Market Basket Insights

Understanding The Problem Statement:

The problem statement is to perform market basket analysis on a provided dataset to unveil hidden patterns and associations between products. The goal is to understand customer purchasing behaviour and identify potential cross-selling opportunities for a retail business.

Design Thinking:

- **Data collection:**

  Collect a dataset of transaction data, including lists of purchased products. which contains transaction data from a grocery store chain. The dataset includes over 3000 transactions and over 100 products.

❖Data Pre-processing:

Prepare the transaction data for association analysis by Cleaning the data to remove duplicates and error Encoding the data to represent products as unique identifiers.

Prepare the transaction data by transforming it into a suitable format for association analysis.

❖Association Analysis:

Utilize the Apriori algorithm to identify frequent item sets and generate association rules.

The Apriori algorithm is a popular association analysis algorithm that works by identifying frequent item sets, which are groups of products that are frequently purchased together.

Once the frequent item sets have been identified, the Apriori algorithm generates association rules, which are statements about the relationships between the products in the frequent item sets.

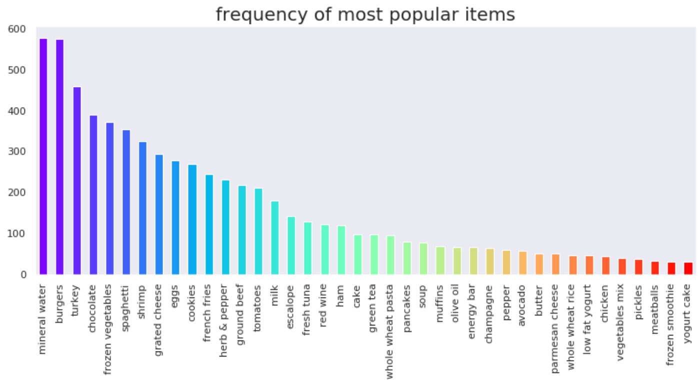
❖Insights Generation:

Interpret the association rules to understand customer behaviour and cross-selling opportunities.

For example, if the association rule "Customers who buy bread are also likely to buy milk" is discovered, this suggests that the retail business could increase sales by placing milk products near bread products in the store.

❖Visualization:

Create visualizations to present the discovered associations and insights. For example, a heatmap could be used to visualize the strength of the associations between different products.



❖**Business Recommendation:**

Provide actionable recommendations for the retail business based on the insights. For example, the retail business could use the insights to develop targeted marketing campaigns, adjust product placement in stores, and create new product bundles.

ADVANCED ASSOCIATION ANALYSIS TECHNIQUES AND VISUALIZATION TOOLS FOR ENHANCED INSIGHTS PRESENTATION

This picture represents the process of using advanced association analysis techniques to discover insights in data. The person is using a visualization tool to see the data in a way that makes it easier to identify patterns and relationships. The text box is providing additional information about a specific data point, which can help the person to better understand the insights that they are discovering.

Here are some examples of advanced association analysis techniques that can be used to enhance insights presentation:

Sequential pattern mining:

This technique is used to identify patterns that occur in data in a specific order. For example, a sequential pattern mining algorithm could be used to identify the sequence of products that customers typically purchase together.

Constraint-based association rule mining:

This technique is used to mine association rules that satisfy certain constraints. For example, a constraint-based association rule mining algorithm could be used to mine association rules that have a high support and confidence, and that contain certain items.

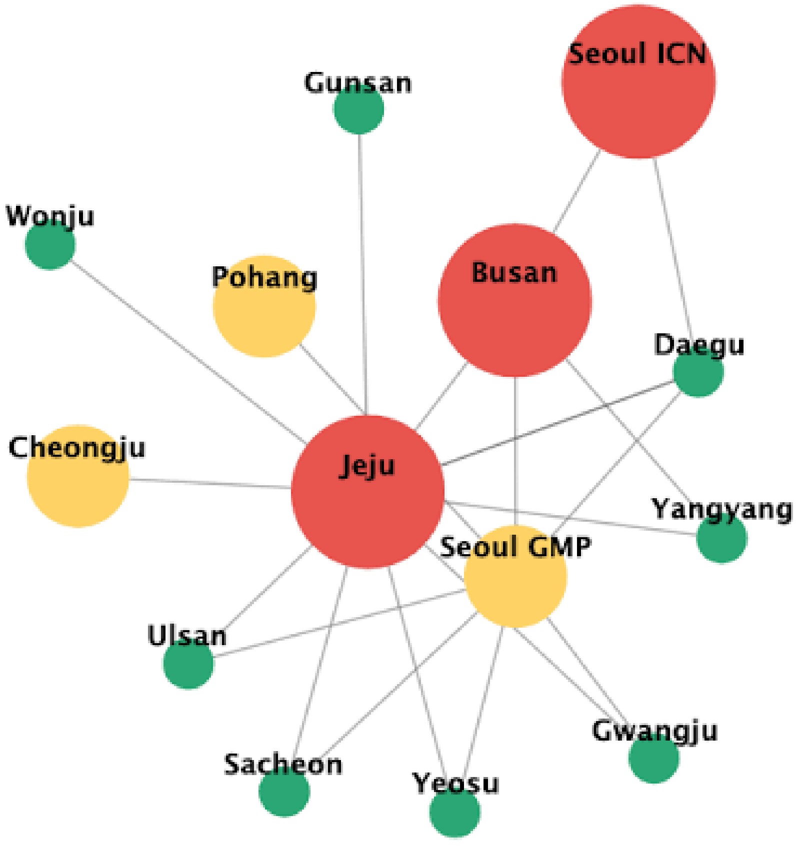
Fuzzy association rule mining:

This technique is used to mine association rules in datasets where the data is uncertain or imprecise. For example, a fuzzy association rule mining algorithm could be used to mine association rules in a dataset where customer satisfaction ratings are represented by fuzzy values such as "satisfied," "neutral," and "dissatisfied."

