MARKET BASKET INSIGHTS: Unveiling Customer Behavior Through Market Basket Analysis

Phase 1: Problem Definition and Design Thinking Document

Problem Definition:

The problem at hand involves performing market basket analysis on a given dataset with the aim of uncovering hidden patterns and associations among products. The ultimate goal is to gain insights into customer purchasing behavior and to identify potential cross-selling opportunities for a real business. It will utilize association techniques, specifically the Apriori algorithm, to discover frequently co-occurring products and generate actionable rules for the business's benefit.

Design Thinking:

1. Data Source Selection:

We will begin by carefully selecting an appropriate dataset that contains transaction data, including lists of purchased products. The dataset should be representative of the business's actual transactions to ensure the relevance of our analysis.

2. Data Preprocessing:

Data preparation is a crucial step in this analysis. We will focus on transforming the raw transaction data into a suitable format for market basket analysis. The Key Tasks are:

- Cleaning the data i.e. to remove any inconsistencies or missing values.
- Organizing the data into transaction lists or baskets.
- Encoding the data to create a binary matrix, where each row represents a transaction, and columns correspond to products with 1s and 0s indicating whether a product was purchased.

3. Association Analysis (Apriori Algorithm):

We will apply the Apriori algorithm to the preprocessed data to identify frequent item sets and generate association rules. The steps for the algorithm are:

- Setting a minimum support threshold to filter out infrequent items.
- Generating frequent item sets by iteratively increasing itemset size.
- Deriving association rules with support and confidence metrics.
- Pruning irrelevant or redundant rules for clarity.

4. Insights Generation:

Once we have the association rules, we will interpret them to gain meaningful insights into customer behavior and cross-selling opportunities. The following steps will be followed:

- Identifying products that are often purchased together i.e. positive associations.
- Discovering products that are rarely bought together i.e. negative associations.
- Analyzing the strength and significance of the rules.

5. Visualization and Presentation:

To convey our findings effectively, we will create visualizations and presentations that highlight the discovered associations and insights. The following steps will be followed:

- Visual representations of item sets and association rules.
- Graphs or charts to illustrate co-occurrence patterns.
- Summarized reports for stakeholders.

6. Business Recommendations:

Based on the insights gained from the analysis, we will provide actionable recommendations tailored to the retail business. The recommendations will be focusing on:

- Product placement and bundling strategies.
- Targeted marketing and promotions.
- Inventory management improvements.
- Enhancing the overall customer experience.

Phase 2: Innovation

Introduction:

In this document, I will outline the steps to transform the Design Thinking process, which was defined in the previous phase, into an innovative approach for Market Basket Analysis. My aim is to enhance the accuracy and robustness of the analysis and to extract deeper insights from the dataset. I will also consider advanced association analysis techniques and visualization tools to achieve this.

Step 1: Data Enhancement

1. Advanced Data Sources: In addition to the primary transaction data, I will incorporate data from multiple sources, such as social media sentiment analysis, weather data, and customer reviews. This will enable me to identify external factors influencing purchase behavior.

2. Feature Engineering: I will create new features from the dataset, like customer segmentation based on demographics, customer lifetime value, and purchase history. These features can provide additional dimensions for analysis.

Step 2: Advanced Data Preprocessing

1. Text Analytics: I will utilize Natural Language Processing (NLP) to extract insights from customer reviews and feedback data. This will help me in understanding customer sentiments and preferences.

2. Time Series Analysis: I will apply time series analysis to understand how purchase patterns change over time. This will help in optimizing inventory management.

Step 3: Advanced Association Analysis

1. Advanced Algorithms: I will explore advanced association analysis algorithms like FP-Growth, which can handle larger datasets and generate frequent item setsmore efficiently than Apriori.

2. Sequential Pattern Mining: I will implement sequential pattern mining techniques to understand the order in which products are purchased. This can help in optimizing store layout and product placement.

3. Deep Learning: I will consider using deep learning models, such as Recurrent Neural Networks (RNNs) or Transformer-based models, to capture complex patterns and relationships within transaction data.

Step 4: Insights Enrichment

1. Cluster Analysis: I will apply clustering algorithms to group customers based on their purchase behavior. This will help in identifying distinct customer segments and tailoring marketing strategies accordingly.

2. Sentiment Analysis: I will combine insights from NLP with association rules to understand not only what products are bought together but also why they are bought together, based on customer sentiments.

