ABSTRACT

Market Basket Analysis (MBA) is a powerful data mining technique used to uncover associations and relationships among items purchased together in a transaction. It is widely applied in the retail and e-commerce industries to understand customer buying patterns, optimize product placement, and design targeted marketing strategies. This study focuses on the application of the Apriori algorithm to identify frequent itemsets and generate association rules based on transaction data.The dataset used in this project contains records of items purchased in various transactions. The Apriori algorithm systematically identifies item combinations that frequently occur together by applying minimum support and confidence thresholds. Support measures the frequency of occurrence of an itemset in the dataset, while confidence reflects the likelihood of purchasing one item when another is already purchased. The lift metric is also used to evaluate the strength of these associations beyond random chance.The implementation is carried out in Python, leveraging libraries such as Pandas for data manipulation, Mlxtend for association rule mining, and Matplotlib/Seaborn for visualization. The workflow includes data preprocessing, frequent itemset generation, rule extraction, and graphical representation of results. Key findings highlight strong relationships between certain products, enabling retailers to make informed decisions such as bundling products, optimizing shelf placement, and personalizing recommendations for customers.The results demonstrate that Market Basket Analysis can significantly enhance business intelligence by providing actionable insights into customer behavior. The approach adopted in this work is scalable and can be applied to various datasets across domains, including retail, healthcare, and web usage mining. Future work can extend this analysis by incorporating temporal patterns, customer segmentation, and advanced algorithms like FP-Growth to improve efficiency and performance.Market Basket Analysis (MBA) is a data mining technique used to discover associations and correlations among items purchased together in transactions. By applying association rule learning methods, such as the Apriori algorithm, businesses can identify frequent itemsets and generate rules that reveal customer buying patterns. These insights help in cross-selling, product placement, and promotional strategies. This project implements MBA using Python's mlxtend library, calculating key metrics like **Support**, **Confidence**, and **Lift** to evaluate the strength of the rules. The implementation is carried out in Python, leveraging libraries such as Pandas for data manipulation, Mlxtend for association rule mining, and Matplotlib/Seaborn for visualization. The workflow includes data preprocessing, frequent itemset generation, rule extraction, and graphical representation of results. Key findings highlight strong relationships between certain products, enabling retailers to make informed decisions such as bundling products, optimizing shelf placement, and personalizing recommendations for customers.

INTRODUCTION:

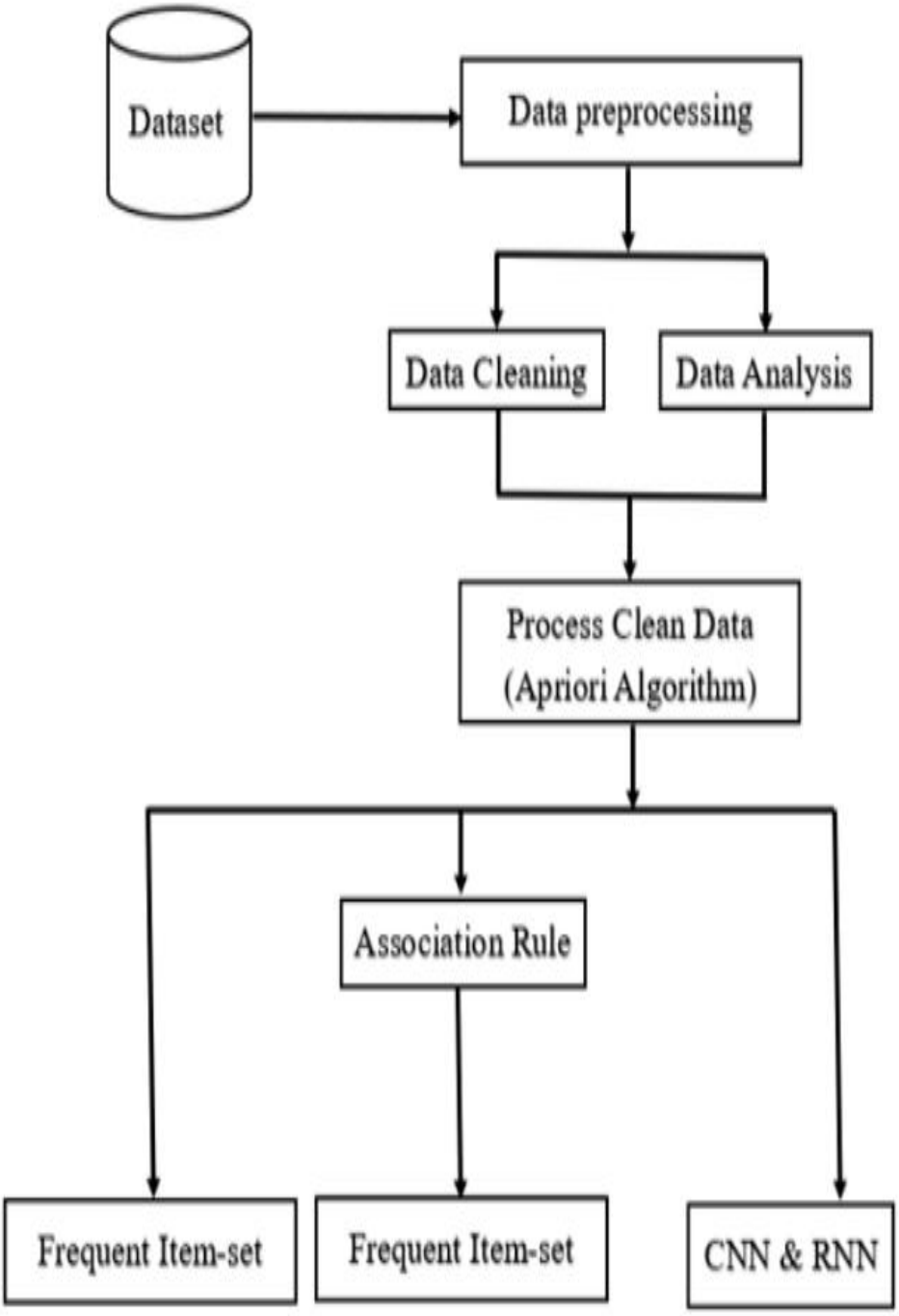
In today's competitive retail environment, understanding customer purchasing behavior is crucial for improving sales, designing effective marketing strategies, and enhancing customer satisfaction. One of the most widely used techniques for uncovering such patterns is **Market Basket Analysis (MBA)**. MBA is a data mining approach that identifies relationships between items purchased together in transactions. It is based on the principle that if a customer buys a certain set of items, they are more likely to buy another related set of items.This technique is particularly valuable for retailers, supermarkets, and e-commerce platforms as it helps in **cross-selling, product placement optimization, and targeted promotions**. By analyzing transaction data, MBA uncovers **association rules** that describe how products are linked. For example, if many customers who purchase bread also purchase butter, the system can generate a rule like: *"If Bread → Butter"*. Such insights allow businesses to place complementary products nearby or create bundle offers, thus increasing revenue.Market Basket Analysis typically uses algorithms such as the **Apriori**, **FP-Growth**, or **Eclat** methods to discover frequent itemsets and generate association rules. The effectiveness of these algorithms is evaluated using statistical measures such as **Support, Confidence, and Lift**.

IMPLEMENTATION:

The Market Basket Analysis was implemented in Python using the **Apriori algorithm** from the mlxtend library to identify frequent itemsets and generate association rules.

1. **Data Collection** – A supermarket transaction dataset was sourced from GitHub containing customer purchase records.
2. **Data Preprocessing** – The dataset was cleaned and converted into a transaction format suitable for association rule mining.
3. **Apriori Algorithm** – Applied to extract frequent itemsets with a minimum support threshold.
4. **Association Rules** – Generated using metrics such as **support**, **confidence**, and **lift** to determine the strength of relationships.
5. **Visualization** – Multiple charts (bar plots, scatter plots) were created using **Matplotlib** and **Seaborn** to display top purchased items and rule metrics for easy analysis.

