo', 'Quantity', 'Price', 'Country']
sizes = [30, 25, 20, 25]  # Replace with your actual data

# 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



```python
# 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

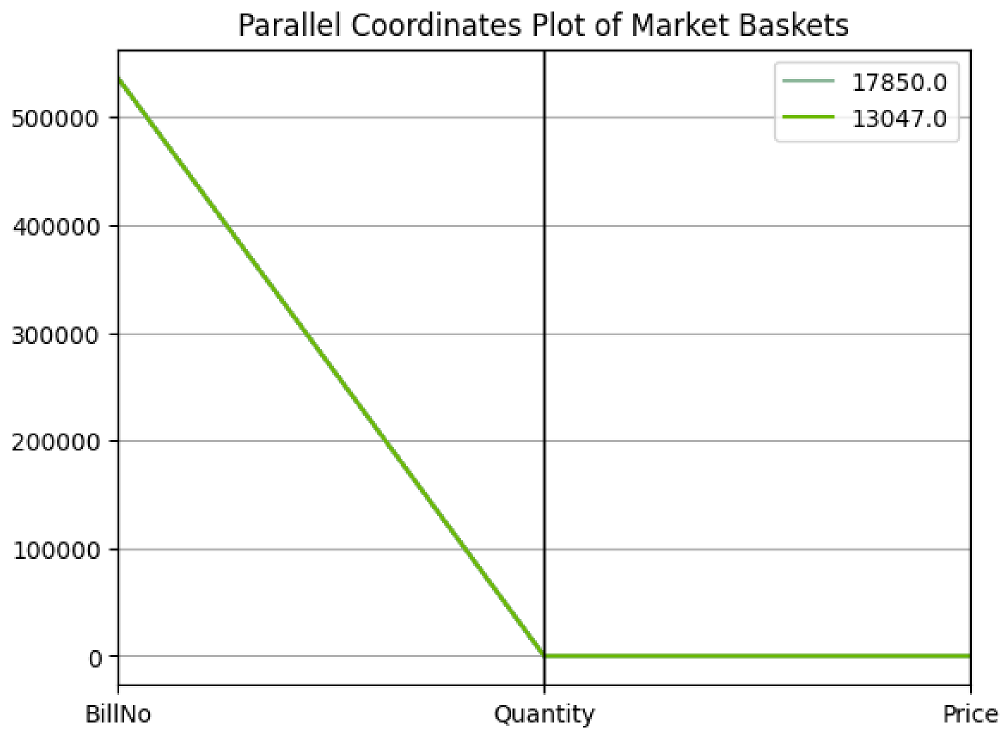


#Plotting with Parellel Coordinators

```python
from pandas.plotting import parallel_coordinates

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

```
plt.show()
```

output




```
from wordcloud import WordCloud

item_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()
```