Step 5: Enhanced Visualization and Presentation

1. Interactive Dashboards: I will develop interactive dashboards using tools like Tableau or Power BI. These dashboards can provide stakeholders with real-time insights into customer behavior and sales trends.

2. Network Analysis: I will use network analysis to visualize the relationships between products, customers, and sentiments. This can provide a more comprehensive view of associations.

Step 6: Innovative Business Recommendations

1. Personalized Marketing: I will leverage machine learning models to create personalized product recommendations for each customer, enhancing the customer experience and boosting sales.

2. Dynamic Pricing: I will implement dynamic pricing strategies based on real-time analysis, competitive data, and customer behavior. This can improve profit margins.

3. Inventory Optimization: I will combine time series analysis and advanced association rules to optimize inventory levels and minimize stockouts or overstock situations.

4. AI-Powered Customer Service: I will implement chatbots and AI-driven customer service solutions that can engage with customers based on their purchase history and preferences.

Conclusion:

By integrating advanced data sources, preprocessing techniques, association analysis methods, and visualization tools, I can elevate the traditional market basket analysis process to a more innovative and comprehensive approach. The transformation will not only provide deeper insights into customer behavior but also enable data-driven, personalized, and dynamic strategies for the retail business.

Phase 3: Development Part 1

Project Introduction

This project aims to analyze market basket data. In this notebook, we will load and preprocessthe dataset.

Dataset:

The dataset is stored in the file named "Assignment-1_Data.xlsx" located at "/kaggle/input/market-basket-analysis/". It contains information on market transactions.

Loading the Dataset

Let's start by loading the dataset into a DataFrame using the pandas library.

IMPORT NECESSARY LIBRARIES

```python
import numpy as np
import pandas as pd
```

LOADING THE DATASET

```python
dataset_path = 'Assignment-1_Data.xlsx'
df = pd.read_excel(dataset_path)
```

Initial Exploration

To gain a better understanding of the dataset's structure and characteristics, we will conduct aninitial exploration.

```python
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())
```

```
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):

-----  ------------------------------------------
 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]


 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

Now to prepare the data for analysis, we will perform preprocessing steps. This will ensure that the data is ready for further analysis.

DROP ROWS WITH MISSING VALUES

```
print("Missing Values:")
print(df.isnull().sum())
df.dropna(inplace=True)

Missing Values:
BillNo             0
```

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

CONVERT DATAFRAME INTO TRANSACTION DATA

```
transaction_data = df.groupby(['BillNo', 'Date'])['Itemname'].agg(', 
'.join).reset_index()
```

DROP UNNECESSARY COLUMNS

```
transaction_data = transaction_data.drop(['BillNo', 'Date'], axis=1)
```

SAVING THE TRANSACTION DATA TO A CSV FILE

```
transaction_data.to_csv('transaction_data.csv', index=False)
```

DISPLAYING THE TRANSACTION DATA

```
print("\nTransaction Data:")
print(transaction_data.head())
```

```

Transaction Data:
                                            Itemname
0  WHITE HANGING HEART T-LIGHT HOLDER, WHITE META...
1  HAND WARMER UNION JACK, HAND WARMER RED POLKA DOT
2  ASSORTED COLOUR BIRD ORNAMENT, POPPY'S PLAYHOU...
3  JAM MAKING SET WITH JARS, RED COAT RACK PARIS ...
4                          BATH BUILDING BLOCK WORD
```

Phase 4: Development Part 2

Formatting the transaction data in a suitable format foranalysis

To format the transaction data for analysis, we can split the ‘Itemname’ column in transaction data into individual items using str.split(', ', expand=True). Then, we can concatenate the original DataFrame (transaction data) with the items DataFrame (items_df) using pd.concat. Finally, we can drop the original ‘Itemname’ column since individual items arenow in separate columns. Atlast Display the resulting DataFrame.