ARCHITECTURE DIAGRAM:



ALGORITHM:

1. **Start the program**
   Begin by collecting the transaction dataset. Each transaction contains a list of items purchased together.
2. **Preprocessing the Data**
   Convert the dataset into a structured format where each row represents a transaction and each column represents an item, marked as 1 (purchased) or 0 (not purchased).
3. **Set Threshold Values**
   Define minimum **support** and **confidence** values.
   - **Support**: Measures how often an itemset appears in the dataset.
   - **Confidence**: Measures how often items in Y appear in transactions that contain X.
4. **Generate Frequent Itemsets**
   - Start with single items and calculate their support.
   - Keep only those items that meet the minimum support threshold.
   - Combine frequent items to create larger itemsets and repeat the process.
5. **Generate Association Rules**
   From the frequent itemsets, generate rules in the form **X → Y**.
   Each rule is tested against the confidence threshold.
6. **Evaluate Rules with Lift**
   Calculate **Lift** to measure the strength of the rule.
   - Lift > 1 indicates a strong positive association.
7. **Output and Visualization**
   Display the frequent itemsets and rules in tabular form and visualize them using graphs for better understanding.
8. **End The Program**

The **Apriori algorithm** is one of the most widely used methods for association rule mining, especially in Market Basket Analysis. It helps identify relationships between items in large datasets of transactions.

PROGRAM:

```
#Done by 2117230020090

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from mlxtend.preprocessing import TransactionEncoder

from mlxtend.frequent_patterns import apriori, association_rules

url = "https://raw.githubusercontent.com/Debasishsaha123/MARKET-BASKET-ANALYSIS/main/Market_Basket_Optimisation%20(1).csv"

df = pd.read_csv(url, header=None)

print("Raw Sample Transactions:")

print(df.head())

transactions = df.fillna('').values.tolist()

transactions = [[item.strip() for item in trans if item.strip() != ''] for trans in transactions]

te = TransactionEncoder()

te_ary = te.fit(transactions).transform(transactions)

onehot = pd.DataFrame(te_ary, columns=te.columns_)

print("\nOne-Hot Encoded Sample:")

print(onehot.head())

frequent_itemsets = apriori(onehot, min_support=0.01, use_colnames=True)

frequent_itemsets = frequent_itemsets.sort_values(by="support", ascending=False)

print("\nTop Frequent Itemsets:")

print(frequent_itemsets.head(10))

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.2)

rules = rules.sort_values(by="confidence", ascending=False)

print("\nTop Association Rules:")

print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(10))

plt.figure(figsize=(10, 6))

item_counts = onehot.sum().sort_values(ascending=False).head(10)
```

```
item_counts.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title("Top 10 Most Frequently Purchased Products", fontsize=16)
plt.xlabel("Products", fontsize=12)
plt.ylabel("Number of Purchases", fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
plt.figure(figsize=(10, 6))
top_support = frequent_itemsets.head(10)
sns.barplot(
    x="support",
    y=top_support["itemsets"].apply(lambda x: ', '.join(list(x))),
    palette="viridis"
)
plt.title("Top 10 Frequent Itemsets by Support", fontsize=16)
plt.xlabel("Support", fontsize=12)
plt.ylabel("Itemsets", fontsize=12)
plt.tight_layout()
plt.show()
plt.figure(figsize=(10, 6))
top_conf = rules.head(10)
sns.barplot(
    x=top_conf["confidence"],
    y=top_conf.apply(lambda r: f"{', '.join(list(r['antecedents']))} → {',
'.join(list(r['consequents']))}", axis=1),
    palette="coolwarm"
)
plt.title("Top 10 Association Rules by Confidence", fontsize=16)
plt.xlabel("Confidence", fontsize=12)
plt.ylabel("Rule", fontsize=12)
```

