Market Basket Insights

Description:

Market basket analysis is a data mining technique used by retailers to uncover associations between products that customers tend to purchase together. It helps retailers make informed decisions about product placement, promotions, and inventory management. If you have specific questions or need insights about market basket analysis, please provide more details, and I'll be happy to assist you further.

Definition:

Market Basket Insights is a data mining and analytics technique that focuses on discovering associations and relationships between items that customers buy together within a single shopping transaction or "basket." It's a crucial aspect of retail analytics and helps businesses understand customer behavior.

Data Collection:

To perform Market Basket Analysis, you need transaction data that records which items were purchased together in each customer transaction. This data includes information about the products bought, the time of purchase, and sometimes additional customer details.

Frequent Itemset Mining:

The first step in Market Basket Insights is to identify frequent itemsets, which are combinations of items that appear together in transactions above a predefined minimum support threshold. This threshold helps filter out less relevant associations.

Association Rule Mining:

Once frequent itemsets are identified, the next step is to generate association rules. These rules typically have three key metrics:

Support:

It measures how frequently a specific itemset or rule occurs in the dataset.

- Confidence:

It quantifies the likelihood that if a customer buys one item, they will also buy another item in the same transaction.

- Lift:

Lift compares the likelihood of the two items being purchased together to the likelihood of them being purchased independently. A lift value greater than 1 indicates a positive association.

Example:Let's say you have a dataset of customer transactions in a grocery store. You find that "bread" and "butter" frequently appear together in shopping baskets. An association rule might be: "If a customer buys bread, they are 80% likely to buy butter in the same transaction." The lift value might be 1.2, indicating a positive correlation.

Business Applications:

- Product Placement:

Businesses can strategically place related items closer together in stores or online to encourage cross-selling.

- Marketing:

Targeted marketing campaigns can be designed based on associations. For example, if chips and salsa often go together, you can promote them as a bundle deal.

- Inventory Management:

Helps in optimizing stock levels and reducing waste by anticipating item demand based on associations.

Challenges:

Market Basket Analysis can be computationally intensive, especially with large datasets. Interpretation of results is also crucial, as correlation does not imply causation. Moreover, customer preferences can change over time, so continuous analysis is necessary.

Predictive market basket analysis. This type considers items purchased in sequence to determine cross-sell.

Differential market basket analysis. This type considers data across different stores, as well as purchases from different customer groups during different times of the day, month or year. If a rule holds in one dimension, such as store, time period or customer group, but does not hold in the others, analysts can determine the factors responsible for the exception. These insights can lead to new product offers that drive higher sales.

Algorithms for market basket analysis:

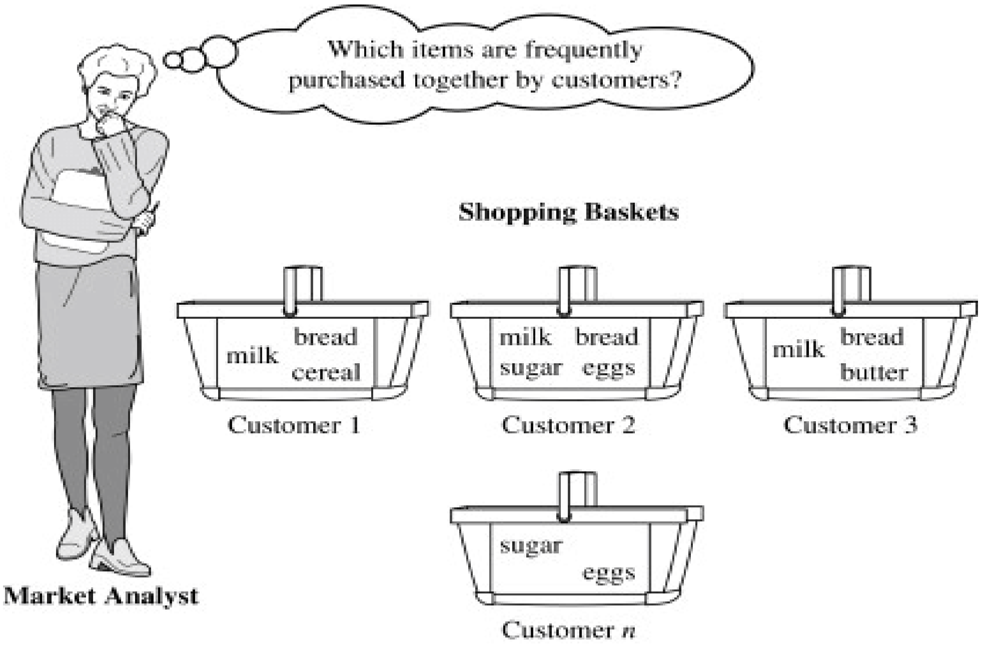
In market basket analysis, association rules are used to predict the likelihood of products being purchased together. Association rules count the frequency of items that occur together, seeking to find associations that occur far more often than expected.

