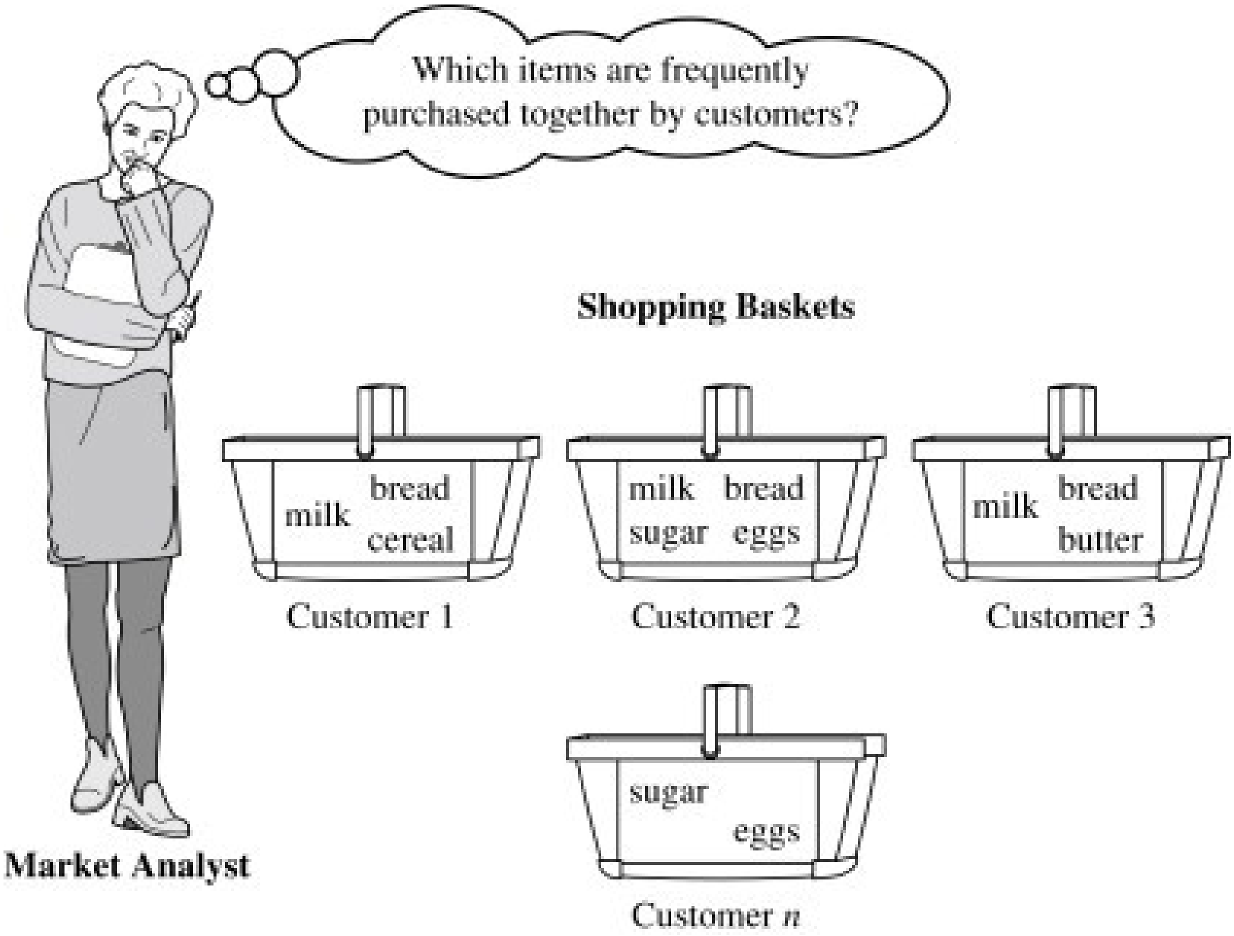
MARKET BASKET ANALYSIS

PHASE 5 SUBMISSION DOCUMENT



Problem Definition and Design Thinking

1. **Data Source:**

- Start by identifying and sourcing the dataset containing transaction data, which should include lists of purchased products. Potential data sources could include:
    - Point-of-sale systems (in-store or online)
    - E-commerce platforms
    - Customer databases
    - Loyalty program records
- Ensure that the dataset is representative of the retail business's customer transactions.

2. **Data Processing:**

    - Prepare the transaction data for association analysis by performing the following data preprocessing steps:
        - Data cleaning: Remove duplicates, handle missing values, and correct any data inconsistencies.
        - Data transformation: Convert the data into a suitable format for association analysis, such as a transaction-item matrix where rows represent transactions, and columns represent products.
        - Data encoding: Use one-hot encoding or similar techniques to convert categorical data (e.g., product names) into binary values.
        - Data aggregation: Aggregate data by customer if needed to analyze customer-level associations.

3. **Association Analysis:**

    - Utilize the Apriori algorithm or other suitable association mining techniques to identify frequent itemsets and generate association rules.
    - Set appropriate thresholds for support and confidence levels to filter meaningful associations.
    - Consider using more advanced techniques like FP-growth if dealing with large datasets for improved efficiency.

4. **Insights Generation:**

    - Interpret the association rules to understand customer purchasing behavior. This involves:

    - Identifying frequently co-purchased products (antecedents and consequents).
    - Analyzing lift and other association rule metrics to prioritize meaningful associations.
    - Identifying patterns related to product combinations, customer segments, and transaction frequency.
    - Identifying cross-selling and up-selling opportunities based on the rules.

5. **Visualization:**

    - Create visualizations to present the discovered associations and insights in an easily understandable format. Visualizations may include:
        - Scatter plots or network graphs to display item associations.
        - Heatmaps to show item co-occurrence patterns.
        - Bar charts or pie charts to represent cross-selling opportunities.
        - Customer segmentation plots to identify distinct customer groups.

6. **Business Recommendations:**

    - Provide actionable recommendations based on the insights gained from the association analysis. These recommendations may include:
        - Product bundling suggestions: Recommend product combinations that are frequently purchased together to create bundled offerings.
        - Targeted marketing strategies: Develop personalized marketing campaigns based on customer segments and their purchasing behavior.
        - Inventory management: Optimize stock levels for frequently co-purchased items.
        - Pricing strategies: Adjust pricing based on associations between products to encourage cross-selling.
        - Store layout and placement: Use insights to optimize product placement in physical stores or on e-commerce websites.

Innovation to solve the problem

Introduction:

In today's rapidly evolving business landscape, the ability to adapt and innovate is paramount. Design market analysis innovation is a strategic approach that merges the power of design thinking with comprehensive market analysis to tackle complex problems and create solutions that resonate with consumers. This innovative framework combines creative problem-solving, data-driven insights, and a deep understanding of consumer needs to not only address existing issues but also uncover new opportunities for growth and differentiation.

As businesses and industries face increasing competition and changing consumer demands, the need for a fresh perspective and novel solutions has never been more pressing. Traditional market analysis and design thinking, in isolation, have limitations. Market analysis alone may provide valuable data, but it often fails to generate transformative ideas, while design thinking, though creative, might lack the market grounding to ensure viability. The synergy of these two approaches allows organizations to bridge this gap and forge a path to sustainable success.

This integration represents a shift from merely addressing problems to embracing a proactive, forward-thinking approach that seeks to anticipate and respond to emerging challenges and opportunities. By weaving together design market analysis innovation, companies can better position themselves to deliver products and services that not only meet immediate needs but also anticipate and shape future trends. This approach promises to empower businesses to stay ahead of the curve, ultimately achieving greater market share and customer satisfaction.

1. **Problem Identification:**

- This stage involves identifying the problem or challenge in a detailed manner. Specify its scope, impact, and relevance in the market context. This is where you set the stage for the data-driven approach to problem-solving.

2. **Data Collection and Preprocessing:**
    - Data is the lifeblood of data-driven problem-solving. Collect relevant data sources, which can include customer data, market data, or any other information pertinent to the problem. Data preprocessing involves cleaning, transforming, and structuring the data for analysis. It's essential to ensure data quality and consistency.

