MARKET BASKET INSIGHT

By:-

Aishwarya.M

3rd year, computer science and engineering
RVS COLLEGE OF ENGINEERING

ABSTRACT

Market basket insight refers to the valuable information obtained from conducting market basket analysis.

These insights can include information about the product association, cross-selling opportunities, and customer preferences.

Market basket insights are the actionable results and statistic guidance that arise from the analysis.

INTRODUCTION

ASSOCIATION RULES

Association rules are a powerful tool for discovering relationships in data sets.Association analysis involves exploring the dataset to identify meaningful patterns in item combinations based on statistical significance.Association rules play a vital role in Machine Learning by exploring intriguing relationships within dataset variables. Their significance extends across various domains, from data mining, where they uncover patterns, to continuous production, where they optimize processes.Association Rule Mining is sometimes referred to as “Market Basket Analysis”, as it was the first application area of association mining.

USES OF ASSOCIATION RULE

Association rules are widely used in various applications, including market basket analysis, recommendation systems (to suggest related products), fraud detection, and more, to reveal valuable insights and drive data-driven decision-making.

- **MARKET BASKET ANALYSIS**

Market basket analysis is one of the most popular examples and uses of association rule mining. Big retailers typically use this technique to determine the association between items

An association rule consists of three components:

- Antecedent (Left-hand side, LHS): This represents the items or products that are observed or considered as a premise.

- Consequent (Right-hand side, RHS): This represents the items or products that are observed or expected as a consequence.

- Support, Confidence, and Lift: These are statistical measures associated with the rule, quantifying the significance and strength of the association between the antecedent and consequent.

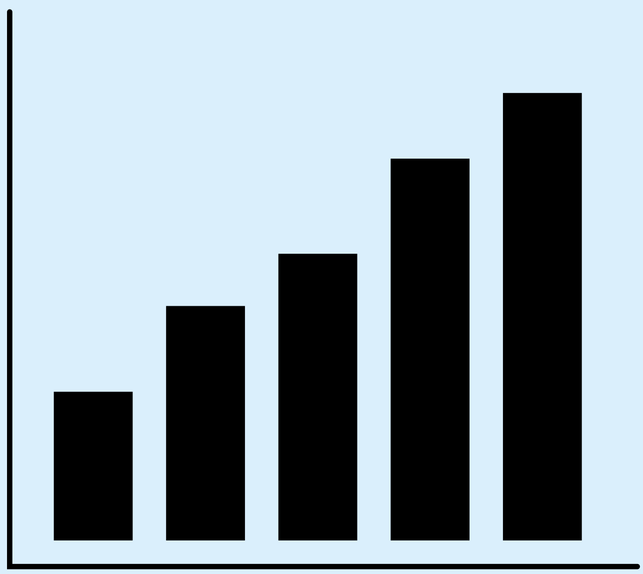
VISUALISATION

Association rule visualization is the graphical representation of association rules discovered through techniques like the Apriori algorithm or FP-growth in data mining and market basket analysis. The purpose of visualization is to make complex patterns and relationships among items or attributes more accessible and understandable for human interpretation.

The choice of visualization method depends on the nature of your data, the number of rules, and the specific insights you want to gain. Effective visualization can help data analysts and decision-makers quickly grasp important patterns and relationships within the association rules, making it a valuable tool in market basket analysis, recommendation systems, and various other applications.

TYPES OF VISUALISATION

- Scatter plot
- Graph
- Matrix visualisation



VISUALISATION TOOLS

Several visualization tools can be used for Market Basket Analysis and the visualization of association rules. These tools can help you better understand and communicate the relationships between products or items in your transactional data.

Popular visualisation tools

- Tableau
- PowerBI
- Excel

CODE

Importing python libraries

```
import numpy as np
import pandas as pd
import import pandas as pdos
```

Importing data set

```
dataset_path = '/kaggle/input/market-basket-
analysis/Assignment-1_Data.xlsx'
df = pd.read_excel(dataset_path)
```

Initial Exploration

```
print("Number of rows and columns:", df.shape)
print("\nData Types and Missing Values:")
print(df.info())
print("\nFirst few rows of the dataset:")
print(df.head())
```