Market basket analysis is a strategic data mining technique used by retailers to enhance sales by gaining a deeper understanding of customer purchasing patterns. This method entails the examination of substantial datasets, such as historical purchase records, in order to unveil inherent product groupings and identify items that tend to be bought together.

By recognizing these patterns of co-occurrence, retailers can make informed decisions to optimize inventory management, devise effective marketing strategies, employ cross-selling tactics, and even refine store layout for improved customer engagement.

For example, if customers are buying milk, how probably are they to also buy bread (and which kind of bread) on the same trip to the supermarket? This information may lead to an increase in sales by helping retailers to do selective marketing based on predictions, cross-selling, and planning their ledge space for optimal product placement.

Now, just think of the universe as the set of items available at the store, then each item has a Boolean variable that represents the presence or absence of that item. Now each basket can then be represented by a Boolean vector of values that are assigned to these variables. The Boolean vectors can be analyzed for purchase patterns that reflect items that are frequently associated or bought together. Such patterns will be represented in the form of association rules.



Source: Sciencedirect

How Does Market Basket Analysis Work?

1. Collect data on customer transactions, such as the items purchased in each transaction, the time and date of the transaction, and any other relevant information.
2. Clean and preprocess the data, removing any irrelevant information, handling missing values, and converting the data into a suitable format for analysis.

3. Use association rules mining algorithms such as Apriori or FP-Growth to identify frequent item sets, sets of items often appearing together in a transaction.
4. Calculate the support and confidence for each frequent itemset, which expresses the likelihood of one item being purchased given the purchase of another item.
5. Generate association rules based on the frequent itemsets and their corresponding support and confidence values. Association rules express the likelihood of one item being purchased given the purchase of another item.
6. Interpret the results of the market basket analysis, identifying which items are frequently purchased together, the strength of the association between items, and any other relevant insights into customer behavior and preferences.
7. Use the insights from the market basket analysis to inform business decisions such as product recommendations, store layout optimization, and targeted marketing campaigns.

3 Types of Market Basket Analysis

1. It involves identifying frequent item sets and generating association rules that express the likelihood of one item being purchased with the purchase of another item. It is used to identify the relationships or associations between items in a transactional dataset.
2. This type of market basket analysis focuses on the order in which items are purchased in a transaction. It identifies frequent item sequences and generates sequential association rules describing the likelihood of one item sequence being followed by another.
3. This type of market basket analysis involves grouping similar items or transactions into clusters or segments based on their attributes. It helps to identify customer segments with similar purchasing behaviors, which can inform product recommendations and marketing strategies.

Algorithms Used in Market Basket Analysis

There are multiple data mining techniques and algorithms used in Market Basket Analysis. One of the important objectives is “to predict the probability of items that are being bought together by customers.”

1. Apriori Algorithm
2. AIS
3. SETM Algorithm
4. FP Growth

1. Apriori Algorithm

Apriori Algorithm is a widely-used and well-known Association Rule algorithm and is a popular algorithm used in market basket analysis. It is also considered accurate and overtop

AIS and SETM algorithms. It helps to find frequent itemsets in transactions and identifies association rules between these items. The limitation of the Apriori Algorithm is frequent itemset generation. It needs to scan the database many times, leading to increased time and reduced performance as a computationally costly step because of a large dataset. It uses the concepts of Confidence and Support.