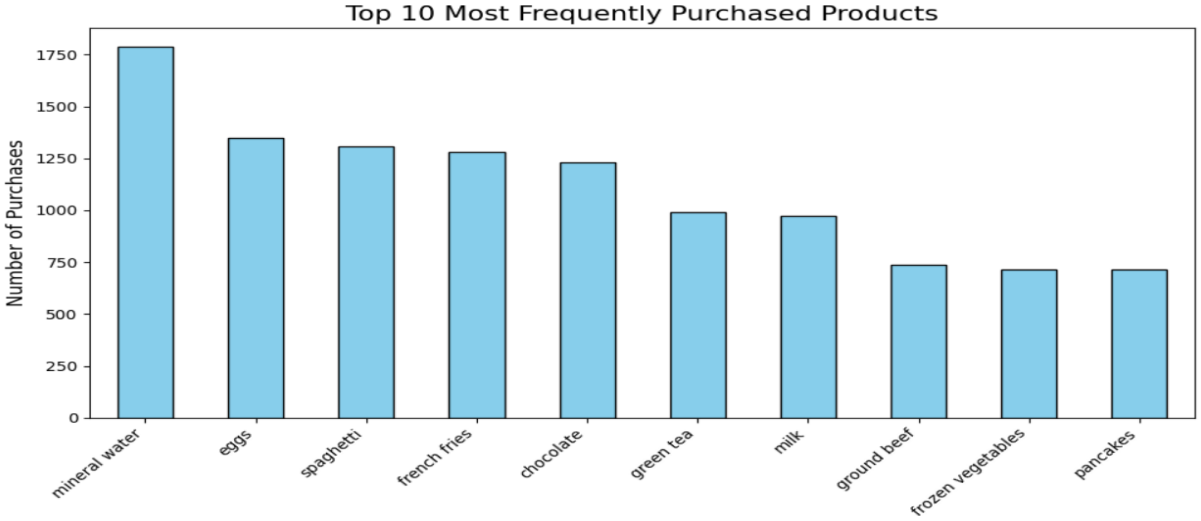
```
plt.tight_layout()
plt.show()
plt.figure(figsize=(10, 6))
top_lift = rules.sort_values(by="lift", ascending=False).head(10)
sns.barplot(
    x=top_lift["lift"],
    y=top_lift.apply(lambda r: f"{', '.join(list(r['antecedents']))} → {', '.join(list(r['consequents']))}", axis=1),
    palette="magma"
)
plt.title("Top 10 Association Rules by Lift", fontsize=16)
plt.xlabel("Lift", fontsize=12)
plt.ylabel("Rule", fontsize=12)
plt.tight_layout()
plt.show()
plt.figure(figsize=(8, 6))
sns.scatterplot(
    x="support",
    y="confidence",
    size="lift",
    data=rules,
    alpha=0.7,
    sizes=(50, 500),
    hue="lift",
    palette="viridis"
)
plt.title("Support vs Confidence (Bubble Size = Lift)", fontsize=16)
plt.xlabel("Support", fontsize=12)
plt.ylabel("Confidence", fontsize=12)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
```