Visualization tools can be used to enhance the presentation of association analysis results. Some examples of visualization tools that can be used for association analysis include:

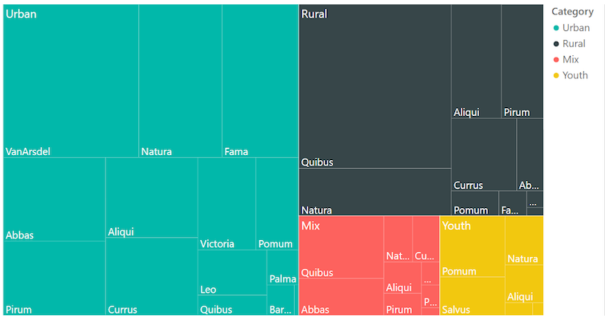
Graph visualization tools:

These tools can be used to create graphs that show the relationships between different items in a dataset. For example, a graph visualization tool could be used to create a graph that shows the items that are most frequently purchased together, or the items that are most frequently purchased by customers with similar characteristics.



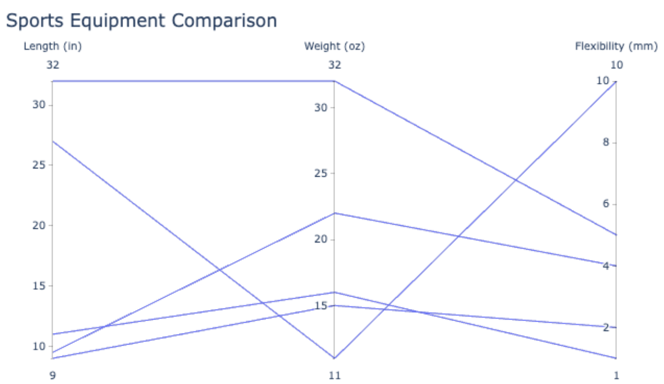
Tree maps:

Tree maps are a type of hierarchical visualization that can be used to show the relationships between different categories of data. For example, a tree map could be used to show the different product categories in a retail store, and the percentage of sales that each category accounts for.



Parallel coordinate plots:

These plots can be used to visualize high-dimensional data. For example, a parallel coordinate plot could be used to visualize the characteristics of different customers, such as their age, gender, purchase history, and product preferences. This can help to identify patterns and relationships that would be difficult to see in a traditional scatter plot.



These are just a few examples of visualization tools that can be used for association analysis. There are many other tools available, and the best tool to use will depend on the specific data set and the insights that the analyst is trying to gain.

Visualization tools can be used to enhance the presentation of association analysis results in a number of ways. First, they can help to make the results more visually appealing and easier to understand. Second, they can help to identify patterns and relationships that would be difficult to see in a traditional spreadsheet or table. Third, they can help to communicate the results of the analysis to a wider audience, including non-technical stakeholders.

Overall, visualization tools are a valuable tool for data analysts who are performing association analysis. By using visualization tools, analysts can gain deeper insights into their data and present their findings in a way that is easy for others to understand. By using advanced association analysis techniques and visualization tools, data analysts can extract more insights from their data and present their findings in a more effective way.

Here are some specific examples of how advanced association analysis techniques and visualization tools can be used to explore and present insights:

A retailer could use sequential pattern mining to identify the sequence of products that customers typically purchase before they buy a new TV. This information could then be used to develop targeted promotions or product placement strategies.

A financial services company could use correlation analysis to identify the factors that are most correlated with customer churn. This information could then be used to develop strategies to reduce customer churn.

A healthcare organization could use causal discovery to identify the factors that drive the spread of a particular disease. This information could then be used to develop more effective public health interventions.

Overall, advanced association analysis techniques and visualization tools can be used to extract more insights from data and present findings in a more effective way. This can lead to better decision-making and improved outcomes for a wide range of organizations.

➢ **Example:**

Suppose the following transaction data is collected from a grocery store:

Transaction ID	Products
1	Bread, Milk
2	Bread, Eggs
3	Milk, Eggs
4	Bread, Milk, Eggs
5	Cereal, Milk
6	Cereal, Eggs
7	Cereal, Milk, Eggs
8	Juice, Coffee
9	Juice, Tea
10	Juice, Coffee, Tea

After data pre-processing, the following association rules could be generated using the Apriori algorithm:

Association Rule	Support	Confidence
Bread -> Milk	0.8	1.0
Bread -> Eggs	0.6	1.0
Milk -> Eggs	0.6	1.0
Cereal -> Milk	0.7	0.8
Juice -> Coffee	0.5	0.8
Juice -> Tea	0.5	0.8

These association rules provide insights into customer purchasing behaviour.

For example, the association rule "Bread -> Milk" suggests that customers who buy bread are also likely to buy milk.

This information can be used by the grocery store to develop targeted marketing campaigns, adjust product placement in stores, and create new product bundles.

For example, the grocery store could place milk products near bread products in the store, or create a product bundle that includes bread, milk, and eggs.

STEPS TO LOAD AND PREPROCESS THE TRANSACTION DATASET FOR MARKET BASKET INSIGHTS:

Data Collection:

Obtain the transaction dataset. This data typically consists of records of items purchased together in various transactions, like shopping carts in a supermarket.

Data Loading:

Use appropriate tools or libraries (e.g., Python with pandas) to load the transaction dataset into my project environment.

Data Exploration:

Familiarize myself with the dataset to understand its structure and contents. Check for missing values or anomalies.