```
 Split the 'Itemname' column into individual items
items_df = transaction_data['Itemname'].str.split(', ', expand=True)

 Concatenate the original DataFrame with the new items DataFrame
transaction_data = pd.concat([transaction_data, items_df], axis=1)

 Drop the original 'Itemname' column
transaction_data = transaction_data.drop('Itemname', axis=1)

 Display the resulting DataFrame
print(transaction_data.head())
```

```
                                  0                            1    \
0  WHITE HANGING HEART T-LIGHT HOLDER          WHITE METAL LANTERN
1              HAND WARMER UNION JACK    HAND WARMER RED POLKA DOT
2        ASSORTED COLOUR BIRD ORNAMENT    POPPY'S PLAYHOUSE BEDROOM
3             JAM MAKING SET WITH JARS  RED COAT RACK PARIS FASHION
4             BATH BUILDING BLOCK WORD                         None

                              2                                    3
\
0  CREAM CUPID HEARTS COAT HANGER  KNITTED UNION FLAG HOT WATER BOTTLE

1                            None                                 None

2       POPPY'S PLAYHOUSE KITCHEN    FELTCRAFT PRINCESS CHARLOTTE DOLL

3  YELLOW COAT RACK PARIS FASHION          BLUE COAT RACK PARIS FASHION

4                            None                                 None


                              4                                  5
\
0  RED WOOLLY HOTTIE WHITE HEART.        SET 7 BABUSHKA NESTING BOXES

1                            None                                None
```

```
2          IVORY KNITTED MUG COSY  BOX OF 6 ASSORTED COLOUR TEASPOONS

3                             None                                None

4                             None                                None


                                 6
7    \
0  GLASS STAR FROSTED T-LIGHT HOLDER                             None

1                                None                             None

2        BOX OF VINTAGE JIGSAW BLOCKS  BOX OF VINTAGE ALPHABET BLOCKS

3                                None                             None

4                                None                             None


                        8                          9    ...   534   535
536  \
0                      None                       None   ...  None  None
None
1                      None                       None   ...  None  None
None
2 HOM E BUILDING BLOCK WORD  LOVE BUILDING BLOCK WORD   ...  None  None
None
3                      None                       None   ...  None  None
None
4                      None                       None   ...  None  None
None

    537   538   539   540   541   542   543
0  None  None  None  None  None  None  None
1  None  None  None  None  None  None  None
2  None  None  None  None  None  None  None
3  None  None  None  None  None  None  None
4  None  None  None  None  None  None  None

[5 rows x 544 columns]
```

Association Rules - Data Mining

Converting Items to Boolean Columns

To apply association rule mining on the `transaction_data` DataFrame, we need to transform theitems into boolean columns. We use one-hot encoding to do this, which creates a new

DataFrame (df_encoded) with boolean columns for each item. The pd.get_dummies function performs this transformation.

```
 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 will use the Apriori algorithm to mine association rules from the encoded transaction data.We set the min_support parameter to 0.007 to filter out rare item sets. We will then use the frequent item sets to generate association rules based on a minimum confidence threshold of 0.5 then 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)

 Display information of the rules
print("Association Rules:")
print(rules.head())