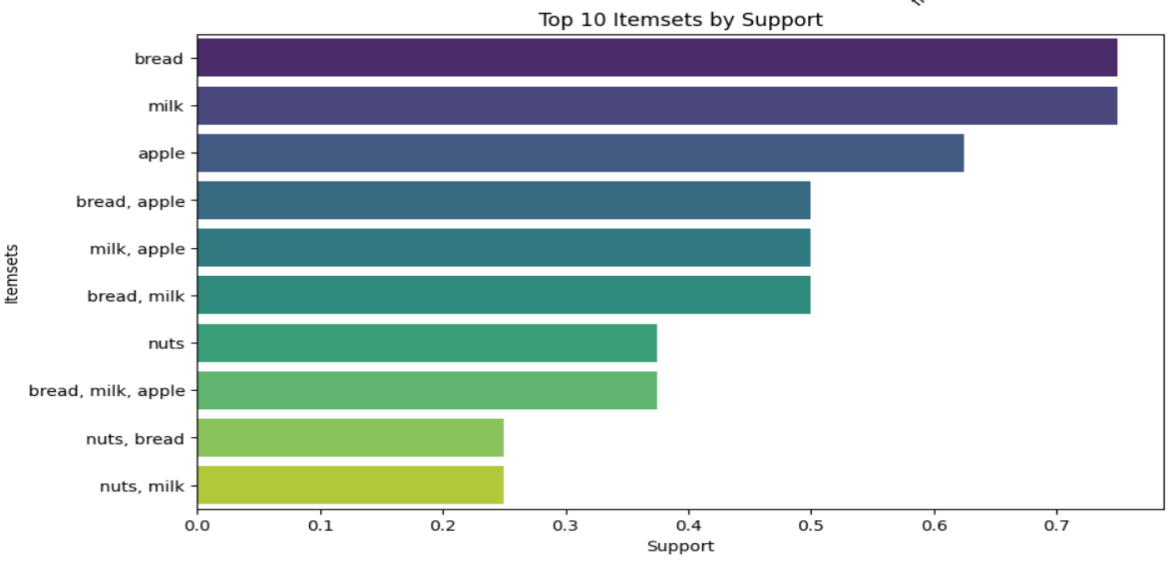
```
plt.tight_layout()

plt.show()

frequent_itemsets.to_csv("frequent_itemsets.csv", index=False)

rules.to_csv("association_rules.csv", index=False)

print("\n Analysis complete. Results and 5 charts generated.")
```

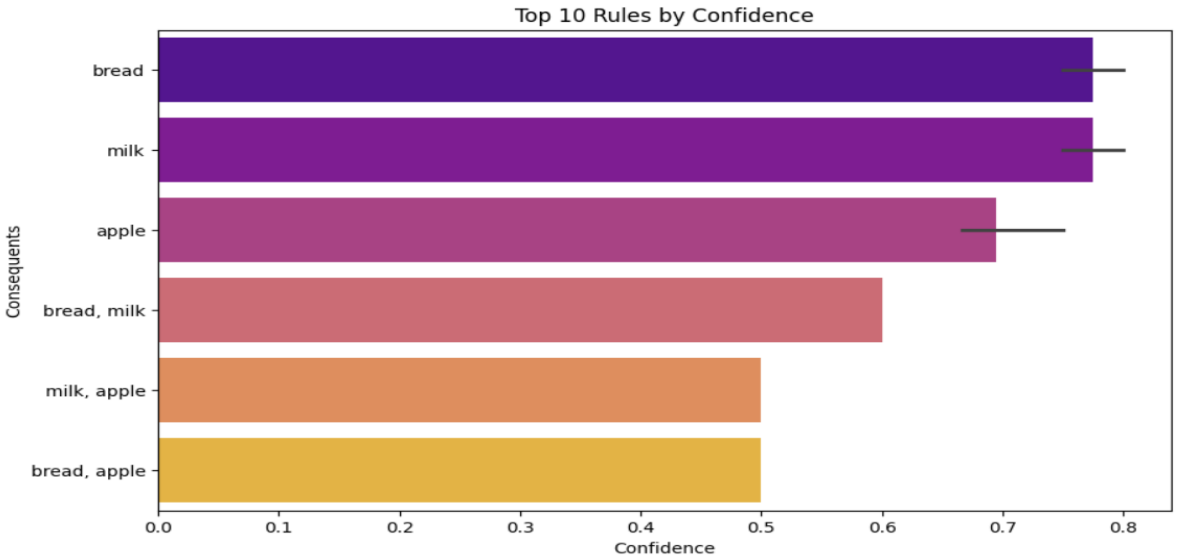
OUTPUT:

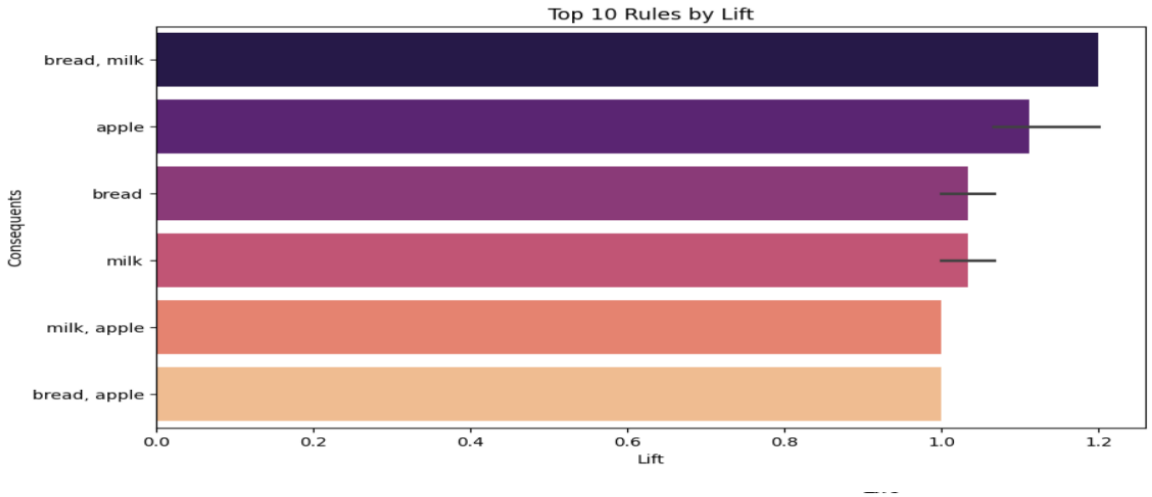
```
Raw Sample Transactions:
              0          1           2                 3             4
0       shrimp     almonds     avocado    vegetables mix  green grapes
1      burgers  meatballs        eggs               NaN           NaN
2      chutney        NaN         NaN               NaN           NaN
3       turkey    avocado         NaN               NaN           NaN
4  mineral water       milk  energy bar  whole wheat rice     green tea

                  5     6                7              8              9   \
0  whole weat flour  yams  cottage cheese  energy drink  tomato juice
1               NaN   NaN              NaN           NaN           NaN
2               NaN   NaN              NaN           NaN           NaN
3               NaN   NaN              NaN           NaN           NaN
4               NaN   NaN              NaN           NaN           NaN

                10          11     12     13              14      15  \
0  low fat yogurt  green tea  honey  salad  mineral water  salmon
1             NaN         NaN    NaN    NaN             NaN     NaN
2             NaN         NaN    NaN    NaN             NaN     NaN
3             NaN         NaN    NaN    NaN             NaN     NaN
4             NaN         NaN    NaN    NaN             NaN     NaN

                  16               17       18          19
0  antioxydant juice  frozen smoothie  spinach  olive oil
1                NaN              NaN      NaN         NaN
2                NaN              NaN      NaN         NaN
3                NaN              NaN      NaN         NaN
4                NaN              NaN      NaN         NaN

One-Hot Encoded Sample:
   almonds  antioxydant juice  asparagus  avocado  babies food  bacon  \
0     True               True      False     True        False  False
1    False              False      False    False        False  False
2    False              False      False    False        False  False
3    False              False      False     True        False  False
4    False              False      False    False        False  False

   barbecue sauce  black tea  blueberries  body spray  ...  turkey  \
0           False      False        False       False  ...   False
1           False      False        False       False  ...   False
2           False      False        False       False  ...   False
3           False      False        False       False  ...    True
4           False      False        False       False  ...   False

   vegetables mix  water spray  white wine  whole weat flour  \
0            True        False       False              True
1           False        False       False             False
2           False        False       False             False
3           False        False       False             False
4           False        False       False             False
```

