**PHASE – V**

|  |  |
| --- | --- |
| DATE | 30-10-2023 |
| TEAM ID / TEAM NAME | Proj\_224020\_Team\_1 |
| PROJECT NAME | Market Basket Insights |
| STUDENT NAME WITH ID | **V Dinesh (au610521104020)**  K Gowtham (au610521104029)  R Gunaseelan (au610521104031)  K.S Illaya Bharathi(au610521104033) |

**Abstract**

Market basket analysis is a powerful data mining technique used to uncover hidden patterns and associations within transaction data, offering valuable insights into customer purchasing behavior and business optimization opportunities. This abstract provides an overview of market basket analysis, its significance, and key components.

Market basket analysis, often associated with the Apriori algorithm, is widely applied across industries, including retail, e-commerce, and hospitality. It helps businesses understand which products or services are frequently purchased together, enabling targeted marketing strategies, cross-selling opportunities, and inventory management improvements.

This technique involves data preprocessing, frequent itemset generation, and rule mining to reveal meaningful associations between products. The mined association rules, defined by metrics such as support, confidence, and lift, offer actionable insights for businesses.

In this abstract, we emphasize the importance of data quality, data preprocessing, and the choice of appropriate thresholds for support and confidence in the analysis. We also explore the challenges of market basket analysis, such as data scalability and privacy considerations.

By uncovering patterns in customer purchasing behavior, market basket analysis contributes to more informed decision-making, improved customer satisfaction, and enhanced operational efficiency. This abstract highlights the significance of market basket analysis as a valuable tool for businesses seeking to harness the potential of their transaction data to drive growth and profitability.

**TABLE OF CONTENT**

**1 INTRODUCTION**

* 1. Purpose of project
  2. Design Thinking Approach
  3. Implementation
  4. Actions

1. **REQUIREMENTS AND FLOW** 
   1. Softwares
   2. Flowchar
   3. Datasets

**3 DATALOADING**

**3.1** Dataset

**3.2** Preprocessing the Dataset

**4 VISUALIZATION**

**4.1** Data transaction

**4.2** Data mining

**4.3** Data visualization

**1. INTRODUCTION**

**1.1 PURPOSE OF THE PROJECT**

The purpose of a market basket analysis project is to uncover hidden patterns and associations in customer transaction data. By identifying which products are often bought together, this project aims to optimize business strategies, enhance customer experiences, increase revenue through cross-selling opportunities, and make data-driven decisions that lead to improved operational efficiency and customer satisfaction. Ultimately, the project serves to provide actionable insights for businesses to tailor their offerings and marketing efforts, resulting in a competitive advantage and greater profitability.

**1.2 DESIGN THINKING APPROACH**

Describe the design thinking methodology used to approach this problem.Explain how empathy, ideation, prototyping, and testing were applied to gain insights into customer behaviour and develop solutions.

**1.3 IMPLEMENTATION**

**1. Data Collection:**

-Explain the data sources and the dataset used for analysis.

-Discuss any data preprocessing steps, such as data cleaning and feature engineering.

2. **Data Preprocessing:**

Clean the data by removing duplicates, handling missing values, and ensuring that each transaction is properly formatted.

**3. Analysis:**

-Detail the steps involved in implementing the Apriori algorithm for association analysis.

-Provide code snippets or pseudocode for clarity.

**4. Insights Generation:**

-Present the findings from the market basket analysis.

-Highlight frequently co-occurring products and their significance.

**5. Visualization:**

Present your findings through visualizations like heatmaps, scatter plots, or tables. Create a comprehensive report for stakeholders, highlighting actionable recommendations**.**

**6. Business recommendation:**

Based on market basket insights derived from your analysis, here are some business recommendations to optimize your retail business:

-Cross-Selling Strategies

-Product Placement

-Targeted Marketing

**1.4 ACTIONS**

**-**Propose actionable strategies based on the insights obtained:

-Cross-selling opportunities: Suggest product pairings or promotions.

**1.Inventory management**:

- Recommend stocking related items together.

**2.Customer segmentation:**

**-**Identify distinct purchasing behavior groups.

**3.Marketing campaigns:**

**-**Tailor marketing efforts based on insights.

**2. REQUIREMENTS AND FLOW**

**2.1 SOFTWARES**

**Python:**

Python is a versatile and widely used programming language for data analysis. Several libraries are available for market basket analysis, including:

**MLxtend:** A Python library that provides a straightforward implementation of the Apriori algorithm and association rule mining.