3. **Consumer-Centric Approach:**
    - Implement design thinking to truly understand the consumers' perspectives and needs. This understanding will guide your data collection efforts, helping you gather the right information to address consumer pain points.

4. **Exploratory Data Analysis (EDA):**
    - EDA involves a deep dive into the data. It allows you to uncover trends, patterns, outliers, and correlations. This step is crucial for discovering hidden insights that can lead to innovative solutions.

5. **Feature Engineering:**
    - Feature engineering is a data science technique where you create new features or modify existing ones to make them more informative. This step can improve the performance of your predictive models.

6. **Data Integration:**
    - This phase involves combining the consumer insights from design thinking with the data from EDA and feature engineering. The integrated data provides a more holistic view, helping you identify connections and opportunities.

7. **Creative Ideation:**

    - Ideation involves brainstorming creative solutions based on the insights gained from data. It's about thinking outside the box and using data to fuel your creative process.

8. **Advanced Regression Techniques:**

    - If your problem is predictive in nature, advanced regression techniques can be employed. These models use the integrated data to make predictions and inform your solution.

9. **Prototyping and Testing:**

    - Develop prototypes or MVPs based on the predictive models. These prototypes should be tested with your target consumers to gather feedback and refine your solution

10. **Iterative Refinement:**

    - Use feedback from testing to iteratively refine the prototypes. This process should be data-driven and agile to ensure your solution aligns with consumer needs.

11. **Model Evaluation and Selection:**

    - Assess the performance of your predictive models in terms of accuracy, precision, recall, or any relevant metrics. Choose the model that best fits your problem.

12. **Model Interpretability:**

    - Ensure that the selected predictive models are interpretable. Understand the model's decision-making process and which features have the most significant influence on predictions.

13. **Deployment and Prediction:**

    - Implement the final solution based on the chosen predictive model. Use this model to make predictions or decisions related to the problem.

14. **Market Viability Analysis:**

- Evaluate the viability of your solution in the market. Consider factors like scalability, cost, and competitive advantage. Data-driven market analysis is crucial for making informed business decisions

15. **Implementation Strategy:**

- Develop a clear plan for bringing your data-driven solution to the market. Define marketing, distribution, and sales strategies that leverage your understanding of the market and consumers.

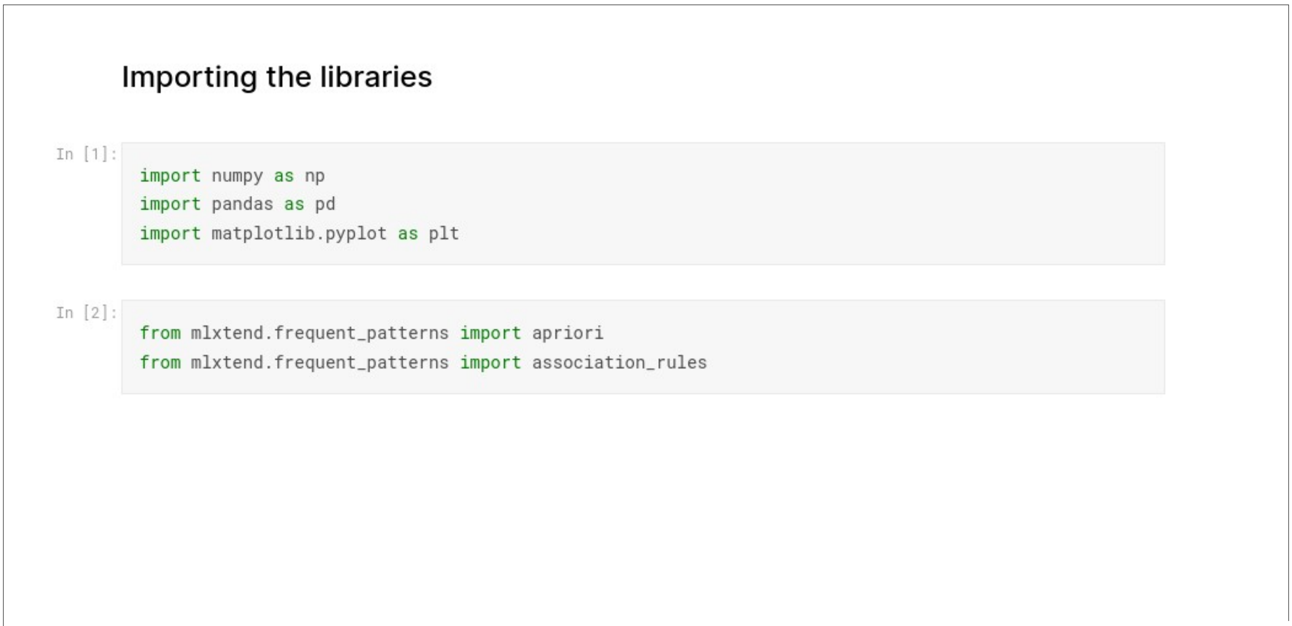
16. **Measuring Success:**

- Establish KPIs to measure the success of your solution. Monitor and analyze these metrics post-implementation to ensure that your data-driven solution is achieving the desired impact

17. **Continuous Improvement:**

- Use feedback and data to make ongoing improvements. Stay agile and adaptable to evolving market conditions, consumer preferences, and data insights. Continuously refine your solution to stay competitive.

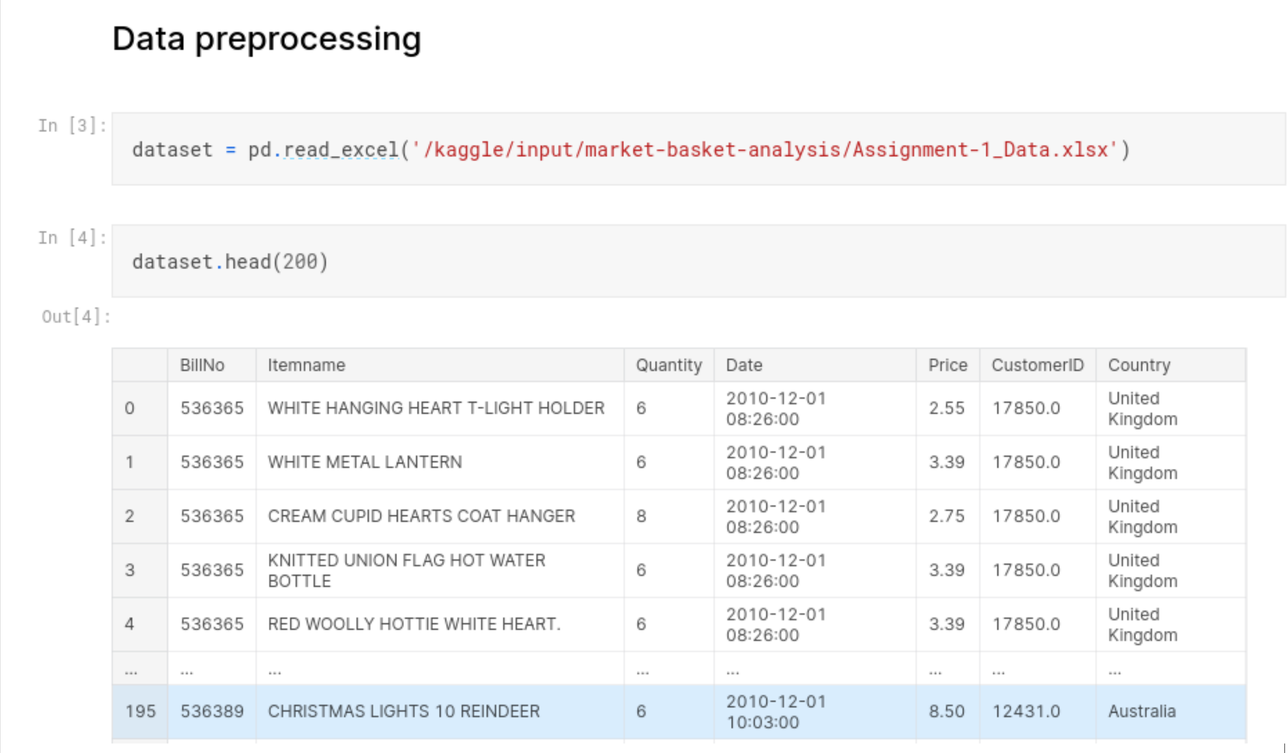
Programs :