Output

```
Number of rows and columns: (522064, 7)

Data Types and Missing Values:
<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  datetime64[ns]
 4   Price       522064 non-null  float64
 5   CustomerID  388023 non-null  float64
 6   Country     522064 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 27.9+ MB
None

First few rows of the dataset:
  BillNo                            Itemname  Quantity                Date  \
0  536365   WHITE HANGING HEART T-LIGHT HOLDER         6 2010-12-01 08:26:00
1  536365                  WHITE METAL LANTERN         6 2010-12-01 08:26:00
2  536365       CREAM CUPID HEARTS COAT HANGER         8 2010-12-01 08:26:00
3  536365  KNITTED UNION FLAG HOT WATER BOTTLE         6 2010-12-01 08:26:00
4  536365       RED WOOLLY HOTTIE WHITE HEART.         6 2010-12-01 08:26:00

Price  CustomerID         Country
0   2.55     17850.0  United Kingdom
1   3.39     17850.0  United Kingdom
2   2.75     17850.0  United Kingdom
3   3.39     17850.0  United Kingdom
4   3.39     17850.0  United Kingdom
```

Preprocessing

```
print("Missing Values:")
print(df.isnull().sum())
df.dropna(inplace=True)
transaction_data = df.groupby(['BillNo', 'Date'])
['Itemname'].apply(lambda x: ', '.join(x)).reset_index()
columns_to_drop = ['BillNo', 'Date']
transaction_data.drop(columns=columns_to_drop,
inplace=True)
transaction_data_path = '/kaggle/working/transaction_data.csv'
transaction_data.to_csv(transaction_data_path, index=False)
print("\nTransaction Data for Association Rule Mining:")
print(transaction_data.head())
transaction_data.shape
```

Output

```
Missing Values:
BillNo          0
Itemname     1455
Quantity        0
Date            0
Price           0
CustomerID 134041
Country         0
dtype: int64
```

Association rule mining

```
items_df =
transaction_data['Itemname'].str.split(', ',
expand=True)
 items DataFrame
transaction_data = pd.concat([transaction_data,
items_df], axis=1)
transaction_data =
transaction_data.drop('Itemname', axis=1)
print(transaction_data.head())

df_encoded =
pd.read_csv('transaction_data_encoded.csv')


from mlxtend.frequent_patterns import apriori,
association_rules
frequent_itemsets = apriori(df_encoded,
min_support=0.007, use_colnames=True)
rules = association_rules(frequent_itemsets,
metric="confidence", min_threshold=0.5)
print("Association Rules:")
print(rules.head())
```