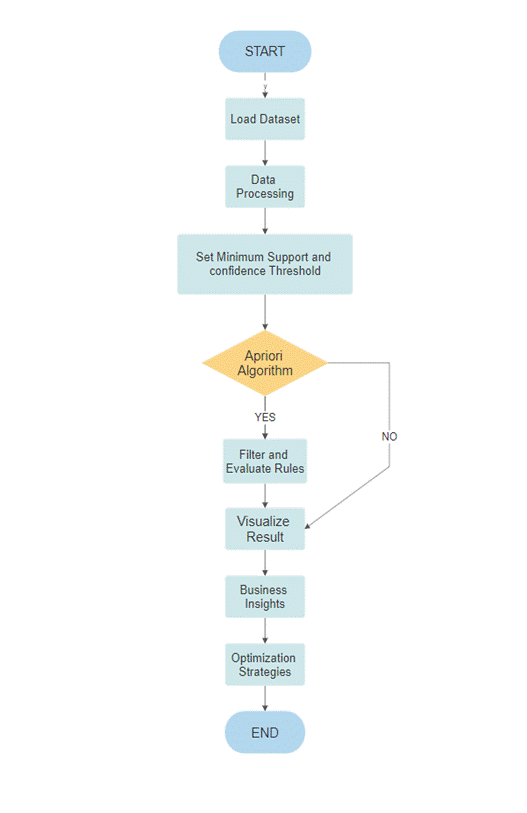
**Pandas:** Often used for data preprocessing and manipulation before applying market basket analysis algorithms.

**Algorithm:**

**Apriori Algorithm:** The Apriori algorithm is one of the most commonly used algorithms for market basket analysis. It identifies frequent itemsets and generates association rules based on minimum support and confidence thresholds.

**MBA (Market Basket Analysis) Package in R:** R offers various packages for association rule mining, including the arules package.

**2.2 FLOWCHAR**

****

**1.     Load Dataset:**

Retrieve and load the transaction dataset containing customer purchase data.

**2.     Data Preprocessing:**

  Clean and format the data, handling missing values or duplicates.

  Convert data into a suitable format for analysis, such as a transaction-item matrix.

**3.     Set Minimum Support and Confidence Thresholds:**

Define the minimum support and confidence thresholds for the Apriori algorithm.

**4.     Apriori Algorithm:**

  Apply the Apriori algorithm to find frequent itemsets.

  Generate association rules based on the frequent itemsets.

**5.     Filter and Evaluate Rules:**

  Filter out rules that meet the minimum support and confidence thresholds.

  Evaluate the remaining rules, considering lift, conviction, or other metrics.

**6.     Visualize Results:**

Create visualizations (e.g., bar charts, network diagrams) to represent the discovered associations and patterns.

**7.     Business Insights:**

Analyze the generated rules and patterns to gain insights into customer behavior and cross-selling opportunities.

**8.     Optimization Strategies:**

Based on the insights, develop strategies for optimizing the retail business, such as product placement, bundling, or promotions.

Conclude the flowchart.

**2.3 DATASETS**

**Retail Store Transactions:** Transaction data from retail stores, such as supermarkets or grocery stores, are frequently used. These datasets typically include information on items purchased in each transaction.

**E-commerce Sales Data**: E-commerce platforms collect transaction data that is rich in product information, customer behavior, and online interactions. This data is valuable for understanding online shopping patterns.

**Restaurant Orders:** Data from restaurants can be used to analyze customer preferences and associations between menu items. It may include dine-in, takeout, or delivery orders.

**Online Shopping Carts:** E-commerce websites often provide data on items added to shopping carts or wish lists. This information can be analyzed to suggest related products or optimize the shopping experience.

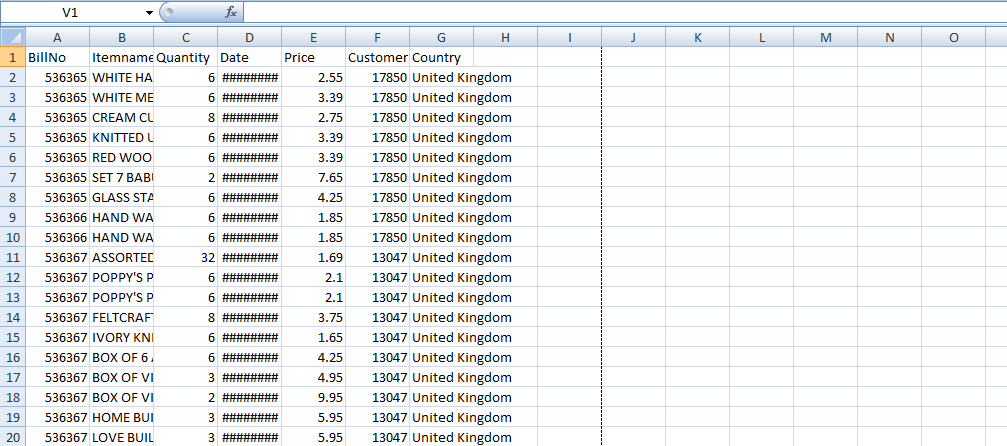
**Customer Loyalty Programs:** Companies with customer loyalty programs can use data from these programs to analyze purchase histories and customer preferences.