2. AIS Algorithm

The AIS algorithm creates multiple passes on the entire database or transactional data. During every pass, it scans all transactions. As you can see, in the first pass, it counts the support of separate items and determines then which of them are frequent in the database. Huge itemsets of every pass are enlarged to generate candidate itemsets. After each scanning of a transaction, the common itemsets between the itemsets of the previous pass and the items of this transaction are determined. This algorithm was the first published algorithm which is developed to generate all large itemsets in a transactional database. It focused on the enhancement of databases with the necessary performance to process decision support. This technique is bounded to only one item in the consequent.

```
AI-Dataset Loading And Preprocessing
```

Step 1: Data Loading

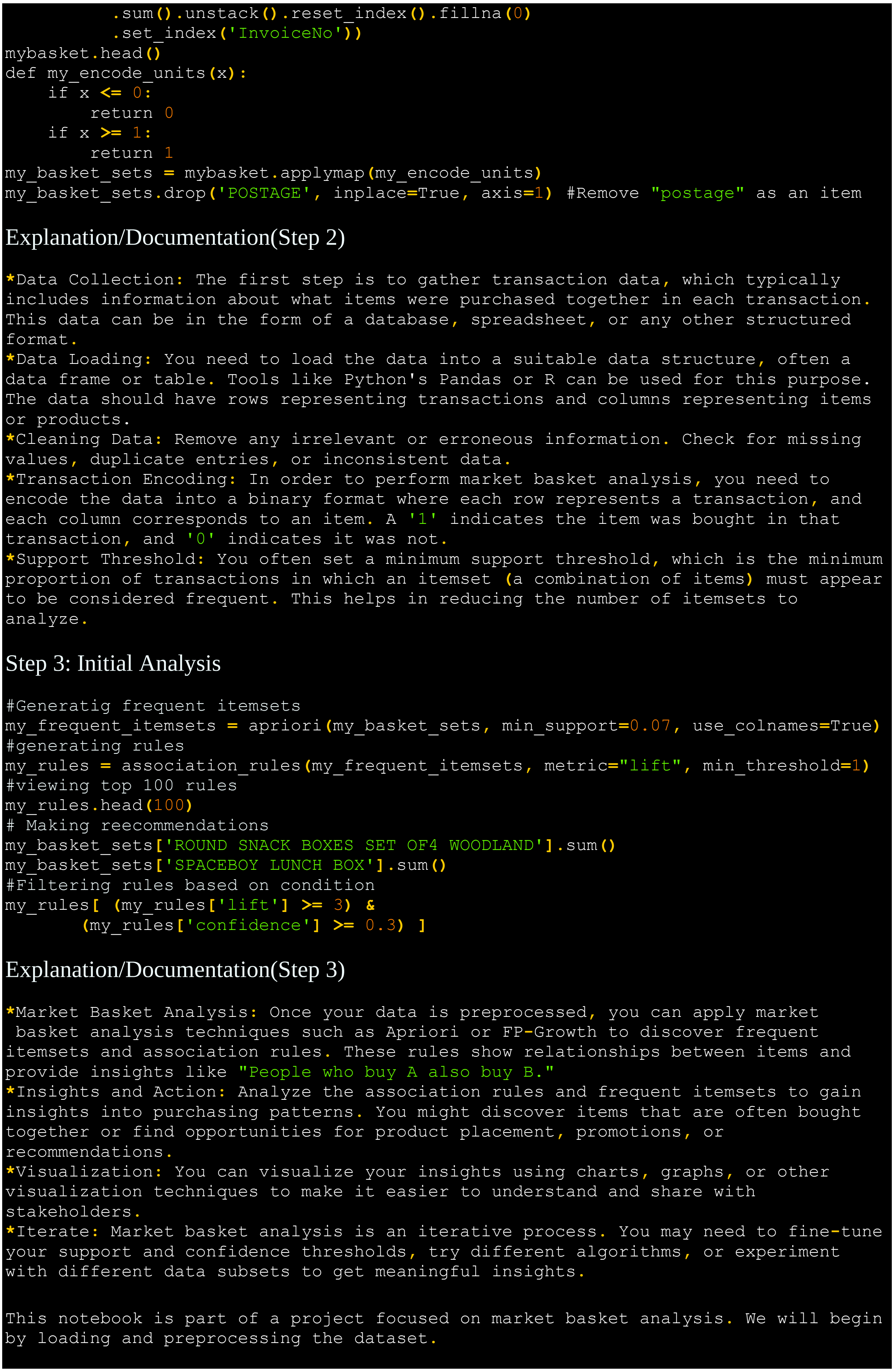
```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
#Reading Data From Web
#myretaildata = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-
databases/00352/Online%20Retail.xlsx')
myretaildata.head()
```