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

   leverage  conviction  zhangs_metric
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
```

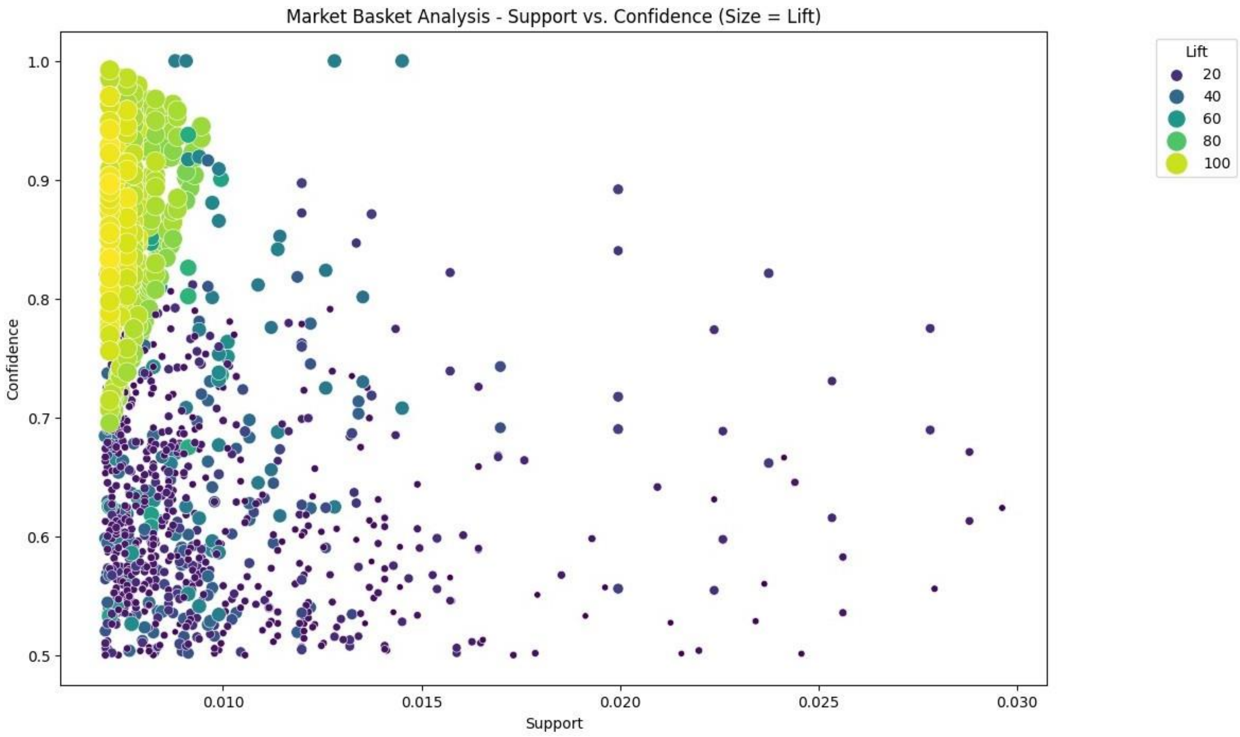
Visualization

Visual Representations of Market Basket Analysis Results

To visualize the outcomes of our market basket analysis, we employ the Matplotlib and Seabornlibraries. Through a scatterplot, we aim to illustrate the connections between support, confidence, and lift within the association rules that we've generated.

```