**Supermarket Basket Data:** Large supermarket chains collect data on the contents of customers' shopping baskets, which can be used for market basket analysis.

**Gaming and Entertainment:** Game developers and entertainment platforms use transaction data to understand user behavior, in-game purchases, and content preferences

**3.DATALOADING**

**3.1 DATASET:**

****

**Note:**

The above data set is saved as excel sheet in the name of dataset.xlsx.

import pandas as pd

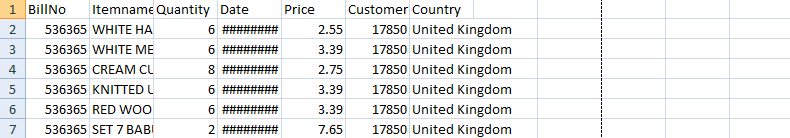
# loading the dataset into a dataframe

df = pd.read\_excel(‘C:\Users\jesus\Desktop\IBM\dataset.xlsx’)

**3.2 PREPROCESSING THE DATASET:**

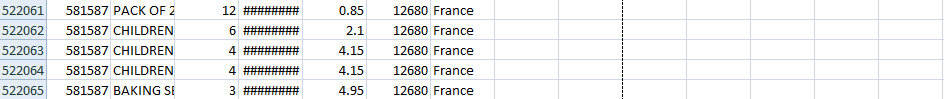
# Display the first 6 rows of data

df.head()



# Display the last 5 rows of data

df.tail()



# Display the shape of the data (number of rows and columns)

df.shape

( 522065,**7**)

# Display the column names

df.columns

Index([‘BillNo’, ‘Itemname’, ‘Quantity’, ‘Date’, ‘Price’, ‘CustomerID’, ‘Country’], datatype = ‘object’)

# Display the data types of each column

df.dtypes

BillNo float64

Itemname object

Quantity float64

Date object

Price float64

CustomerID float64

Country object

**4. VISUALIZATION**

**4.1 DATA TRANSACTION**

Formatting the transaction data in a suitable format for analysis.Developing the preprocessed data into analysis. Split the 'Itemname' column in transaction\_data into individual items using str.split(', ', expand=True).Concatenate the original DataFrame (transaction\_data) with the items DataFrame (items\_df) using pd.concat.Drop the original 'Itemname' column since individual items are now in separate columns.Display the resulting DataFrame.

df.dropna(inplace=True)

transaction\_data = df.groupby(['BillNo', 'Date'])['Itemname'].apply(lambda x: ', '.join(x)).reset\_index()

#Drop Unnecessary Columns

columns\_to\_drop = ['BillNo', 'Date']

transaction\_data.drop(columns=columns\_to\_drop, inplace=True)

# Save the transaction data to a CSV file

transaction\_data\_path = 'transaction\_data.csv'

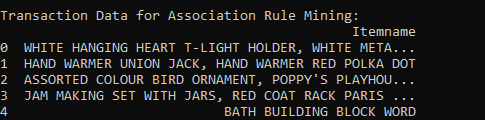
transaction\_data.to\_csv(transaction\_data\_path, index=False)

# Display the first few rows of the transaction data

print("\nTransaction Data for Association Rule Mining:")

print(transaction\_data.head())

transaction\_data.shape



items\_df = transaction\_data['Itemname'].str.split(', ', expand=True)

# Concatenate the original DataFrame with the new items DataFrame

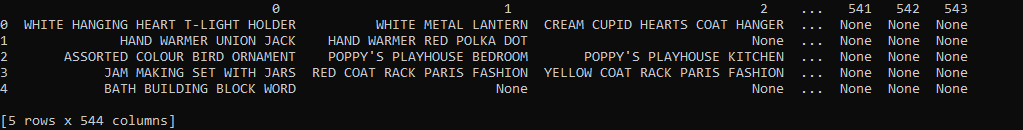
transaction\_data = pd.concat([transaction\_data, items\_df], axis=1)