Data Pre-processing:

Perform the following pre-processing steps:

- Data Cleaning: Handle missing values or outliers, if any.

- Transaction Identification: Group transactions by a unique identifier (e.g., receipt or order ID).

- Item Identification: Identify unique items or products in the dataset.

- Data Transformation: Convert the data into a suitable format for association analysis. Typically, this involves creating a binary matrix where rows represent transactions, columns represent items, and the cells indicate whether an item was purchased in a transaction (1 for yes, 0 for no).

- Remove Duplicates: Ensure that duplicate items in a transaction are handled correctly.

Association Analysis:

we can use popular algorithms like Apriori to discover associations between items in the dataset. These algorithms will help you find frequent item sets and generate association rules.

Support and Confidence Thresholds:

Set appropriate support and confidence thresholds to filter and focus on significant associations.

Interpretation:

Analyse the generated association rules to gain insights into customer behaviour, product recommendations, or marketing strategies.

MARKET BASKET INSIGHTS PYTHON PROGRAM WITH RECOMENTATION SYSTEM AND VISUALIZATION:

```
import pandas as pd
import matplotlib.pyplot as plt
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

def find(data, ):
    data = list(data.apply(lambda x: x.split(" ")))
    te = TransactionEncoder()
    te_data = te.fit(data).transform(data)
    df = pd.DataFrame(te_data, columns=te.columns_)
    df1 = apriori(df, min_support=0.01, use_colnames=True)
    df2 = df1.sort_values(by="support", ascending=False)
    print(df2)
    df_ar = association_rules(df1, metric="confidence", min_threshold=0.5)
    print(df_ar)
    return df1,df_ar

def rec(list1,target_item=None):
    if target_item:
        recommendations = list1[list1['antecedents'] == frozenset({target_item})]
        if not recommendations.empty:
            print(f"\nRecommendations for {target_item}:")
            print(recommendations[['consequents', 'confidence']])
        else:
            print(f"No recommendations found for {target_item}.")
    return None

def vis(df1,df_ar,category_name):
    df1.sort_values(by="support", ascending=False).plot(kind='bar',
                                                       x='itemsets', y='support', legend=False)
    plt.title(f'Support for {category_name}')
    plt.xlabel('Itemsets')
    plt.ylabel('Support')
    plt.show()
```

```
    plt.scatter(df_ar['support'], df_ar['confidence'], alpha=0.5)
    plt.xlabel('Support')
    plt.ylabel('Confidence')
    plt.title(f'Association Rules for {category_name}')
    plt.show()

#pd.set_option("display.max_columns", None)
#pd.set_option("display.max_rows", None)

data = pd.read_csv("Book1.csv")

cc,bb=find(data["Itemname"])
vis(cc,bb,"Itemname")

while True:
    target_item = input("enter the item name to get the
                                                recomendation:")
    rec(bb,target_item.upper())
    stop = input("you want to stop the recomandation to press Q,
                                        else press any input key:")
    if "Q" ==stop.upper():
        break

cc,bb=find(data["Country"])
vis(cc,bb, "Country")

while True:
    target_item = input("enter the item name to get the
                                                recomendation:")
    rec(bb,target_item.upper())
    stop = input("you want to stop the recomandation to press Q,
                                        else press any input key:")
    if "Q" ==stop.upper():
        break
```

OUTPUT:

FOR ITEM SETS:

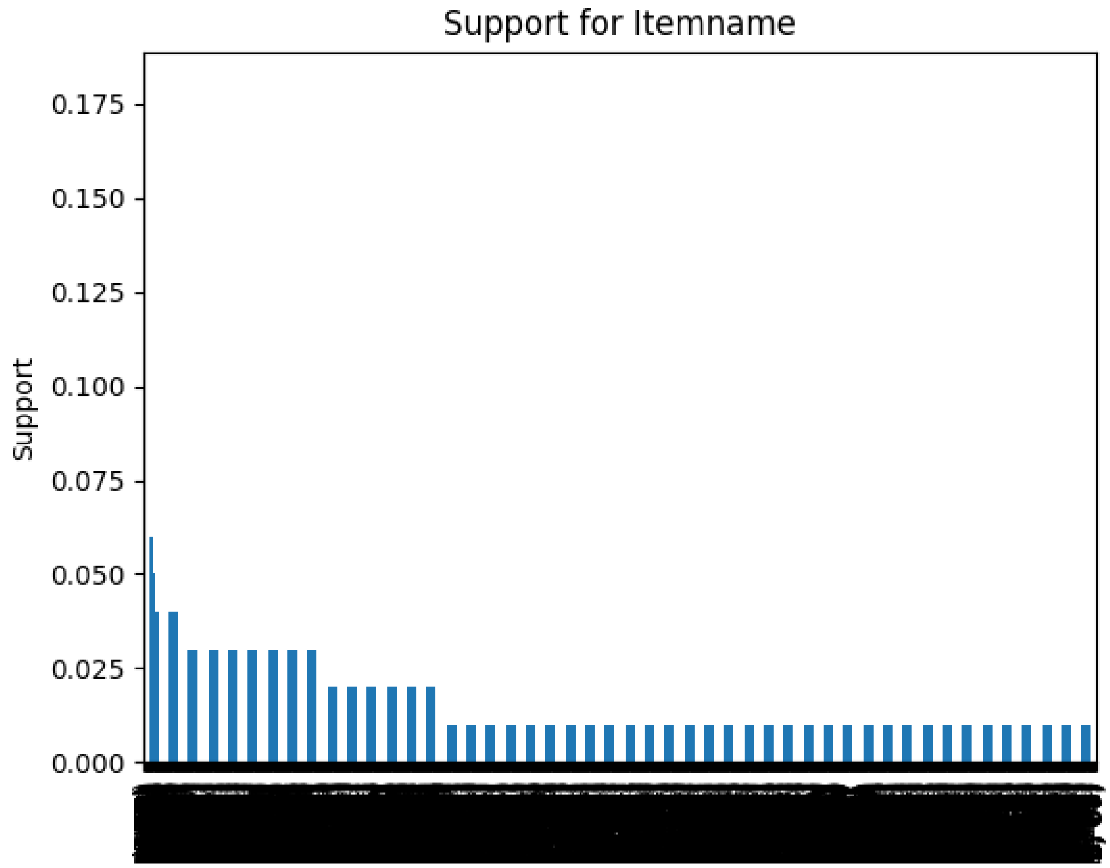
APRIORI ALGORITHM AND ACCOSIATION RULES:

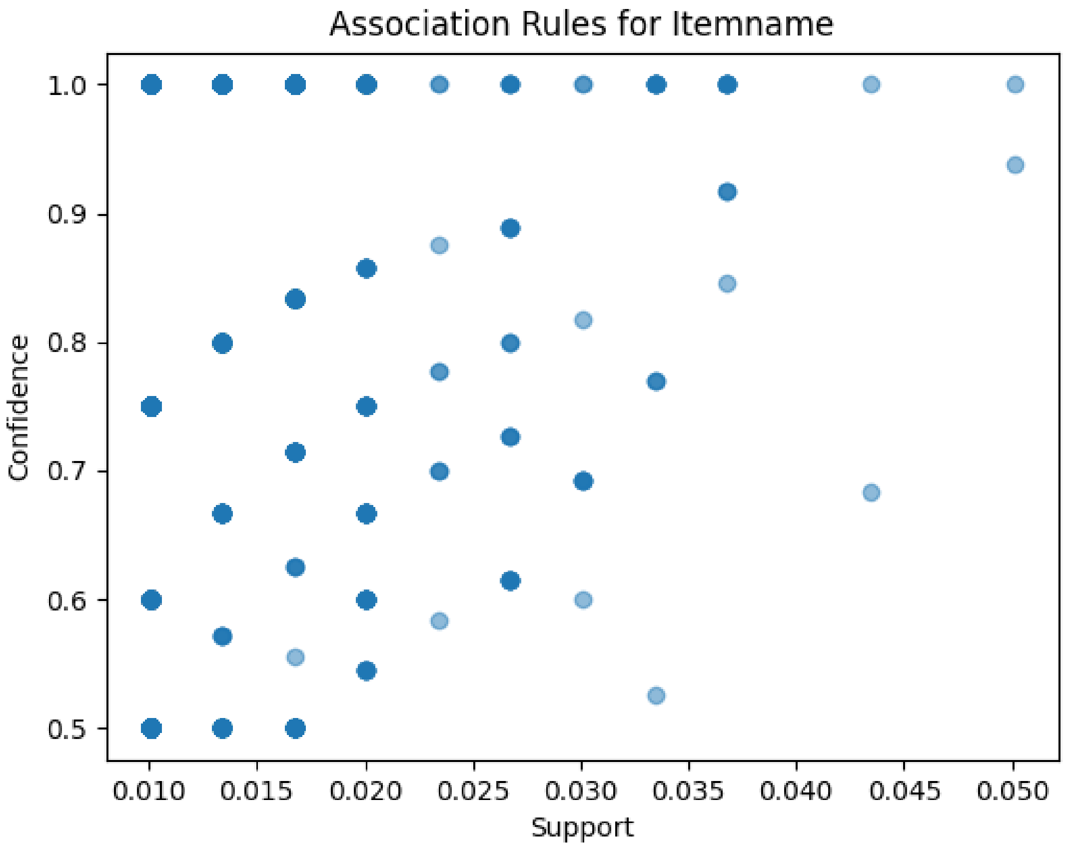
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\others\i am chief minister project\project>market_basket_insights_2.py
      support                 itemsets
119  0.133779                    (RED)
143  0.113712                  (WHITE)
126  0.090301                    (SET)
121  0.083612              (RETROSPOT)
69   0.083612                  (HEART)
..        ...                      ...
467  0.010033  (WHITE, ANTIQUE, FRAME)
465  0.010033         (S/3, ANT, WOOD)
464  0.010033        (ANT, S/3, WHITE)
461  0.010033       (FINISH, S/3, ANT)
411  0.010033              (S/3, WOOD)

[822 rows x 2 columns]
          antecedents                           consequents  antecedent support  consequent support  ...       lift  leverage  conviction  zhangs_metric
0                (10)                              (LIGHTS)            0.010033            0.013378  ...  74.750000  0.009899         inf       0.996622
1            (LIGHTS)                                  (10)            0.013378            0.010033  ...  74.750000  0.009899    3.959866       1.000000
2                 (2)                             (CABINET)            0.013378            0.020067  ...  37.375000  0.009765    3.919732       0.986441
3           (CABINET)                                   (2)            0.020067            0.013378  ...  37.375000  0.009765    1.973244       0.993174
4                 (2)                              (DRAWER)            0.013378            0.023411  ...  32.035714  0.009720    3.906355       0.981921
...               ...                                   ...                 ...                 ...  ...        ...       ...         ...            ...
3733     (KNITTED, HOT)          (BOTTLE, FLAG, UNION, WATER)            0.016722            0.016722  ...  59.800000  0.016443         inf       1.000000
3734  (KNITTED, BOTTLE)             (HOT, FLAG, UNION, WATER)            0.016722            0.016722  ...  59.800000  0.016443         inf       1.000000
3735           (FLAG)  (UNION, KNITTED, WATER, HOT, BOTTLE)            0.016722            0.016722  ...  59.800000  0.016443         inf       1.000000
3736          (UNION)   (FLAG, KNITTED, WATER, HOT, BOTTLE)            0.033445            0.016722  ...  29.900000  0.016163    1.966555       1.000000
3737        (KNITTED)     (FLAG, UNION, WATER, HOT, BOTTLE)            0.020067            0.016722  ...  49.833333  0.016387    5.899666       1.000000