Importing the libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```



Data preprocessing

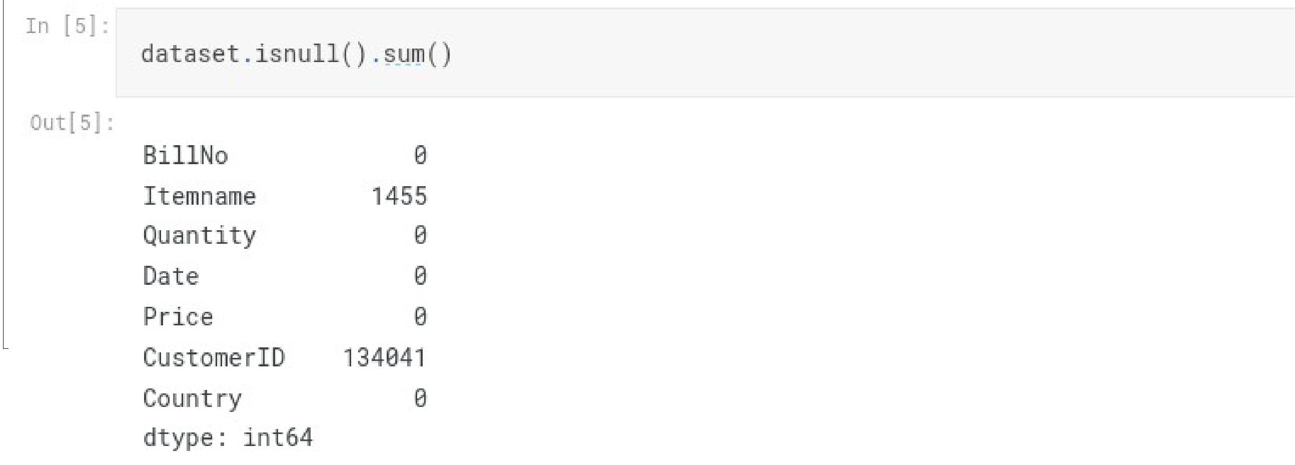
In [3]:
```
dataset = pd.read_excel('/kaggle/input/market-basket-analysis/Assignment-1_Data.xlsx')
```

In [4]:
```
dataset.head(200)
```

Out[4]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
195	536389	CHRISTMAS LIGHTS 10 REINDEER	6	2010-12-01 10:03:00	8.50	12431.0	Australia

18.Continuous Improvement:

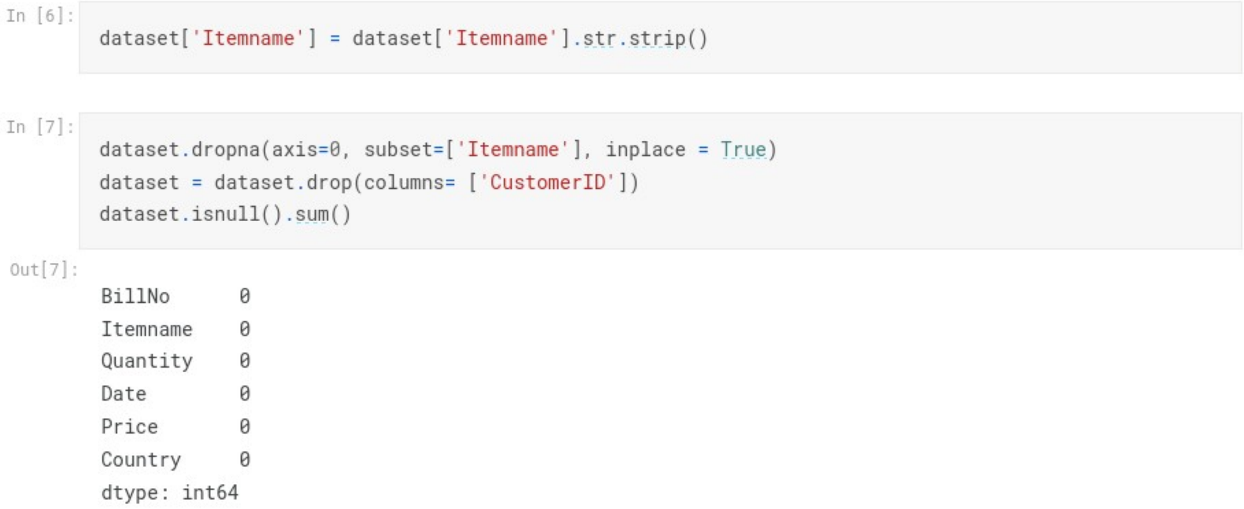


In [5]:
```
dataset.isnull().sum()
```

Out[5]:
```
BillNo             0
Itemname        1455
Quantity           0
Date               0
Price              0
CustomerID    134041
Country            0
dtype: int64
```

Conclusion:

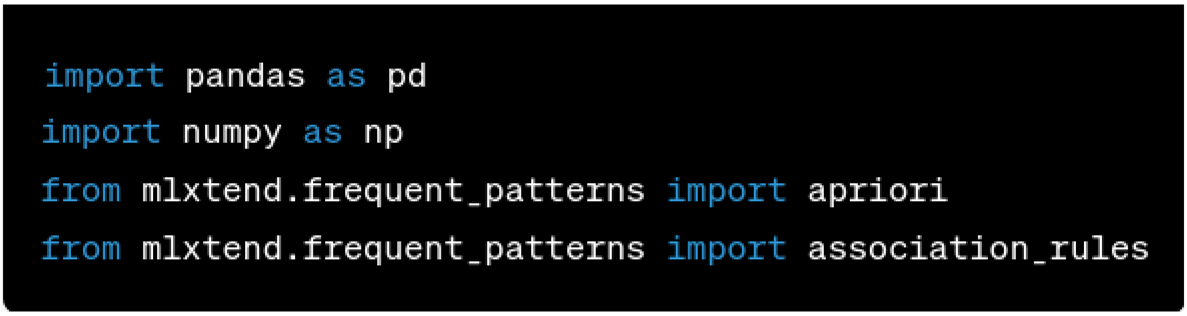
In summary, the process of "Design Market Analysis Innovation to Solve the Problem" combines design thinking, data-driven insights, and market analysis to address complex challenges. It emphasizes a consumer-centric and data-driven approach, integrates innovative ideation and predictive modeling, and focuses on continuous improvement for sustainable success.



```
In [6]:
dataset['Itemname'] = dataset['Itemname'].str.strip()

In [7]:
dataset.dropna(axis=0, subset=['Itemname'], inplace = True)
dataset = dataset.drop(columns= ['CustomerID'])
dataset.isnull().sum()

Out[7]:
BillNo       0
Itemname     0
Quantity     0
Date         0
Price        0
Country      0
dtype: int64
```

Start building the Market basket analysis model by loading and pre-processing the dataset