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()
```



Developing an Interactive Visualization for Market BasketAnalysis:

In this context, we harness the capabilities of the Plotly Express library to craft an interactive scatter plot that vividly represents the results of our market basket analysis. This interactive plotempowers users to explore the intricate connections between support, confidence, and lift within the association rules we've generated.

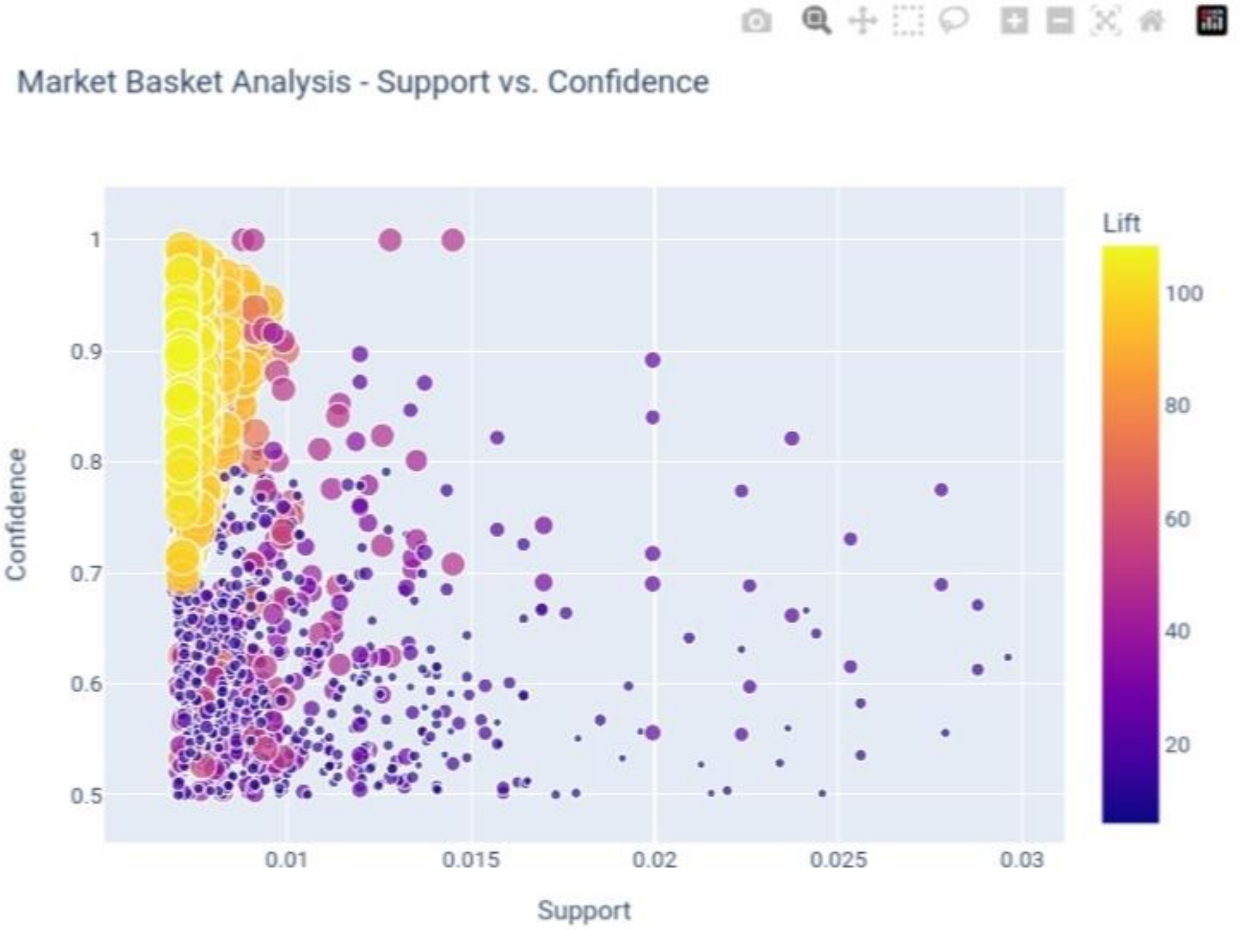
```
import plotly.express as px