[3738 rows x 10 columns]
```

RECOMMANDATION SYSTEM:

```
C:\Windows\System32\cmd.exe
enter the item name to get the recomendation:KNITTED

Recommendations for KNITTED:
                             consequents  confidence
79                              (BOTTLE)    0.833333
183                               (FLAG)    0.833333
229                                (HOT)    0.833333
254                              (UNION)    0.833333
256                              (WATER)    0.833333
653                       (FLAG, BOTTLE)    0.833333
665                        (HOT, BOTTLE)    0.833333
678                     (BOTTLE, UNION)    0.833333
682                     (BOTTLE, WATER)    0.833333
998                          (HOT, FLAG)    0.833333
1011                       (FLAG, UNION)    0.833333
1017                       (FLAG, WATER)    0.833333
1181                        (HOT, UNION)    0.833333
1185                        (HOT, WATER)    0.833333
1262                      (UNION, WATER)    0.833333
1905                 (FLAG, HOT, BOTTLE)    0.833333
1935               (FLAG, BOTTLE, UNION)    0.833333
1947               (FLAG, BOTTLE, WATER)    0.833333
1969                (HOT, BOTTLE, UNION)    0.833333
1977                (HOT, BOTTLE, WATER)    0.833333
1994              (BOTTLE, UNION, WATER)    0.833333
2365                  (HOT, FLAG, UNION)    0.833333