Explanation/Documentation(Step 1)

```
*The first step is to gather transaction data, which typically includes information
about what items were purchased together in each transaction. This data can be in
the form of a database, spreadsheet, or any other structured format.
*You need to load the data into a suitable data structure, often a data frame or
table. Tools like Python's Pandas or R can be used for this purpose. The data
should have rows representing transactions and columns representing items or
products.
```

Step 2: Data Preprocessing

```
myretaildata['Description'] = myretaildata['Description'].str.strip() #removes
spaces from beginning and end
myretaildata.dropna(axis=0, subset=['InvoiceNo'], inplace=True) #removes duplicate
invoice
myretaildata['InvoiceNo'] = myretaildata['InvoiceNo'].astype('str') #converting
invoice number to be string
myretaildata = myretaildata[~myretaildata['InvoiceNo'].str.contains('C')] #remove
the credit transactions
myretaildata.head()
myretaildata['Country'].value_counts()
#myretaildata.shape
#Separating transactions for Germany
mybasket = (myretaildata[myretaildata['Country'] =="Germany"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
```



```
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
mybasket.head()
def my_encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1
my_basket_sets = mybasket.applymap(my_encode_units)
my_basket_sets.drop('POSTAGE', inplace=True, axis=1) #Remove "postage" as an item
```

Explanation/Documentation(Step 2)

*Data Collection: The first step is to gather transaction data, which typically includes information about what items were purchased together in each transaction. This data can be in the form of a database, spreadsheet, or any other structured format.
*Data Loading: You need to load the data into a suitable data structure, often a data frame or table. Tools like Python's Pandas or R can be used for this purpose. The data should have rows representing transactions and columns representing items or products.
*Cleaning Data: Remove any irrelevant or erroneous information. Check for missing values, duplicate entries, or inconsistent data.
*Transaction Encoding: In order to perform market basket analysis, you need to encode the data into a binary format where each row represents a transaction, and each column corresponds to an item. A '1' indicates the item was bought in that transaction, and '0' indicates it was not.
*Support Threshold: You often set a minimum support threshold, which is the minimum proportion of transactions in which an itemset (a combination of items) must appear to be considered frequent. This helps in reducing the number of itemsets to analyze.