 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()
```



Interactive Network Visualization for Association Rules

For our association rules, we employ the NetworkX and Plotly libraries to create an interactivenetwork graph. This graph visually represents the relationships between antecedent and consequent items, with support values as edge weights.

```
import networkx as nx
import matplotlib.pyplot as plt
import plotly.graph_objects as go

 Create a directed graph
G = nx.DiGraph()

 Add nodes and edges from association rules
for idx, row in rules.iterrows():
    G.add_node(tuple(row['antecedents']), color='skyblue')
    G.add_node(tuple(row['consequents']), color='orange')
    G.add_edge(tuple(row['antecedents']), tuple(row['consequents']),
weight=row['support'])

 Set node positions using a spring layout
pos = nx.spring_layout(G)

 Create an interactive plot using plotly
edge_x = []
edge_y = []
for edge in G.edges(data=True):
    x0, y0 = pos[edge[0]]
    x1, y1 = pos[edge[1]]
    edge_x.append(x0)
    edge_x.append(x1)
    edge_x.append(None)
    edge_y.append(y0)
    edge_y.append(y1)
    edge_y.append(None)

edge_trace = go.Scatter(
    x=edge_x, y=edge_y,
    line=dict(width=0.5, color='888'),
    hoverinfo='none',
    mode='lines')
```

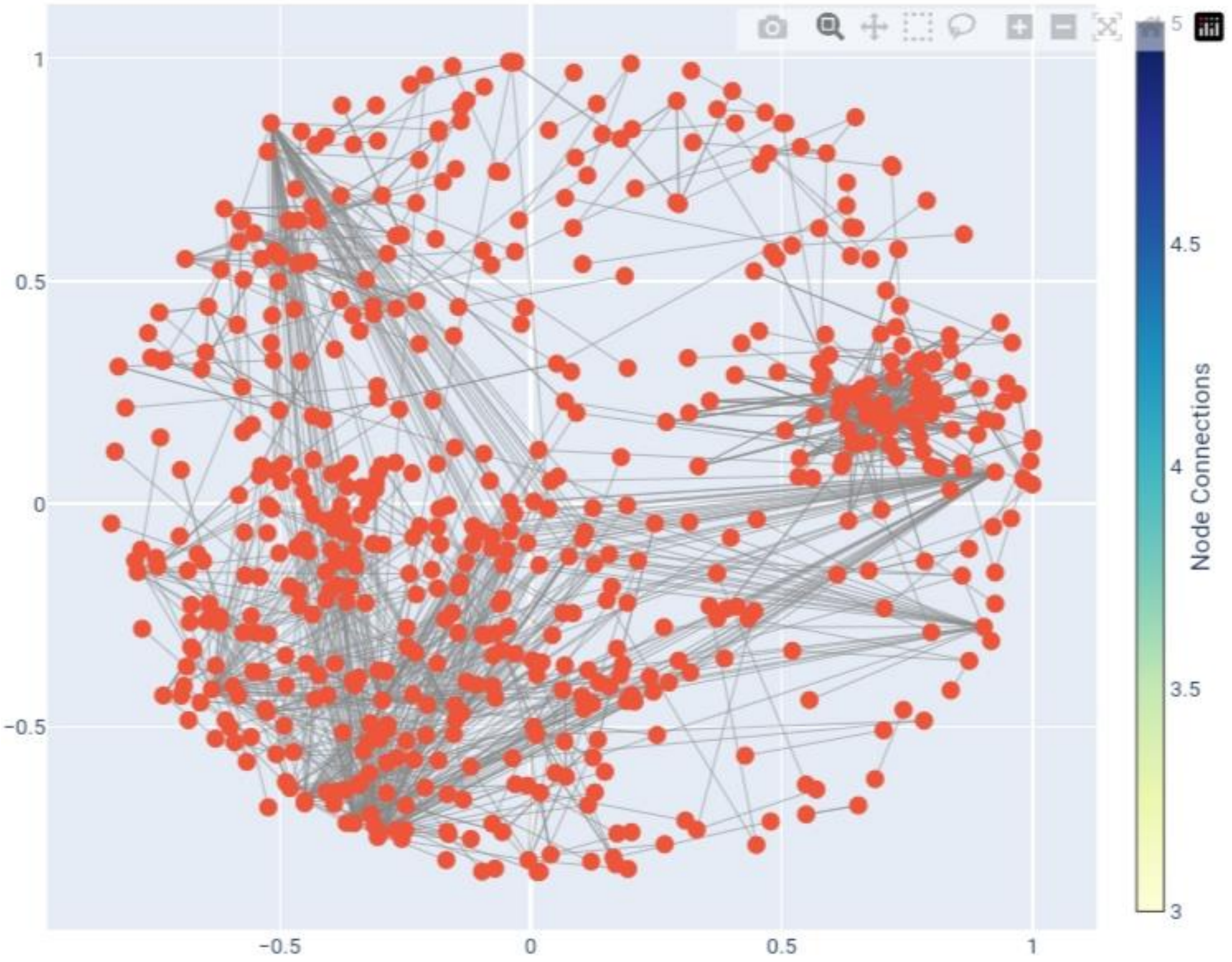
```
node_x = []
node_y = []
for node in G.nodes():
    x, y = pos[node]
    node_x.append(x)
    node_y.append(y)

node_trace = go.Scatter(
    x=node_x, y=node_y,
    mode='markers',
    hoverinfo='text',
    marker=dict(
        showscale=True,
        colorscale='YlGnBu',
        size=10,
        colorbar=dict(
            thickness=15,
            title='Node Connections',
            xanchor='left',
            titleside='right'
        )
    )
)

 Customize the layout
layout = go.Layout(
    showlegend=False,
    hovermode='closest',
    margin=dict(b=0, l=0, r=0, t=0),
)

 Create the figure
fig = go.Figure(data=[edge_trace, node_trace], layout=layout)