# Drop the original 'Itemname' column

transaction\_data = transaction\_data.drop('Itemname', axis=1)

# Display the resulting DataFrame

print(transaction\_data.head())



**4.2 DATA MINING**

**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)

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

**4.3 DATA VISUALIZATION**

## 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)')

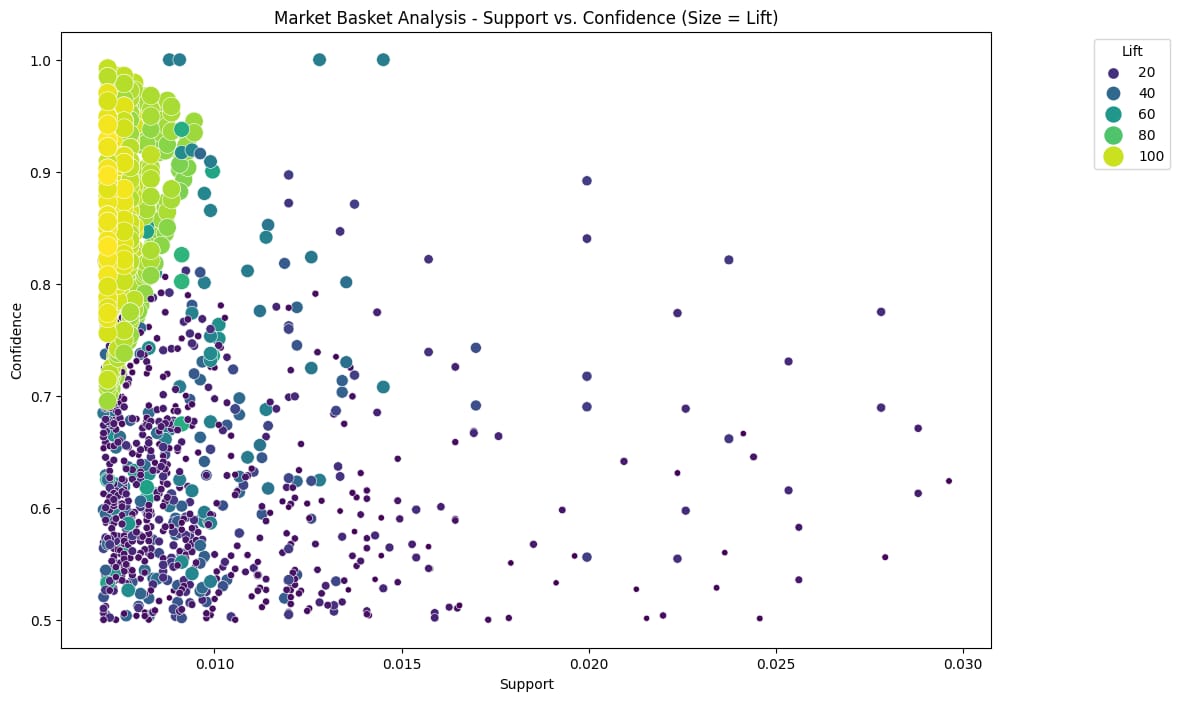
plt.xlabel('Support')

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',

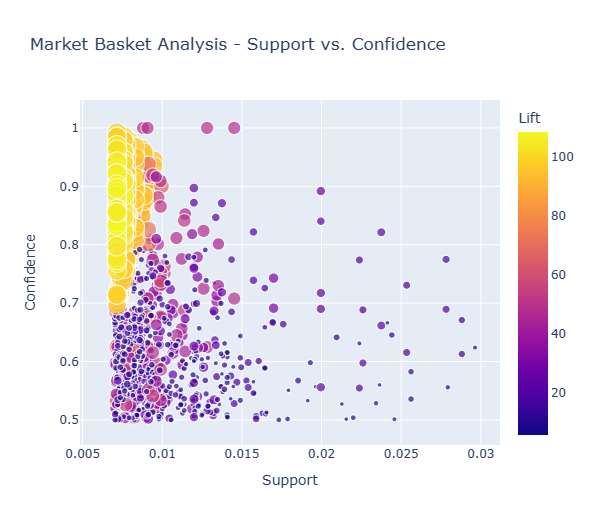
showlegend=True

)

# Show the interactive plot

fig.show()

**OUTPUT:**

****