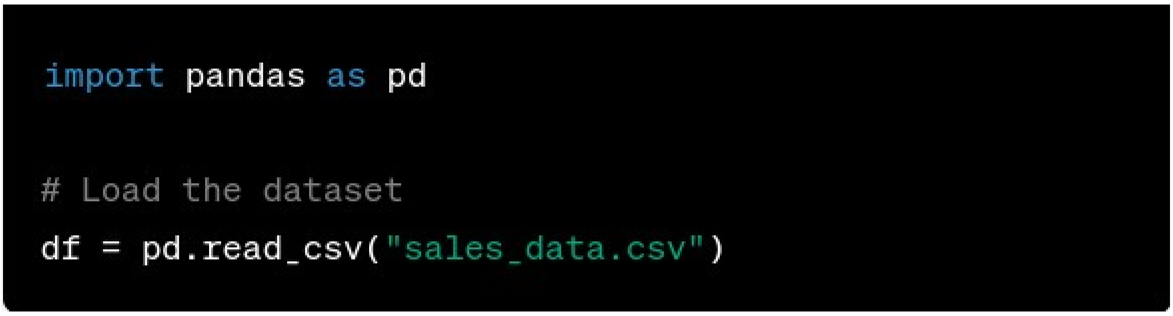
1.Import libraries :



```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

2. Load the Dataset:

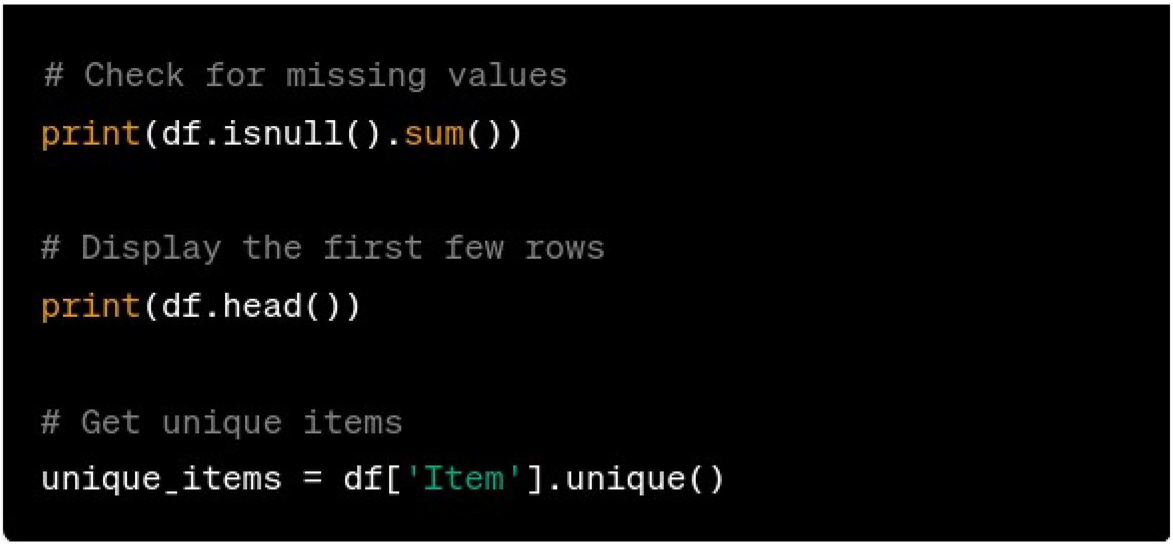
You can use various data formats like CSV, Excel, or a database. Python libraries like Pandas make it easy to read data. Here's an example to load a CSV file:



```
import pandas as pd

# Load the dataset
df = pd.read_csv("sales_data.csv")
```

3. Exploratory Data Analysis (EDA):

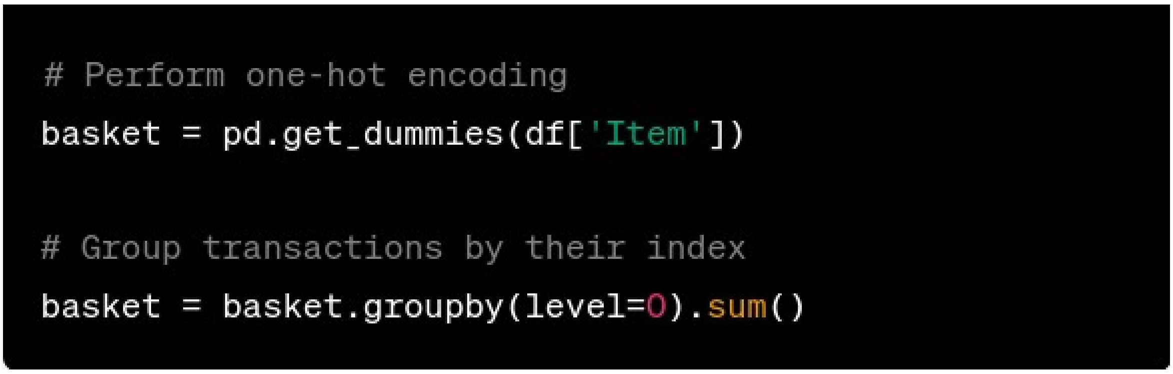
Perform initial data exploration to understand the dataset, including checking for missing values, unique items, and general statistics.



```
# Check for missing values
print(df.isnull().sum())

# Display the first few rows
print(df.head())

# Get unique items
unique_items = df['Item'].unique()
```

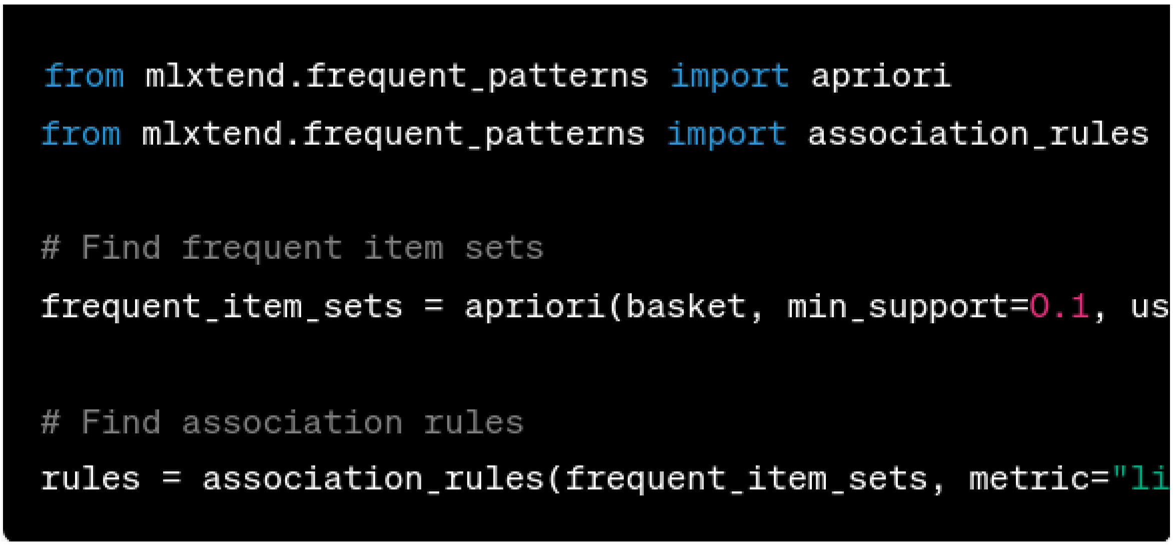
3. Data Preprocessing:

In MBA, you typically need to transform the data into a suitable format. The most common format is a one-hot encoded DataFrame, where each row represents a transaction, and each column represents an item, with binary values indicating the presence of an item in the transaction.



```
# Perform one-hot encoding
basket = pd.get_dummies(df['Item'])

# Group transactions by their index
basket = basket.groupby(level=0).sum()
```

4. Market Basket Analysis (Apriori Algorithm):



```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Find frequent item sets
frequent_item_sets = apriori(basket, min_support=0.1, us

# Find association rules
rules = association_rules(frequent_item_sets, metric="li
```

Now that your data is in the right format, you can use an MBA algorithm like Apriori to discover associations between items. You can use libraries like mlxtend in Python for this.

5. Interpret the Results:

Analyze the generated association rules, considering metrics like support, confidence, and lift. These rules represent item associations and can be used for business recommendations.