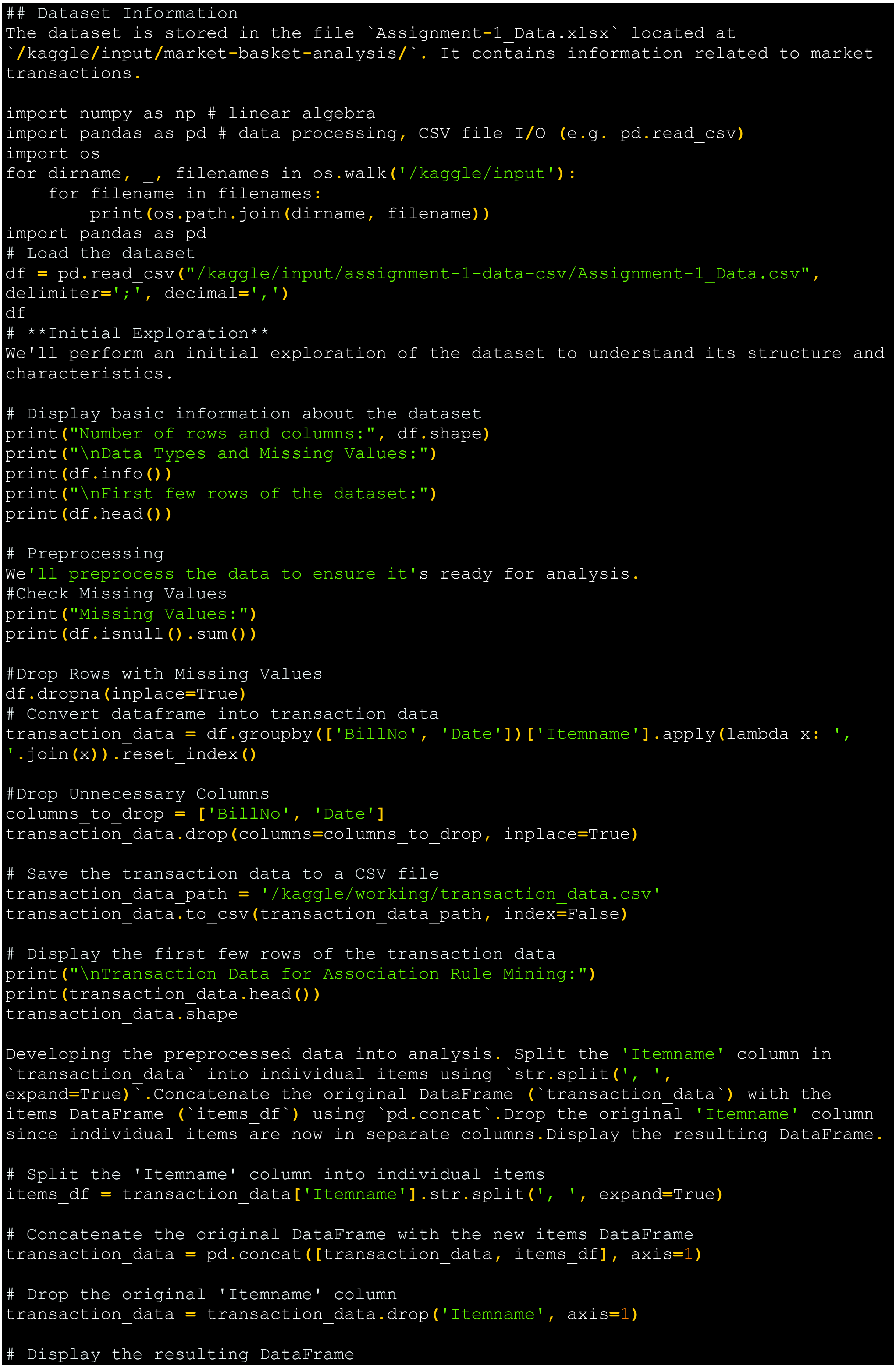
Step 3: Initial Analysis

```
#Generatig frequent itemsets
my_frequent_itemsets = apriori(my_basket_sets, min_support=0.07, use_colnames=True)
#generating rules
my_rules = association_rules(my_frequent_itemsets, metric="lift", min_threshold=1)
#viewing top 100 rules
my_rules.head(100)
# Making reecommendations
my_basket_sets['ROUND SNACK BOXES SET OF4 WOODLAND'].sum()
my_basket_sets['SPACEBOY LUNCH BOX'].sum()
#Filtering rules based on condition
my_rules[ (my_rules['lift'] >= 3) &
       (my_rules['confidence'] >= 0.3) ]
```