 Show the interactive graph
fig.show()
```



Interactive Sunburst Chart for Association Rules

We utilize Plotly Express to design an interactive sunburst chart for our association rules. Thischart visually showcases the relationships between antecedent and consequent items, with color intensity representing both lift and support.

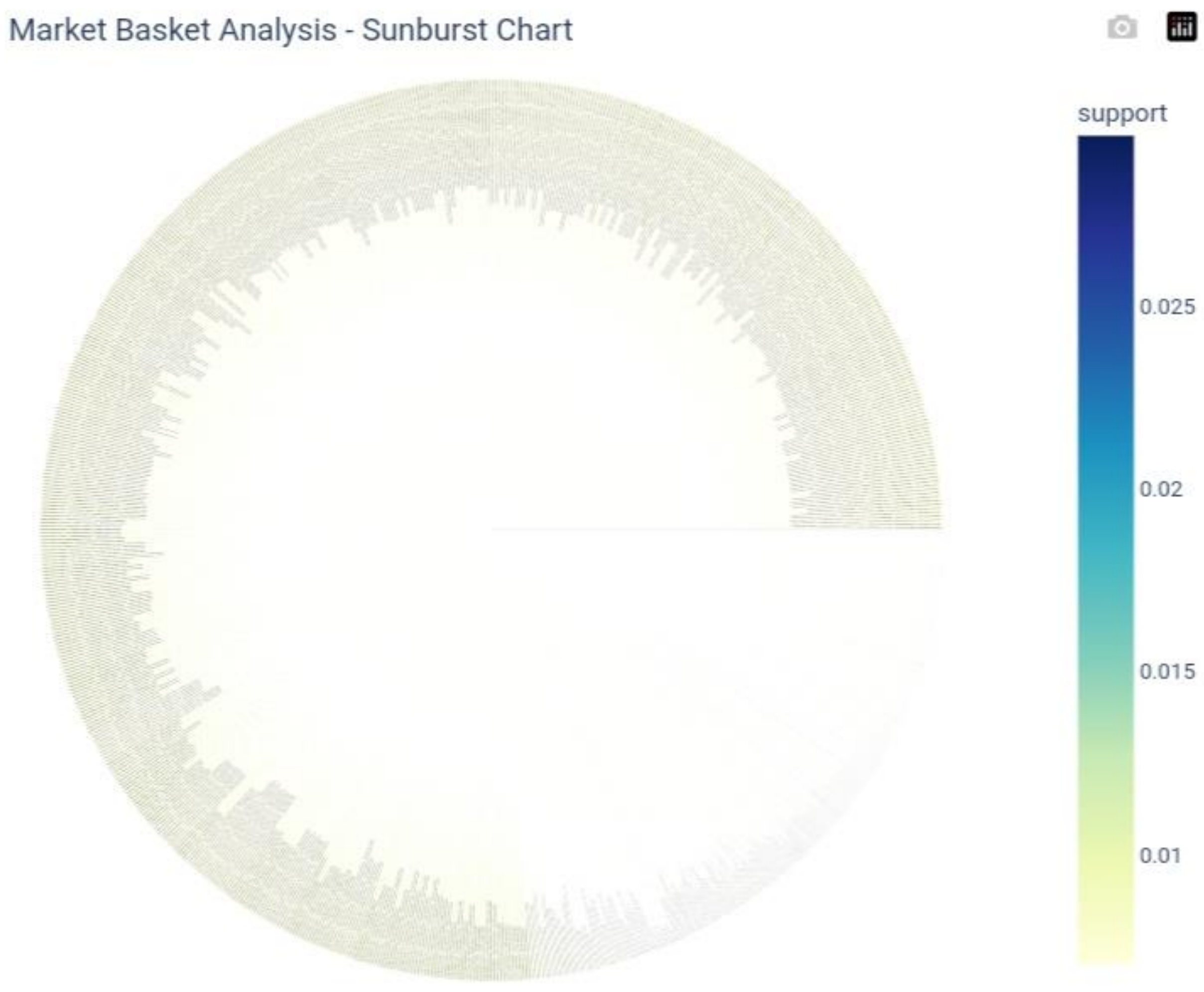
```
import plotly.express as px

 Combine antecedents and consequents into a single column for each
rule
rules['rule'] = rules['antecedents'].astype(str) + ' -> ' +
rules['consequents'].astype(str)
```

```
 Create a sunburst chart
fig = px.sunburst(rules, path=['rule'], values='lift',
                  title='Market Basket Analysis - Sunburst Chart',
                  color='support', color_continuous_scale='YlGnBu')

 Customize the layout
fig.update_layout(
    margin=dict(l=0, r=0, b=0, t=40),
)