```
# Display association rules
print(rules)
```

Importance of loading and processing dataset :

Loading and processing a dataset is a crucial initial step in any data analysis or machine learning task, including Market Basket Analysis. Here are the key reasons why loading and processing a dataset is important:

1. **Data Availability:** Without loading the dataset, you cannot access or analyze the data. It's the first step in making the data available for your analysis.
2. **Data Understanding:** By loading the data, you gain an initial understanding of its structure, size, and format. You can see the raw information and start to identify any potential issues or anomalies.
3. **Data Cleaning:** Datasets are rarely perfect. Loading the data allows you to identify and address missing values, duplicates, inconsistent formatting, and other data quality issues. Clean data is essential for accurate analysis.
4. **Data Transformation:** Depending on the analysis you want to perform, you may need to transform the data. For Market Basket Analysis, this often includes converting transactional data into a suitable format, like one-hot encoding, where items are represented as binary variables indicating presence or absence.
5. **Data Exploration (EDA):** Once the data is loaded, you can perform exploratory data analysis to gain insights. EDA involves examining summary statistics, visualizing the data, and identifying trends, patterns, and outliers. This helps you make informed decisions about the analysis approach.
6. **Preparation for Analysis:** Data processing is critical for preparing the data to work with specific analysis techniques or machine

learning algorithms. For Market Basket Analysis, this could involve creating transaction-item matrices, which serve as the foundation for finding associations.

7. **Efficiency:** Processing the data, such as optimizing data structures and cleaning, ensures that your analysis runs efficiently. It can save computational resources and time during the analysis phase.

8. **Accuracy and Reliability:** Properly processing the data helps ensure that the results of your analysis are accurate and reliable. By addressing issues like missing values and duplicates, you reduce the risk of drawing incorrect conclusions.

9. **Data Privacy and Compliance:** Processing data can also involve handling sensitive information appropriately, ensuring that you comply with privacy regulations and protect personal data.

10. **Customization:** Depending on the specific requirements of your analysis, you may need to customize the data processing steps. This customization can address unique data challenges and improve the quality of the analysis.

Challenges involved in loading and preprocessing a market basket dataset :

Loading and preprocessing a house price dataset can be a complex task, as it involves handling various types of data and dealing with several challenges. Here are some of the common challenges involved in this process:

1. **Data Quality:** Datasets may contain missing values, outliers, errors, or inconsistencies. Cleaning and imputing these issues is a crucial step in preprocessing.

2. **Data Size:** Large datasets can be memory-intensive and slow to process. You may need to consider techniques for managing and working with big data.

3. **Data Types:** House price datasets often include a mix of data types, such as numerical (e.g., price, square footage) and

categorical (e.g., location, property type). Handling these different data types requires distinct preprocessing steps.

4. **Feature Engineering:** Extracting meaningful features from raw data is essential. This may involve creating new variables, aggregating data, or transforming existing features to improve the model's performance.
5. **Normalization and Scaling:** To ensure that numerical features have the same scale, preprocessing may involve normalization or standardization. This can be important for machine learning models that are sensitive to feature scales.
6. **Categorical Data Handling:** Categorical variables must be encoded (e.g., one-hot encoding) so that they can be used in machine learning algorithms. Handling a large number of categories can be challenging.
7. **Dealing with Outliers:** Outliers can significantly impact the accuracy of house price predictions. Deciding how to handle outliers, whether through removal or transformation, is a critical step.
8. **Dimensionality Reduction:** Some datasets may have high dimensionality, with many features. Dimensionality reduction techniques like Principal Component Analysis (PCA) may be needed.
9. **Data Imbalance:** If the dataset has an imbalanced distribution of target values (e.g., very few luxury properties compared to standard ones), the model may have difficulty learning patterns. Techniques like oversampling, undersampling, or using appropriate evaluation metrics are needed.
10. **Handling Time-Series Data:** Some datasets may include time-series data, such as historical price trends. Special techniques for time-series analysis may be required.
11. **Geospatial Data:** Datasets that include geographical information (e.g., latitude, longitude) may require specialized preprocessing to account for spatial relationships.

12. **Handling Missing Data:** Deciding how to deal with missing data is important. Imputation techniques, such as mean, median, or machine learning-based imputation, need to be chosen.
13. **Data Normalization and Encoding:** Ensuring that data is in a suitable format for machine learning models is crucial. This includes encoding text data, normalizing numeric data, and splitting the dataset into training and testing sets.
14. **Data Privacy and Security:** Handling sensitive information like property addresses and owner details requires safeguarding privacy and complying with data protection regulations.
15. **Data Splitting:** Properly splitting the dataset into training, validation, and test sets is essential for model evaluation and avoiding data leakage.
16. **Feature Selection:** Identifying the most relevant features and eliminating irrelevant ones can improve model performance and reduce complexity.

How to overcome the challenges of loading and preprocessing a house price dataset :

Overcoming the challenges of loading and preprocessing a house price dataset involves a systematic approach and a combination of data preprocessing techniques. Here are some strategies to address these challenges:

1. **Data Quality Issues:**
    - Handle missing data by imputing it with appropriate values, such as the mean, median, or using more advanced imputation methods like K-Nearest Neighbors (KNN) imputation.
    - Identify and handle outliers using methods like Z-scores, interquartile range (IQR), or robust statistical methods.
    - Correct errors and inconsistencies in the data, such as typos or incorrect entries.

2. **Data Size:**
    - If the dataset is large and memory-intensive, consider using data sampling to work with a subset for exploratory analysis and model building.
    - Use distributed computing frameworks like Apache Spark for big data processing.
3. **Data Types:**
    - Categorize and differentiate between numerical and categorical features, as they require different preprocessing steps.
    - Convert categorical data into a numerical format using techniques like one-hot encoding or label encoding.
4. **Feature Engineering:**
    - Create new features that might be more informative for predicting house prices, such as square footage per bedroom, age of the property, or proximity to amenities.
    - Aggregate or transform existing features to create meaningful variables.
5. **Normalization and Scaling:**
    - Normalize or standardize numerical features to ensure they have a similar scale, especially when using machine learning algorithms that are sensitive to feature scales.
6. **Categorical Data Handling:**
    - Apply one-hot encoding to convert categorical variables into binary (0/1) format.
    - Consider feature selection techniques to reduce dimensionality if one-hot encoding leads to a large number of new binary columns.
7. **Dealing with Outliers:**
    - Decide on a strategy for handling outliers, such as truncation (capping), transformation, or treating them as a separate category.
8. **Dimensionality Reduction:**

- Use dimensionality reduction techniques like Principal Component Analysis (PCA) to reduce the number of features and avoid multicollinearity.

9. **Data Imbalance:**
   - Implement techniques like oversampling or undersampling for imbalanced datasets to ensure the model can learn from minority classes.
   - Choose appropriate evaluation metrics that account for imbalanced data, such as area under the Receiver Operating Characteristic curve (AUC-ROC) or F1-score.
10. **Handling Time-Series Data:**
    - Utilize time-series analysis techniques if your dataset involves time-dependent variables, such as historical price trends.
11. **Geospatial Data:**
    - Use geospatial analysis methods to extract insights from geographical features like latitude, longitude, and distances to certain locations.
12. **Handling Missing Data:**
    - Choose an imputation method that is suitable for the nature of missing data (e.g., mean for numerical, mode for categorical) and document the imputation process.
13. **Data Normalization and Encoding:**
    - Standardize data formats, ensuring that text data is encoded, numeric data is normalized, and date-time data is formatted consistently.
14. **Data Privacy and Security:**
    - Safeguard sensitive information and adhere to data protection regulations. Consider anonymization techniques when working with personal data.
15. **Data Splitting:**
    - Properly split the dataset into training, validation, and test sets to avoid data leakage and ensure model evaluation is representative of real-world performance.

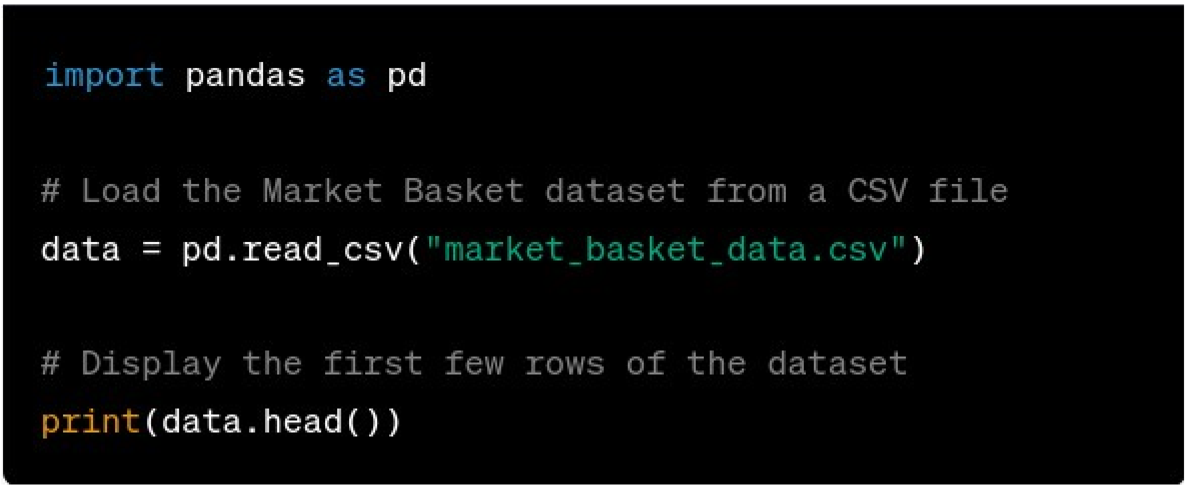
16.**Feature Selection:**

- Employ feature selection methods, such as Recursive Feature Elimination (RFE) or feature importance from tree-based models, to reduce the number of features to the most relevant ones.

loading dataset :

Loading a dataset for Market Basket Analysis typically involves reading transactional data from a file, such as a CSV file, into a format suitable for analysis. Below is a Python code example using the Pandas library to load a sample Market Basket dataset from a CSV file

Assuming you have a CSV file named "market_basket_data.csv" with transaction data, here's how you can load it:



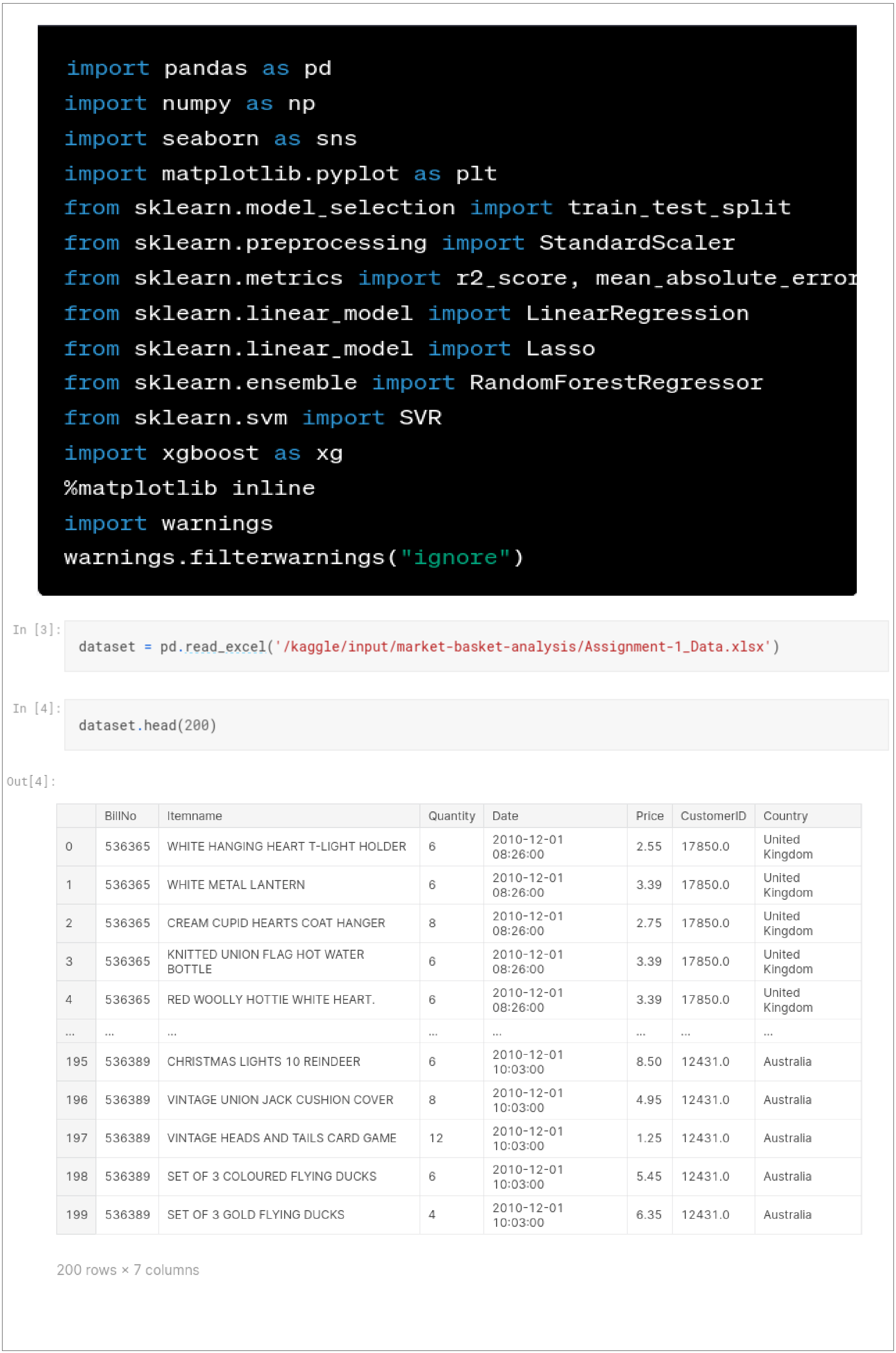
```python
import pandas as pd

