**PHASE – IV**

**DEVELOPMENT PART 2**

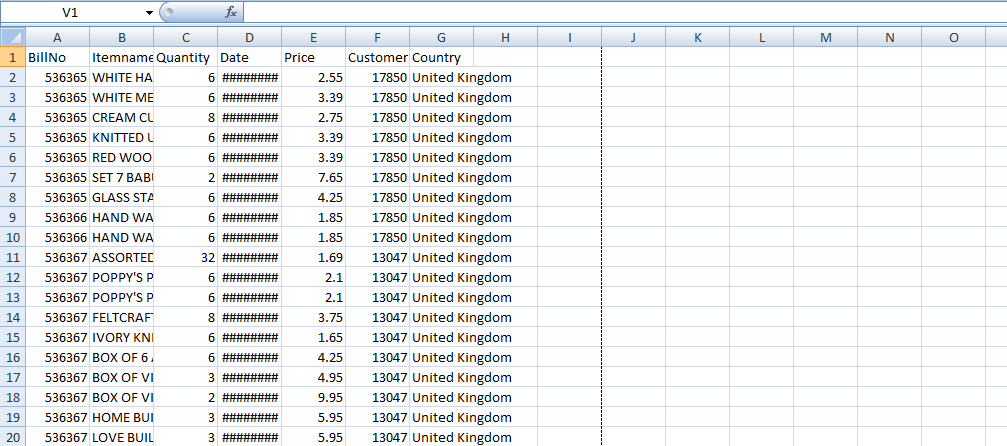
|  |  |
| --- | --- |
| DATE | 26-10-2023 |
| TEAM ID / TEAM NAME | Proj\_224020\_Team\_1 |
| PROJECT NAME | Market Basket Insights |
| STUDENT NAME WITH ID | **R Gunaseelan**  K.S Illayabharathi  V Dinesh  K Gowtham |

**Market Basket Insights**

**Introduction:**

In the ever-evolving landscape of retail, understanding customer behavior and optimizing business operations has become paramount. To thrive in this highly competitive industry, businesses must harness the power of data-driven insights to uncover hidden patterns and associations that drive consumer choices. Market Basket Analysis, a powerful data mining technique, offers a solution to precisely that challenge.This document embarks on a journey into the world of Market Basket Analysis, where we explore the relationships between products, unveil the mysteries of customer purchasing behavior, and identify potential cross-selling opportunities.

**DATASET:**

****

**Visualization**

**Converting Items to Boolean Columns**

To prepare the data for association rule mining, we convert the items in the transaction\_data DataFrame into boolean columns using one-hot encoding. This is achieved through the pd.get\_dummies function, which creates a new DataFrame (df\_encoded) with boolean columns representing the presence or absence of each item.

*# Convert items to boolean columns*

df\_encoded = pd.get\_dummies(transaction\_data, prefix='', prefix\_sep='').groupby(level=0, axis=1).max()

*# Save the transaction data to a CSV file*

df\_encoded.to\_csv('transaction\_data\_encoded.csv', index=False)

**Association Rule Mining**

We apply the Apriori algorithm to perform association rule mining on the encoded transaction data. The min\_support parameter is set to 0.007 to filter out infrequent itemsets. The resulting frequent itemsets are then used to generate association rules based on a minimum confidence threshold of 0.5.Finally, we print the generated association rules.

*# Load transaction data into a DataFrame*

df\_encoded = pd.read\_csv('transaction\_data\_encoded.csv')

from mlxtend.frequent\_patterns import apriori, association\_rules

*# Association Rule Mining*

frequent\_itemsets = apriori(df\_encoded, min\_support=0.007, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="confidence", min\_threshold=0.5)

## Visualizing Market Basket Analysis Results

We use matplotlib and seaborn libraries to create a scatterplot visualizing the results of the market basket analysis. The plot depicts the relationship between support, confidence, and lift for the generated association rules.

import matplotlib.pyplot as plt

import seaborn as sns

# Plot scatterplot for Support vs. Confidence

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)')