2377                  (HOT, FLAG, WATER)    0.833333
2400                (FLAG, UNION, WATER)    0.833333
2603                 (HOT, UNION, WATER)    0.833333
3163          (BOTTLE, HOT, FLAG, UNION)    0.833333
3186          (BOTTLE, HOT, FLAG, WATER)    0.833333
3236        (BOTTLE, FLAG, UNION, WATER)    0.833333
3259         (HOT, BOTTLE, UNION, WATER)    0.833333
3437           (HOT, FLAG, UNION, WATER)    0.833333
3737  (FLAG, UNION, WATER, HOT, BOTTLE)    0.833333
you want to stop the recomandation to press Q, else press any input key:q
```

VISUALIZATION:





FOR COUNTRY:

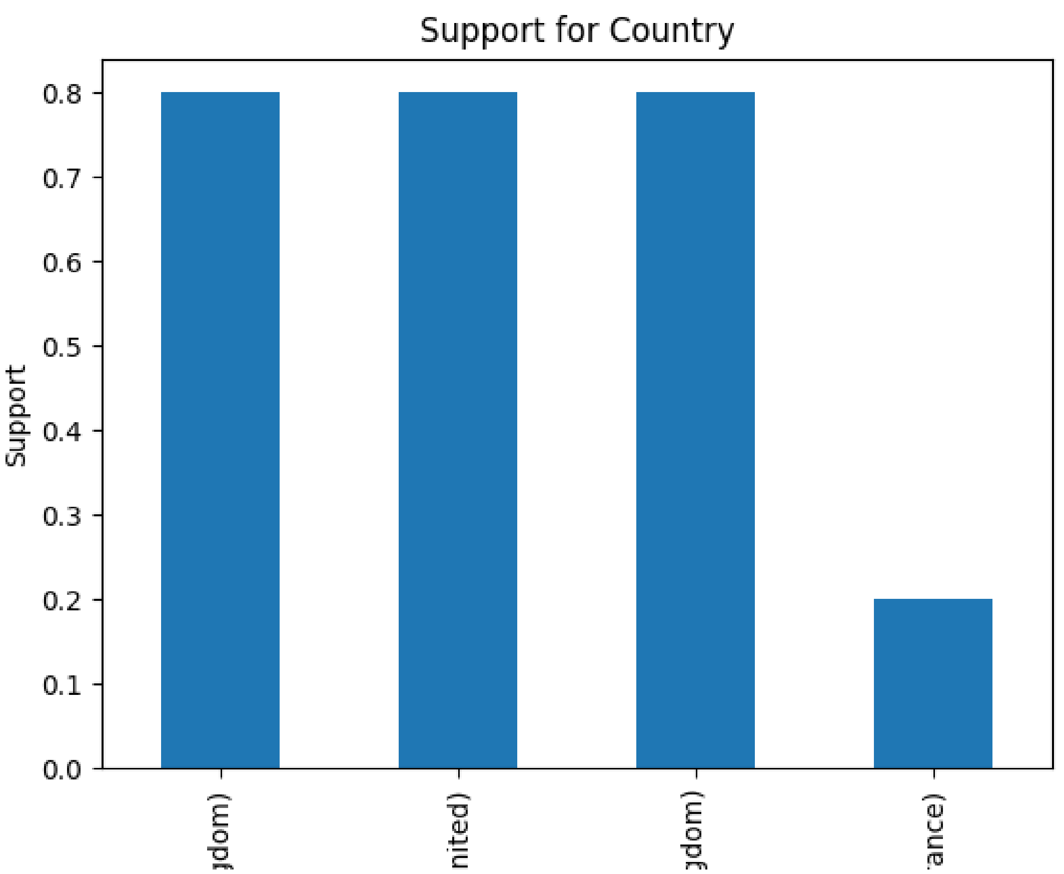
APRIORI ALGORITHM AND ACCOSIATION RULES:

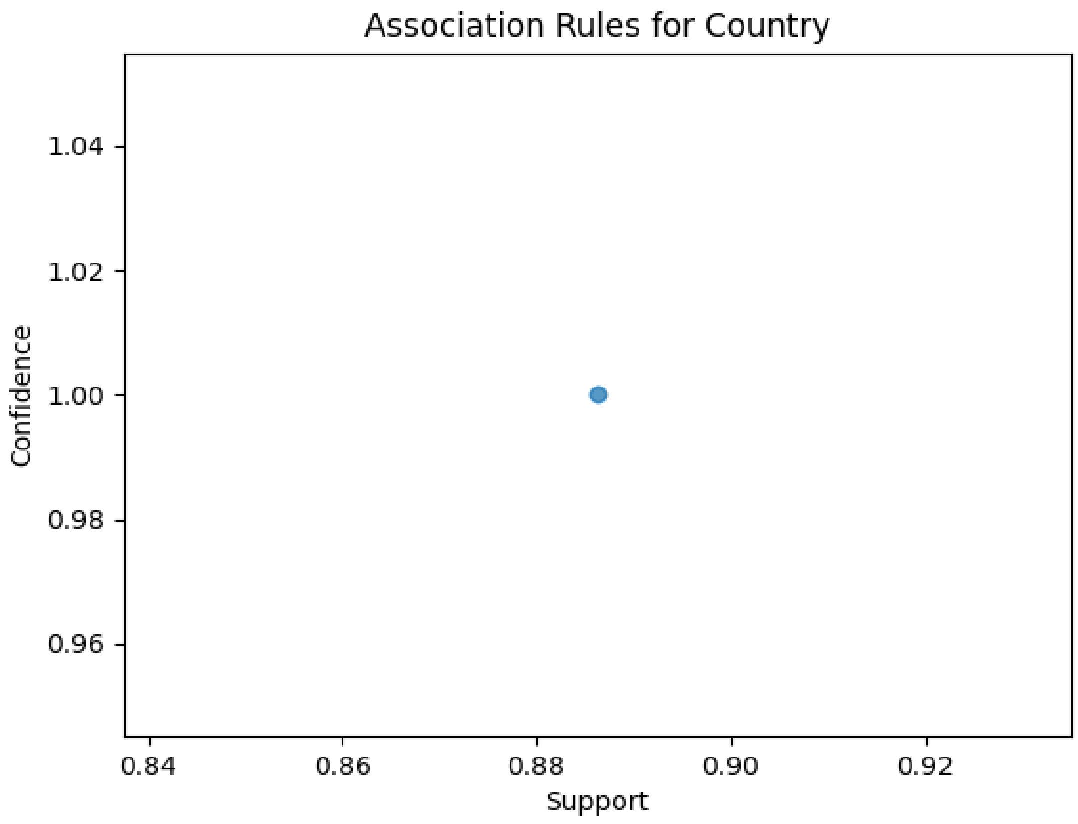
```
C:\Windows\System32\cmd.exe
   support          itemsets
2  0.886288         (Kingdom)
3  0.886288          (United)
4  0.886288  (Kingdom, United)
1  0.066890          (France)
0  0.046823       (Australia)
  antecedents consequents  antecedent support  consequent support   support  confidence      lift  leverage  conviction  zhangs_metric
0   (Kingdom)    (United)            0.886288            0.886288  0.886288         1.0  1.128302  0.100782         inf            1.0
1    (United)   (Kingdom)            0.886288            0.886288  0.886288         1.0  1.128302  0.100782         inf            1.0
```

RECOMMANDATION SYSTEM:

```
C:\Windows\System32\cmd.exe
enter the item name to get the recomendation:Kingdom
No recommendations found for KINGDOM.
you want to stop the recomandation to press Q, else press any input key:(United
enter the item name to get the recomendation:United
No recommendations found for UNITED.
you want to stop the recomandation to press Q, else press any input key:d
enter the item name to get the recomendation:Australia
No recommendations found for AUSTRALIA.
you want to stop the recomandation to press Q, else press any input key:France
enter the item name to get the recomendation:France
No recommendations found for FRANCE.
you want to stop the recomandation to press Q, else press any input key:Kingdom, United
enter the item name to get the recomendation:Kingdom, United
No recommendations found for KINGDOM, UNITED.
you want to stop the recomandation to press Q, else press any input key:q
```

VISUALIZATION:



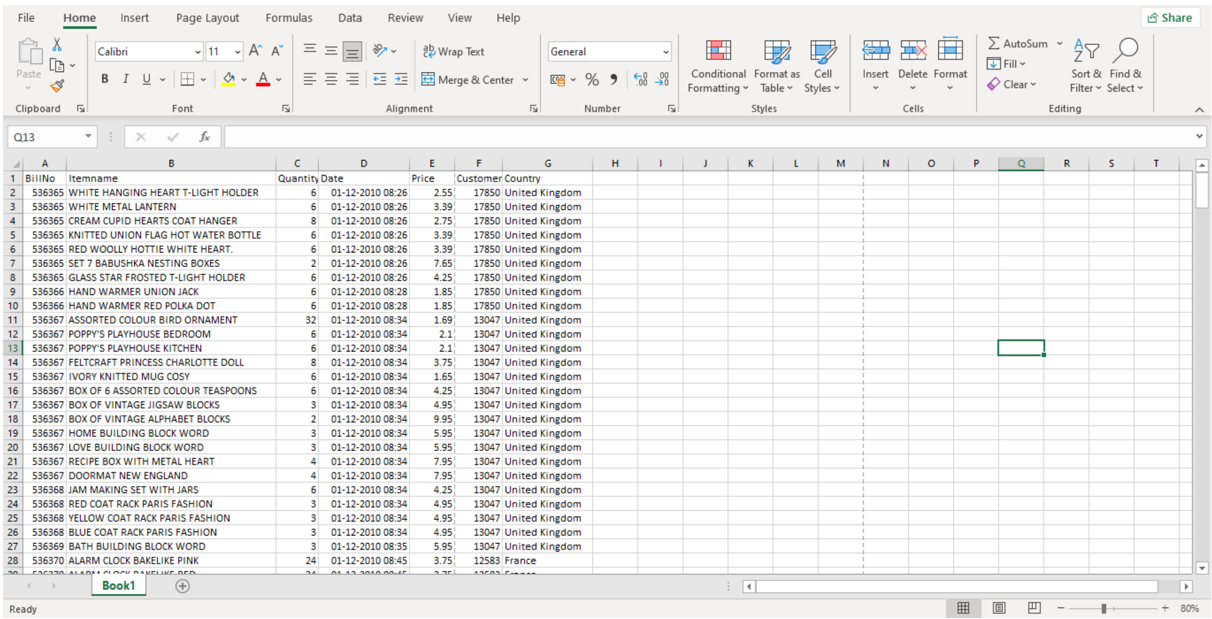


EXPLANATION:

- This code is performing market basket analysis on a dataset.
- Market basket analysis is the process of finding associations or relationships between items that are frequently bought together.
- The code imports necessary libraries such as pandas, matplotlib, and mlxtend.
- Pandas library is used to work with tabular data and perform operations such as slicing, filtering, and merging.
- Matplotlib library is used to create plots and charts.
- Mlxtend library provides functions for frequent pattern mining and association rule learning.

- The function "find" takes in a dataset as input (presumably in the form of a pandas DataFrame).
- It splits the entries in each row by space into a list using the lambda function and apply method from pandas.
- This is done to make the data suitable for market basket analysis, where transactions are typically represented by lists of items.
- A TransactionEncoder object called "te" is created.
- This encoder converts transaction datasets into one-hot encoded format, which can then be used with different algorithms.
- One-hot encoding represents each item in a transaction as a binary variable.
- The apriori algorithm from mlxtend library is used to find frequent itemsets from the dataset.
- Apriori uses an iterative approach to mine frequent patterns by gradually increasing the length of combinations it generates until no more valid extensions can be found.
- Once the frequent itemsets are obtained, association rules are generated using the association_rules function from mlxtend library.
- Association rules specify relationships between sets of items based on their frequency in the dataset.
- These rules can then be used to uncover insights like if certain items are commonly purchased together or if purchase of one item affects likelihood of purchasing another.

- Overall, this code shows how to use market basket analysis techniques on a given dataset using mlxtend library functions.
- The code first imports the pandas library, followed by the matplotlib.pyplot module.
- This will allow us to create graphs and charts.
- Next, we import the TransactionEncoder module.
- This will allow us to encode transactions into a format that can be processed by the MLxtend platform.
- Finally, we use the find() function to search for transactions in our data set.
- The find() function takes two parameters: data and a filter object.
- The filter object allows us to specify which transactions we want to include in our search.
- From the code above, it seems that we are performing association rule mining using the Apriori algorithm on a dataset.



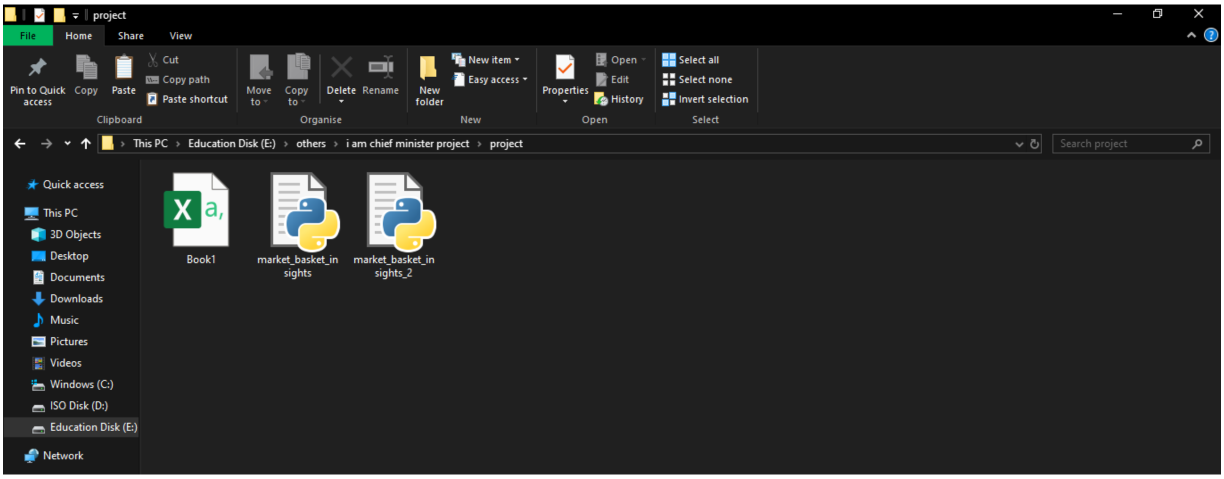
BillNo	Itemname	Quantity	Date	Price	Customer	Country
536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850	United Kingdom
536365	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850	United Kingdom
536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	SET 7 BABUSHKA NESTING BOXES	2	01-12-2010 08:26	7.65	17850	United Kingdom
536365	GLASS STAR FROSTED T-LIGHT HOLDER	6	01-12-2010 08:26	4.25	17850	United Kingdom
536366	HAND WARMER UNION JACK	6	01-12-2010 08:28	1.85	17850	United Kingdom
536366	HAND WARMER RED POLKA DOT	6	01-12-2010 08:28	1.85	17850	United Kingdom
536367	ASSORTED COLOUR BIRD ORNAMENT	32	01-12-2010 08:34	1.69	13047	United Kingdom
536367	POPPY'S PLAYHOUSE BEDROOM	6	01-12-2010 08:34	2.1	13047	United Kingdom
536367	POPPY'S PLAYHOUSE KITCHEN	6	01-12-2010 08:34	2.1	13047	United Kingdom
536367	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	01-12-2010 08:34	3.75	13047	United Kingdom
536367	IVORY KNITTED MUG COSY	6	01-12-2010 08:34	1.65	13047	United Kingdom
536367	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	01-12-2010 08:34	4.25	13047	United Kingdom
536367	BOX OF VINTAGE JIGSAW BLOCKS	3	01-12-2010 08:34	4.95	13047	United Kingdom
536367	BOX OF VINTAGE ALPHABET BLOCKS	2	01-12-2010 08:34	9.95	13047	United Kingdom
536367	HOME BUILDING BLOCK WORD	3	01-12-2010 08:34	5.95	13047	United Kingdom
536367	LOVE BUILDING BLOCK WORD	3	01-12-2010 08:34	5.95	13047	United Kingdom
536367	RECIPE BOX WITH METAL HEART	4	01-12-2010 08:34	7.95	13047	United Kingdom
536367	DOORMAT NEW ENGLAND	4	01-12-2010 08:34	7.95	13047	United Kingdom