# Load the Market Basket dataset from a CSV file
data = pd.read_csv("market_basket_data.csv")

# Display the first few rows of the dataset
print(data.head())
```

Make sure to replace "market_basket_data.csv" with the actual filename of your dataset. This code uses Pandas to read the data into a DataFrame, which is a common data structure used for data analysis in Python.

The dataset should have each row representing a transaction, and the columns should represent items or products. For Market Basket Analysis, this is the typical format.

Once you've loaded the dataset, you can further preprocess it and perform Market Basket Analysis to discover associations between items that are frequently purchased together. This may include converting the data into the required format, removing any unnecessary columns, and using specialized libraries like mlxtend for frequent itemset mining and association rule generation.

Program :



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xg
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
dataset = pd.read_excel('/kaggle/input/market-basket-analysis/Assignment-1_Data.xlsx')
```

In [4]:

```
dataset.head(200)
```

Out[4]:

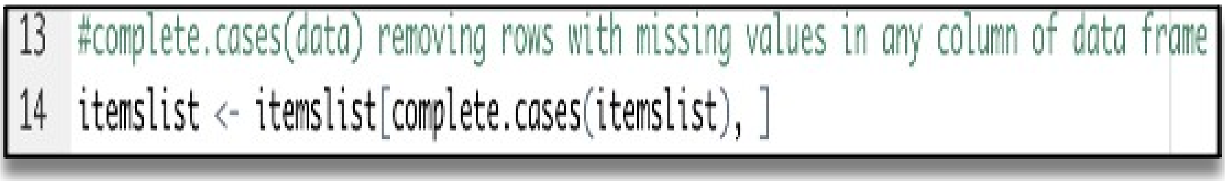
	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
195	536389	CHRISTMAS LIGHTS 10 REINDEER	6	2010-12-01 10:03:00	8.50	12431.0	Australia
196	536389	VINTAGE UNION JACK CUSHION COVER	8	2010-12-01 10:03:00	4.95	12431.0	Australia
197	536389	VINTAGE HEADS AND TAILS CARD GAME	12	2010-12-01 10:03:00	1.25	12431.0	Australia
198	536389	SET OF 3 COLOURED FLYING DUCKS	6	2010-12-01 10:03:00	5.45	12431.0	Australia
199	536389	SET OF 3 GOLD FLYING DUCKS	4	2010-12-01 10:03:00	6.35	12431.0	Australia