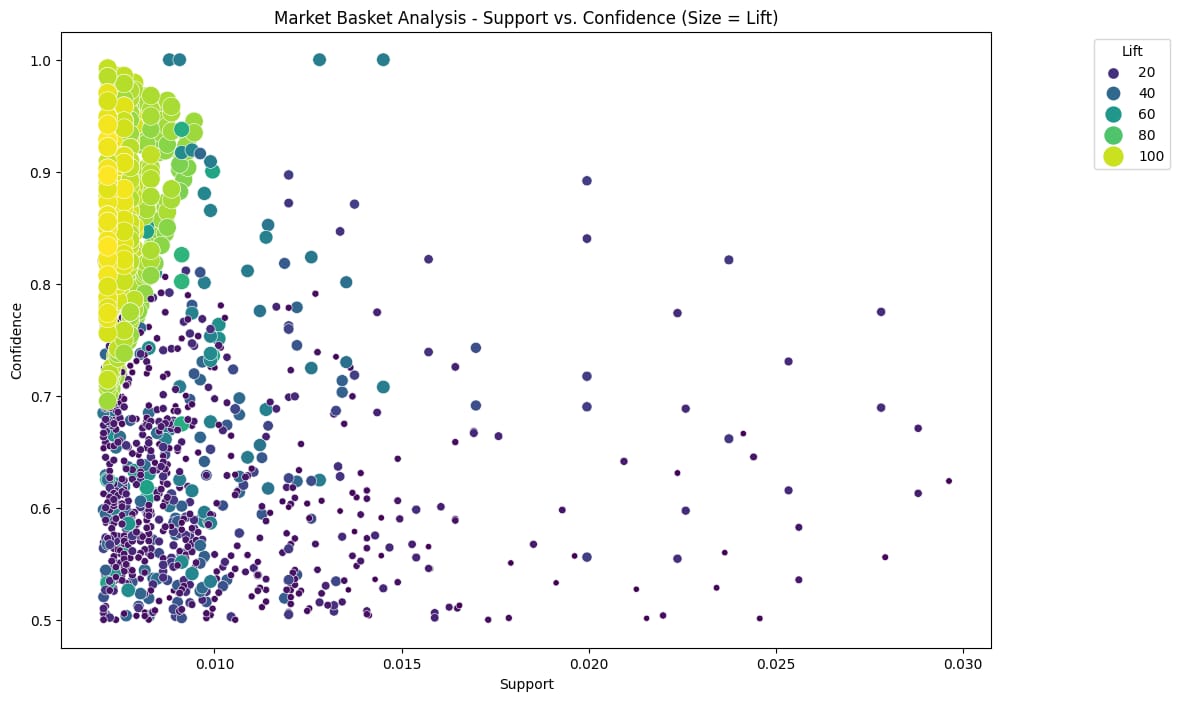
plt.xlabel('Support')

plt.ylabel('Confidence')

plt.legend(title='Lift', loc='upper right', bbox\_to\_anchor=(1.2, 1))

plt.show()

**OUTPUT:**

****

**Interactive Market Basket Analysis Visualization**

We leverage the Plotly Express library to create an interactive scatter plot visualizing the results of the market basket analysis. This plot provides an interactive exploration of the relationship between support, confidence, and lift for the generated association rules.

import plotly.express as px

# Convert frozensets to lists for serialization

rules['antecedents'] = rules['antecedents'].apply(list)

rules['consequents'] = rules['consequents'].apply(list)

# Create an interactive scatter plot using plotly express

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'})

# Customize the layout

fig.update\_layout(

xaxis\_title='Support',

yaxis\_title='Confidence',

coloraxis\_colorbar\_title='Lift',

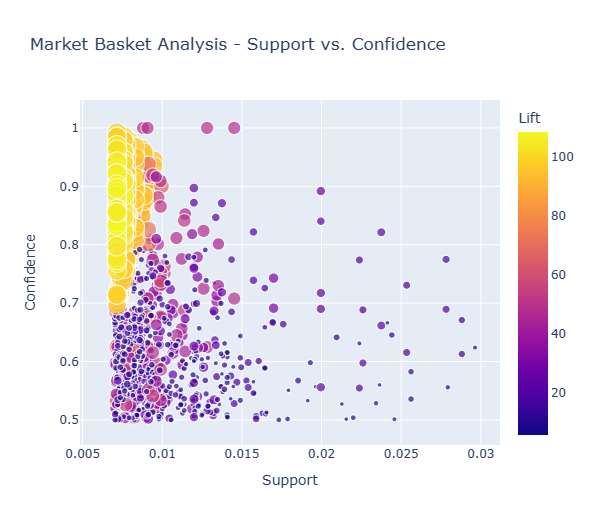
showlegend=True

)

# Show the interactive plot

fig.show()

**OUTPUT:**

****

**Conclusion:**

-Description: In the concluding stage, we summarize key findings and discuss their implications for the retail business.

-Visual Representation: A rectangular shape labeled "Conclusion" connected to the outputs of the various actions.

**End:**

-Description: The endpoint of the flowchart, representing the conclusion of the project.

-Visual Representation: A rounded rectangle labeled "End" connected to the "Conclusion" stage.