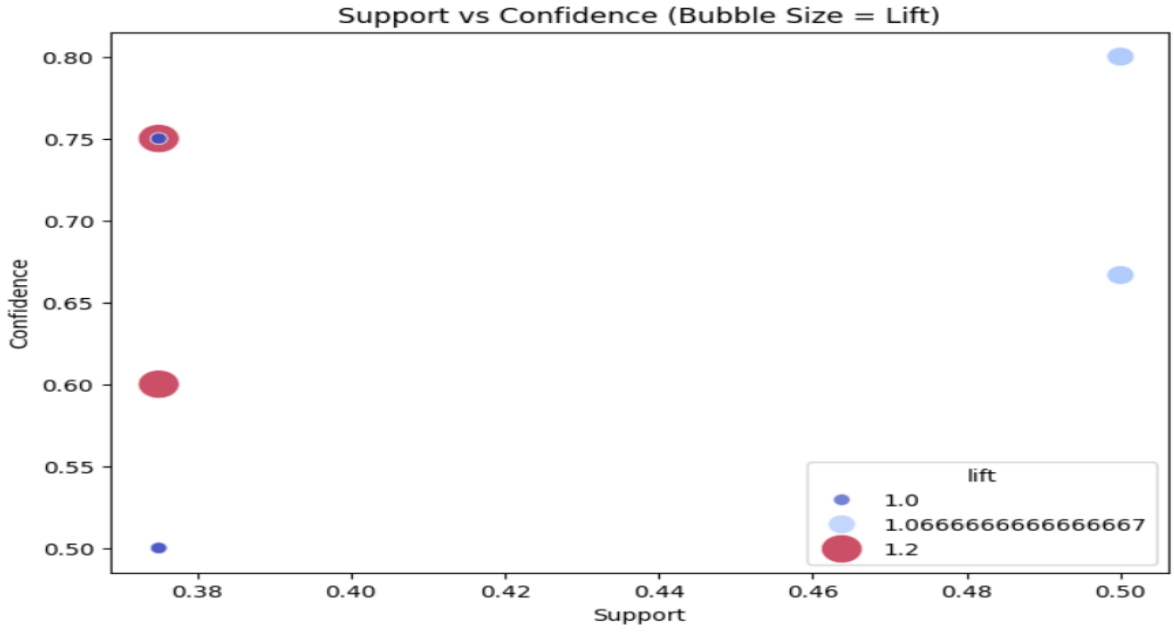
```
Top Frequent Itemsets:
      support                itemsets
46   0.238368       (mineral water)
19   0.179709                (eggs)
63   0.174110           (spaghetti)
24   0.170911         (french fries)
13   0.163845            (chocolate)
32   0.132116            (green tea)
45   0.129583                 (milk)
33   0.098254          (ground beef)
30   0.095321   (frozen vegetables)
53   0.095054             (pancakes)

Top Association Rules:
                    antecedents       consequents    support  confidence  \
340         (ground beef, eggs)  (mineral water)  0.010132    0.506667
284         (milk, ground beef)  (mineral water)  0.011065    0.503030
289   (chocolate, ground beef)  (mineral water)  0.010932    0.473988
278  (milk, frozen vegetables)  (mineral water)  0.011065    0.468927
45                      (soup)  (mineral water)  0.023064    0.456464
262       (spaghetti, pancakes)  (mineral water)  0.011465    0.455026
327      (olive oil, spaghetti)  (mineral water)  0.010265    0.447674
147           (milk, spaghetti)  (mineral water)  0.015731    0.443609
187           (chocolate, milk)  (mineral water)  0.013998    0.435685
106    (spaghetti, ground beef)  (mineral water)  0.017064    0.435374

         lift
340  2.125563
284  2.110308
289  1.988472
278  1.967236
45   1.914955
262  1.908923
327  1.878079
147  1.861024
187  1.827780
106  1.826477
```