200 rows × 7 columns

Preprocessing the dataset:

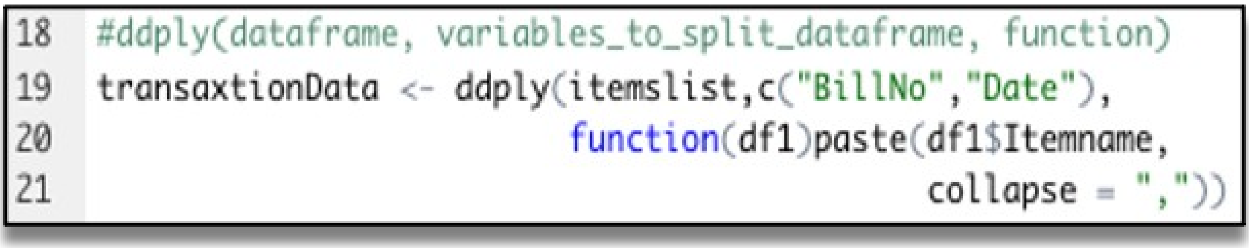
Preprocessing in the context of dataset refers to a set of operations and techniques applied to raw data before it is used for analysis, machine learning, or any other data-related task. The goal of preprocessing is to clean, transform, and prepare the data so that it is suitable for the specific task at hand

After we will clear our data frame, will remove missing values.



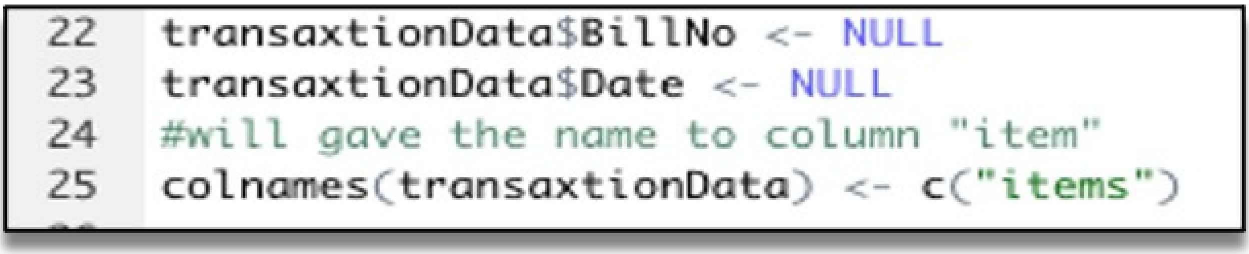
```
#complete.cases(data) removing rows with missing values in any column of data frame
itemslist <- itemslist[complete.cases(itemslist), ]
```

The summary gives us some useful information:To apply Association Rule mining, we need to convert dataframe into transaction data to make all items that are bought together in one invoice will be in one row. Below lines of code will combine all products from one BillNo and Date and combine all products from that BillNo and Dat



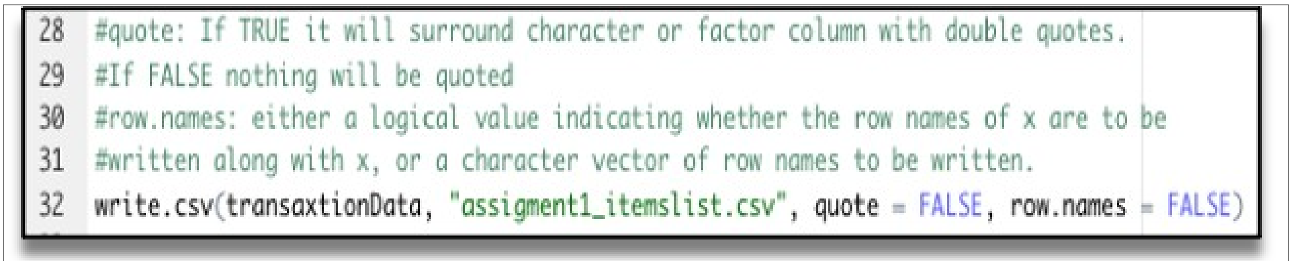
```
#ddply(dataframe, variables_to_split_dataframe, function)
transaxtionData <- ddply(itemslist,c("BillNo","Date"),
                         function(df1)paste(df1$Itemname,
                                            collapse = ","))
```

We don’t need BillNo and Date, we will make it as Null.
Next, you have to store this transaction data into .csv

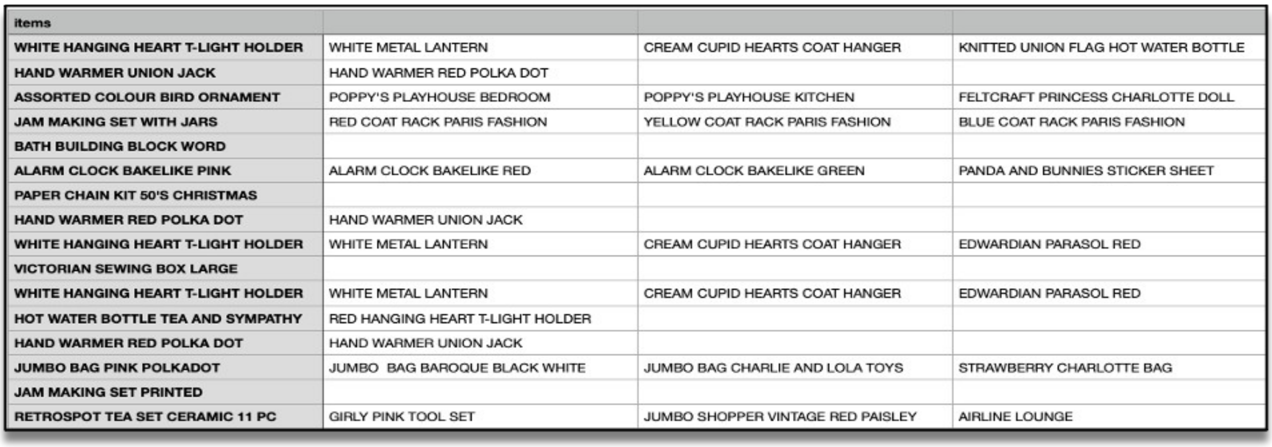


```
transaxtionData$BillNo <- NULL
transaxtionData$Date <- NULL
#will gave the name to column "item"
colnames(transaxtionData) <- c("items")
```

This how should look transaction data before we will go to next step.



```
#quote: If TRUE it will surround character or factor column with double quotes.
#If FALSE nothing will be quoted
#row.names: either a logical value indicating whether the row names of x are to be
#written along with x, or a character vector of row names to be written.
write.csv(transaxtionData, "assigment1_itemslist.csv", quote = FALSE, row.names = FALSE)
```



items			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	KNITTED UNION FLAG HOT WATER BOTTLE
HAND WARMER UNION JACK	HAND WARMER RED POLKA DOT		
ASSORTED COLOUR BIRD ORNAMENT	POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN	FELTCRAFT PRINCESS CHARLOTTE DOLL
JAM MAKING SET WITH JARS	RED COAT RACK PARIS FASHION	YELLOW COAT RACK PARIS FASHION	BLUE COAT RACK PARIS FASHION
BATH BUILDING BLOCK WORD			
ALARM CLOCK BAKELIKE PINK	ALARM CLOCK BAKELIKE RED	ALARM CLOCK BAKELIKE GREEN	PANDA AND BUNNIES STICKER SHEET
PAPER CHAIN KIT 50'S CHRISTMAS			
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
VICTORIAN SEWING BOX LARGE			
WHITE HANGING HEART T-LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	EDWARDIAN PARASOL RED
HOT WATER BOTTLE TEA AND SYMPATHY	RED HANGING HEART T-LIGHT HOLDER		
HAND WARMER RED POLKA DOT	HAND WARMER UNION JACK		
JUMBO BAG PINK POLKADOT	JUMBO BAG BAROQUE BLACK WHITE	JUMBO BAG CHARLIE AND LOLA TOYS	STRAWBERRY CHARLOTTE BAG
JAM MAKING SET PRINTED			
RETROSPOT TEA SET CERAMIC 11 PC	GIRLY PINK TOOL SET	JUMBO SHOPPER VINTAGE RED PAISLEY	AIRLINE LOUNGE