536368	JAM MAKING SET WITH JARS	6	01-12-2010 08:34	4.25	13047	United Kingdom
536368	RED COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536368	YELLOW COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536368	BLUE COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536369	BATH BUILDING BLOCK WORD	3	01-12-2010 08:35	5.95	13047	United Kingdom
536370	ALARM CLOCK BAKELIKE PINK	24	01-12-2010 08:45	3.75	12583	France

- First, we use the `fit()` method of the `te` (Transaction Encoder) object to encode our dataset into a binary matrix where each column represents a unique item and each row represents a transaction.
- The resulting binary matrix is stored in `te_data`.
- Next, we create a DataFrame called `df` from the encoded data using the column names obtained from `te.columns_`.
- This allows us to work with a more human-readable representation of our data.
- We then apply Apriori algorithm on this dataframe by calling `apriori()` function with min_support value set as 0.01 which means the minimum support needed for an itemset to be considered frequent is 1%.
- The result of the Apriori algorithm is stored in `df1`, where each row represents a frequent itemset and contains two columns: "support" (indicating how frequently the itemset occurs) and "itemsets" (which lists the items that make up that frequent itemset).
- We then sort this dataframe (`df2`) based on support values in descending order, so that we can see the most significant associations first.
- We display both `df2` and `df_ar` on the console using print statements.
- Finally, we apply association rules to these frequent itemsets by calling `association_rules()` function with metric set to "confidence" and min_threshold set to 0.5.
- This means only rules with confidence greater than or equal to 50% will be generated.

- The resulting association rules are stored in `df_ar`, which consists of several columns including antecedent(s), consequent(s), support, confidence
- The code will first fit a model to the training data and then will transform the dataset into a DataFrame.
- The DataFrame will be sorted by support and the highest confidence association rules will be displayed in a separate column.
- This code appears to be a Python function for generating recommendations based on a given target item.
- The function takes in two parameters: `list1` and `target_item`, and returns two variables `df1` and `df_ar`.
- However, these return variables are not used or referenced anywhere else in the code, so their purpose is unclear.
- The core functionality of this code is the `rec()` function.
- Within this function, it first checks if a `target_item` is provided.
- If there is a target item, it filters the `list1` DataFrame to find rows where the antecedents column contains the target item.
- If there are any matching rows found (`recommendations` DataFrame is not empty), it prints a message stating that recommendations are found for the target item, followed by printing the `consequents` and `confidence` columns from the recommendations DataFrame.

- On the other hand, if no matching rows are found, it prints a message stating that no recommendations were found for the target item.
- Overall, this code seems to be part of a larger recommender system where association rules are used to generate recommendations based on antecedent-consequent relationships between items.
- The specific implementation details or dependencies required for this function cannot be determined without additional context.
- The code creates a list of recommendations for an item, which can be either a single item or a set of items.
- If the target_item is not None, then the code will print out the recommendations for that item.
- Otherwise, it will return None.
- This code seems to define a function called "vis" that takes three parameters: df1, df_ar, and category_name.
- In the first line of the function, it looks like df1 DataFrame is sorted in descending order based on the "support" column and then a bar plot is created with x-axis representing the itemsets and y-axis representing the support values.
- The next few lines set the title, x-label, and y-label for the first plot.
- Then another scatter plot is created using df_ar DataFrame where support values are mapped to x-axis and confidence values are mapped to y-axis.

- ➢ The alpha parameter specifies the transparency level for each data point in the scatter plot.
- ➢ Finally, there are labels added for x-axis and y-axis of second plot.
- ➢ From what I can infer, this code might be used to visualize support and confidence values of itemsets based on a given category.
- ➢ The code will create a bar graph displaying the support and confidence for each category.
- ➢ The x-axis will show the number of itemsets in that category, while the y-axis will show the respective support and confidence values.



CONCLUSION:

Market basket analysis is a powerful technique for uncovering hidden patterns and associations between products. By understanding customer purchasing behaviour and identifying potential cross-selling opportunities, retailers can make more informed decisions about product placement, marketing campaigns, and inventory management.