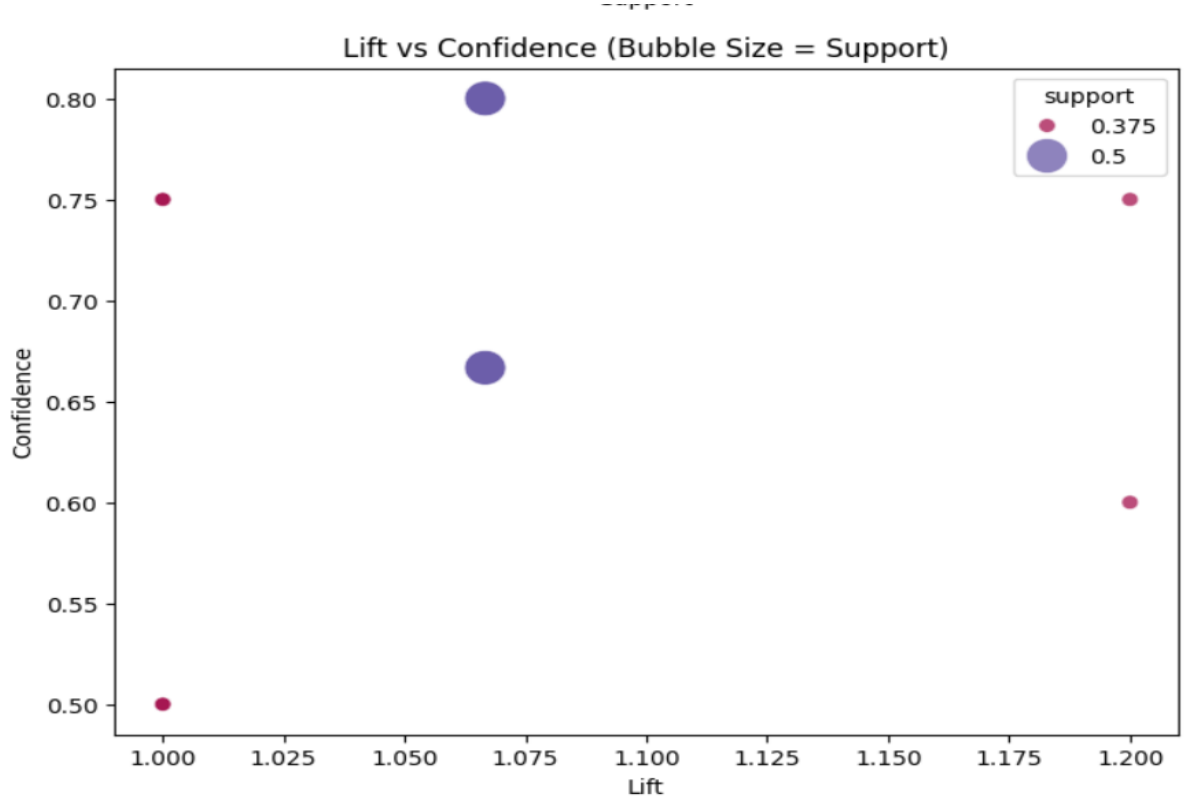





Top 10 Rules by Confidence
bread
milk
apple
bread, milk
milk, apple
bread, apple
Consequents
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Confidence


Top 10 Rules by Lift
bread, milk
apple
bread
milk
milk, apple
bread, apple
Consequents
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Lift


Support vs Confidence (Bubble Size = Lift)
0.80
0.75
0.70
0.65
0.60
0.55
0.50
Confidence
0.38
0.40
0.42
0.44
0.46
0.48
0.50
Support
lift
1.0
1.0666666666666667
1.2



CONCLUSION:

Market Basket Analysis (MBA) is a powerful data mining technique used to uncover hidden relationships between products in transactional datasets. By applying association rule mining methods such as the Apriori algorithm, businesses can identify frequent itemsets and generate rules that reveal which products are often purchased together.In this mini project, transactional data was processed to compute **support**, **confidence**, and **lift** values, enabling the identification of strong and meaningful associations. Visualization through charts helped in better interpreting the results, making it easier to understand customer buying patterns.The analysis can be applied in various domains such as **retail, e-commerce, and inventory management** to improve product placement, create effective marketing campaigns, and enhance cross-selling strategies.Overall, this project demonstrates that MBA is not only a valuable analytical tool for decision-making but also a strategic approach to understanding consumer behavior. By leveraging these insights, businesses can optimize sales and customer satisfaction, ultimately leading to higher profitability and competitive advantage.