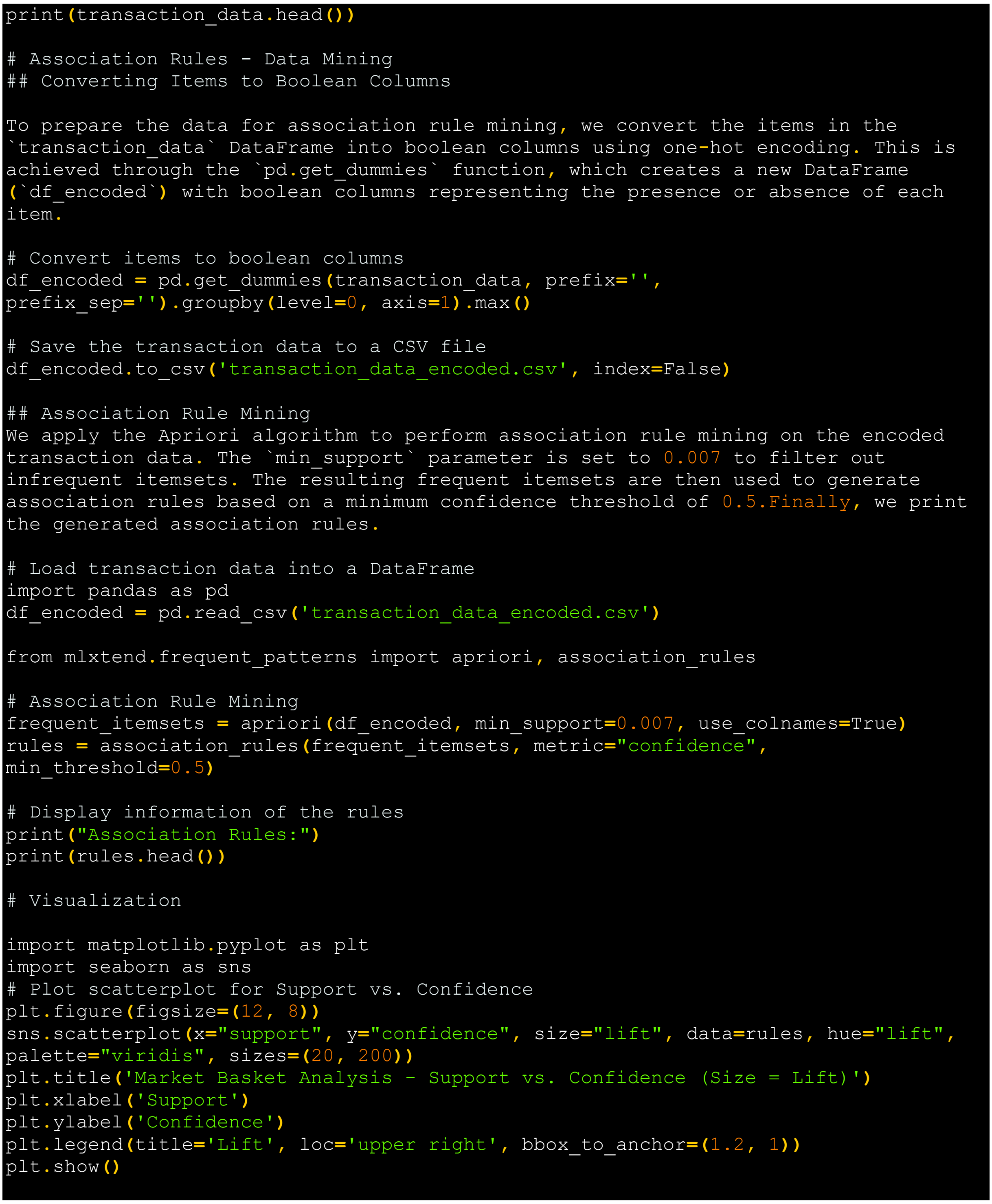
Explanation/Documentation(Step 3)

*Market Basket Analysis: Once your data is preprocessed, you can apply market basket analysis techniques such as Apriori or FP-Growth to discover frequent itemsets and association rules. These rules show relationships between items and provide insights like "People who buy A also buy B."
*Insights and Action: Analyze the association rules and frequent itemsets to gain insights into purchasing patterns. You might discover items that are often bought together or find opportunities for product placement, promotions, or recommendations.
*Visualization: You can visualize your insights using charts, graphs, or other visualization techniques to make it easier to understand and share with stakeholders.
*Iterate: Market basket analysis is an iterative process. You may need to fine-tune your support and confidence thresholds, try different algorithms, or experiment with different data subsets to get meaningful insights.

This notebook is part of a project focused on market basket analysis. We will begin by loading and preprocessing the dataset.



```
## Dataset Information
The dataset is stored in the file `Assignment-1_Data.xlsx` located at
`/kaggle/input/market-basket-analysis/`. It contains information related to market
transactions.

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import pandas as pd
# Load the dataset
df = pd.read_csv("/kaggle/input/assignment-1-data-csv/Assignment-1_Data.csv",
delimiter=';', decimal=',')
df
# **Initial Exploration**
We'll perform an initial exploration of the dataset to understand its structure and
characteristics.

# Display basic information about the dataset
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())

# Preprocessing
We'll preprocess the data to ensure it's ready for analysis.
#Check Missing Values
print("Missing Values:")
print(df.isnull().sum())

#Drop Rows with Missing Values
df.dropna(inplace=True)
# Convert dataframe into transaction data
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 = '/kaggle/working/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

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.

# 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())

# Association Rules - 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
import pandas as pd
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())

# Visualization

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()
```

Dataset:https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis

---

PRADEEPA.K **01.11.2023**

NANDHA COLLEGE OF TECHNOLOGY