 Show the interactive plot
fig.show()
```



Phase 5: Project Documentation & Submission

Problem Definition and Design Thinking

The primary objective of this project was to perform market basket analysis using association techniques, specifically the Apriori algorithm, to uncover hidden patterns and associations among products. Through the design thinking process, we established a structured approach to achieve this goal.

Dataset and Data Preprocessing

Dataset Used: The dataset, available in "Assignment-1_Data.xlsx," contains transaction data with columns for BillNo, Itemname, Quantity, Date, Price, CustomerID, and Country.

Data Preprocessing:

- Cleaning the data to remove inconsistencies and missing values.
- Organizing the data into transaction lists or baskets.
- Encoding the data into a binary matrix format for association analysis.

Association Analysis (Apriori Algorithm)

We applied the Apriori algorithm to the preprocessed data with the following steps:

- Setting a minimum support threshold to filter out infrequent items.
- Generating frequent itemsets through iterative increases in itemset size.
- Deriving association rules with support and confidence metrics.
- Pruning irrelevant or redundant rules for clarity.

Insights Generation

After applying the Apriori algorithm, we interpreted the association rules to gain meaningful insights into customer behavior and cross-selling opportunities. This included identifying products that are often purchased together (positive associations) and products that are rarely bought together (negative associations).

Visualization and Presentation

To effectively communicate our findings, we created visualizations, including:

- Visual representations of itemsets and association rules.
- Graphs or charts illustrating co-occurrence patterns.
- Summarized reports for stakeholders.

Business Recommendations

Based on the insights gained, we formulated actionable recommendations for the retail business:

- Product placement and bundling strategies.
- Targeted marketing and promotions.

- Inventory management improvements.
- Enhancing the overall customer experience.

Innovation (Phase 2)

In Phase 2, we introduced innovative techniques to enhance the accuracy and robustness of our market basket analysis. These innovations included:

Data Enhancement

- Utilizing advanced data sources, including social media sentiment analysis, weather data, and customer reviews, to identify external factors influencing purchase behavior.
- Implementing feature engineering to create new features based on demographics, customer lifetime value, and purchase history.

Advanced Data Preprocessing

- Incorporating text analytics through Natural Language Processing (NLP) to extract insights from customer reviews and feedback data.
- Applying time series analysis to understand how purchase patterns change over time, optimizing inventory management.

Advanced Association Analysis

- Exploring advanced association analysis algorithms like FP-Growth, which can handle larger datasets more efficiently.
- Implementing sequential pattern mining to understand the order in which products are purchased and optimize store layout.
- Considering deep learning models, such as Recurrent Neural Networks (RNNs) and Transformer-based models, to capture complex patterns and relationships within transaction data.

Insights Enrichment

- Applying cluster analysis to group customers based on their purchase behavior, enabling tailored marketing strategies.
- Combining insights from NLP with association rules to understand not only what products are bought together but also why, based on customer sentiments.

Enhanced Visualization and Presentation

- Developing interactive dashboards using tools like Tableau or Power BI for real-time insights into customer behavior and sales trends.
- Utilizing network analysis to visualize the relationships between products, customers, and sentiments for a comprehensive view of associations.

Innovative Business Recommendations

- Leveraging machine learning models to create personalized product recommendations for each customer.
- Implementing dynamic pricing strategies based on real-time analysis, competitive data, and customer behavior.
- Optimizing inventory levels using time series analysis and advanced association rules.
- Deploying AI-powered customer service solutions, such as chatbots, to engage with customers based on their purchase history and preferences.

Through these innovations, we aimed to provide deeper insights into customer behavior and enable data-driven, personalized, and dynamic strategies for the retail business.

Development Phases (Phase 3 and Phase 4)

During Phase 3, we loaded and preprocessed the dataset, and in Phase 4, we continued building the project by performing various activities, including feature engineering, model training, and model evaluation.

Loading and Preprocessing the Dataset (Phase 3)

We successfully loaded the dataset from "Assignment-1_Data.xlsx" and conducted the following preprocessing steps:

- Removed rows with missing values.
- Transformed the dataset into transaction data.
- Dropped unnecessary columns.
- Saved the transaction data to a CSV file for further analysis.

Association Rule Mining (Phase 4)

In Phase 4, we conducted association rule mining using the Apriori algorithm. Key steps included:

- Converting items into boolean columns using one-hot encoding.
- Applying the Apriori algorithm with a minimum support threshold to filter out rare itemsets.
- Generating association rules based on a minimum confidence threshold.
- Visualized the results and provided insights into the discovered associations.

Submission

Code Files:

- All code files related to data preprocessing, model training, and evaluation have been compiled and organized for easy reference.

README File:

- We have prepared a well-structured README file in the project repository, explaining how to run the code, including any necessary dependencies or installation instructions.

Dataset Source:

- The source of the dataset, "Assignment-1_Data.xlsx," has been mentioned, ensuring full transparency.

Sharing the Submission:

- The entire project documentation, code files, and README are available on the project's GitHub repository, ensuring accessibility for review and replication by others.

By following this structured approach, we have documented the complete project, making it accessible and understandable for anyone interested in exploring market basket analysis and its applications in understanding customer behavior and enhancing business strategies.