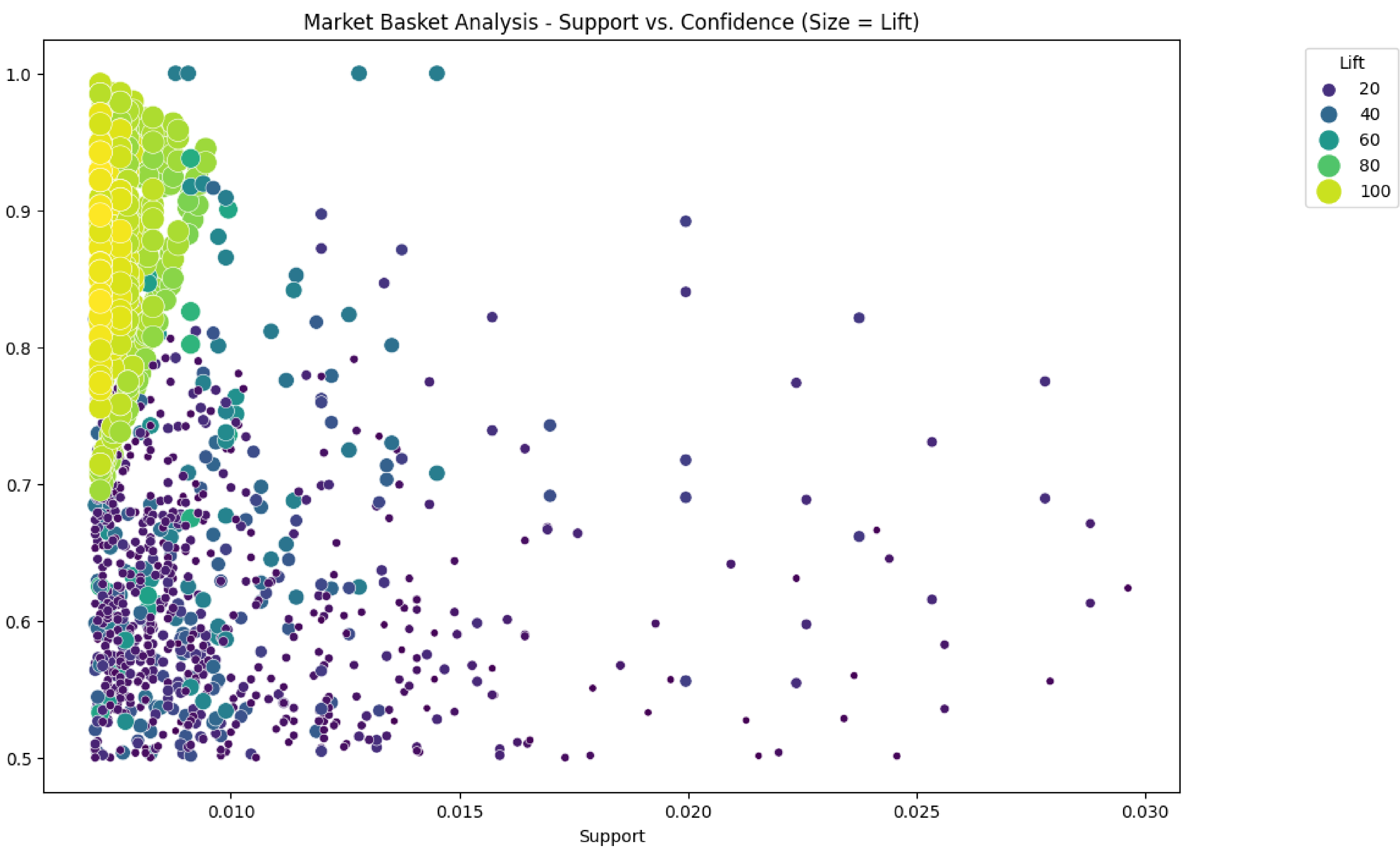
Output

Association Rules:

```
                    antecedents                    consequents  \
0          (CHOCOLATE BOX RIBBONS)         (6 RIBBONS RUSTIC CHARM)
1  (60 CAKE CASES DOLLY GIRL DESIGN)  (PACK OF 72 RETROSPOT CAKE CASES)
2     (60 TEATIME FAIRY CAKE CASES)  (PACK OF 72 RETROSPOT CAKE CASES)
3   (ALARM CLOCK BAKELIKE CHOCOLATE)      (ALARM CLOCK BAKELIKE GREEN)
4   (ALARM CLOCK BAKELIKE CHOCOLATE)       (ALARM CLOCK BAKELIKE PINK)

   antecedent support  consequent support   support  confidence      lift  \
0           0.012368           0.039193  0.007036    0.568889  14.515044
1           0.018525           0.054529  0.010059    0.543027   9.958409
2           0.034631           0.054529  0.017315    0.500000   9.169355
3           0.017150           0.042931  0.011379    0.663462  15.454151
4           0.017150           0.032652  0.009125    0.532051  16.294742

0  0.006551   2.228676      0.942766
1  0.009049   2.068984      0.916561
2  0.015427   1.890941      0.922902
3  0.010642   2.843862      0.951613
4  0.008565   2.067210      0.955009
```

VISUALISATION

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12, 8))
sns.scatterplot(x="support", y="confidence", size="lift", data=rules,
hue="lift", palette="viridis", sizes=(20, 200))
plt.title('Market Basket Analysis - Support vs. Confidence (Size =
Lift)')
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.legend(title='Lift', loc='upper right', bbox_to_anchor=(1.2, 1))
plt.show()

import plotly.express as px
rules['antecedents'] = rules['antecedents'].apply(list)
rules['consequents'] = rules['consequents'].apply(list)
fig = px.scatter(rules, x="support", y="confidence", size="lift",
            color="lift", hover_name="consequents",
            title='Market Basket Analysis - Support vs. Confidence',
            labels={'support': 'Support', 'confidence': 'Confidence'})
fig.update_layout(
   xaxis_title='Support',
   yaxis_title='Confidence',
   coloraxis_colorbar_title='Lift',
   showlegend=True
)
fig.show()
```


Market Basket Analysis - Support vs. Confidence (Size = Lift)
Lift
20
40
60
80
100
1.0
0.9
0.8
0.7
0.6
0.5
0.010
0.015
0.020
0.025
0.030
Support

Thank You!