Conclusion :

In conclusion, loading and preprocessing the dataset for Market Basket Analysis is a fundamental and essential step in uncovering valuable insights into customer buying behavior and optimizing sales and marketing strategies. Key takeaways specific to Market Basket Analysis include:

1. **Data Loading:** Importing transactional data into an accessible data structure, such as a Pandas DataFrame in Python, is the first step in the analysis.
2. **Data Exploration:** An initial examination of the dataset helps you understand its structure, identify missing values, and inspect the raw information.
3. **Data Transformation:** Conversion of the data into a suitable format, where transactions become rows, and items become columns with binary values (1 for presence, 0 for absence) is essential. This is typically achieved through one-hot encoding.
4. **Data Summary:** Calculating summary statistics, such as the number of transactions and the average number of items per transaction, provides insights into the dataset's characteristics.

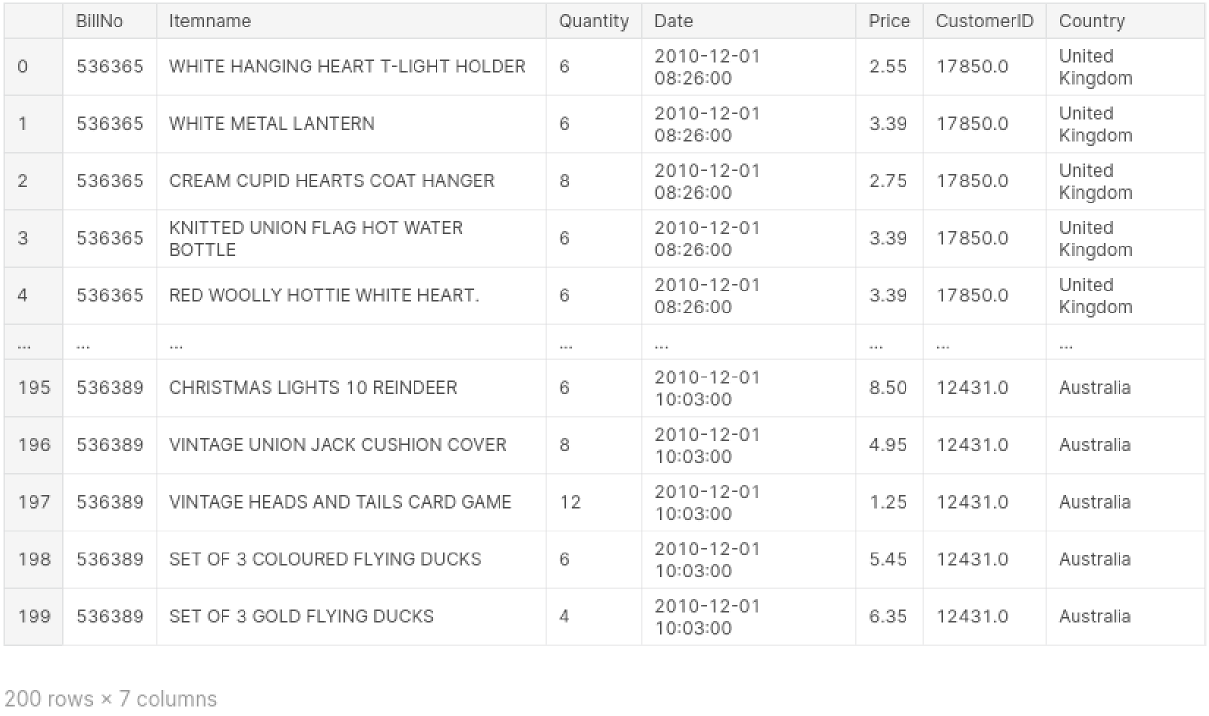
5. **Remove Rare Items:** Consider excluding items that appear infrequently in transactions to improve the relevance of association rules.
6. **Handling Missing Values:** Address any missing values in the dataset, typically through imputation or data cleaning techniques.
7. **Save the Processed Data:** Optionally, you can save the preprocessed data for future use, streamlining Market Basket Analysis in the future.

Through these preprocessing steps, you prepare your dataset for frequent itemset mining and association rule generation. This well-structured and clean data serves as the foundation for discovering item associations and optimizing business strategies in retail and e-commerce.

Market Basket Analysis (MBA), also known as Association Rule Mining or Affinity Analysis, is a data mining and analytical technique used in retail and e-commerce to discover patterns, relationships, and associations among products that customers frequently purchase together.

The primary goal of MBA is to uncover insights into customer buying behavior, enhance marketing strategies, and optimize product placement in stores.

<u>Given detaset :</u>



	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
195	536389	CHRISTMAS LIGHTS 10 REINDEER	6	2010-12-01 10:03:00	8.50	12431.0	Australia
196	536389	VINTAGE UNION JACK CUSHION COVER	8	2010-12-01 10:03:00	4.95	12431.0	Australia
197	536389	VINTAGE HEADS AND TAILS CARD GAME	12	2010-12-01 10:03:00	1.25	12431.0	Australia
198	536389	SET OF 3 COLOURED FLYING DUCKS	6	2010-12-01 10:03:00	5.45	12431.0	Australia
199	536389	SET OF 3 GOLD FLYING DUCKS	4	2010-12-01 10:03:00	6.35	12431.0	Australia

200 rows × 7 columns

Feature Engineering:

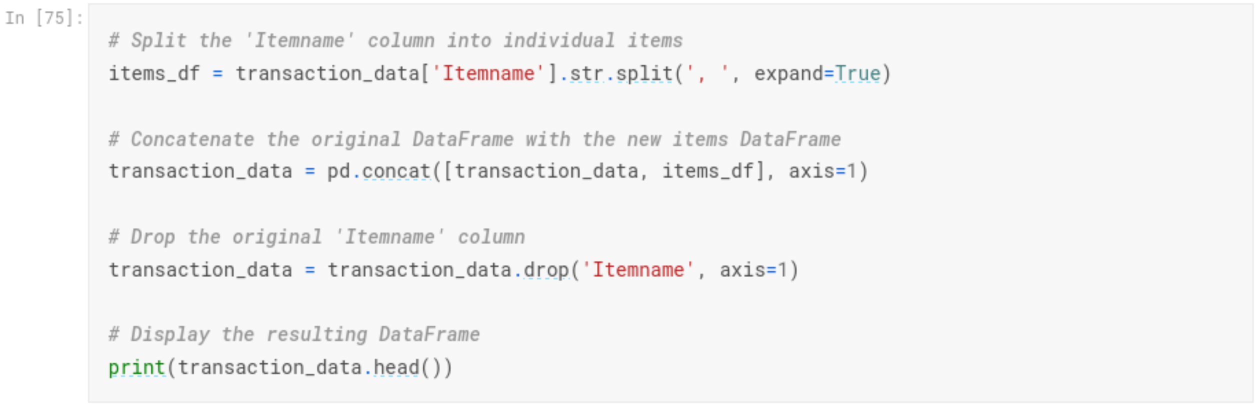
In a Market Basket Analysis project, feature engineering is mainly about transforming your data into a suitable format for association rule mining. We've already done some of this during the preprocessing step. However, you might consider adding features like the number of items in a transaction, the total amount spent, or the time of the transaction if applicable to your dataset.

Feature Selection:

Identify the most relevant features (product attributes) for your analysis. You may consider factors like product category, price, brand, and customer demographics.

Creating Association Rules:

Implement the chosen association rule mining algorithm (e.g., Apriori or FP-growth) to generate frequent itemsets and association rules.



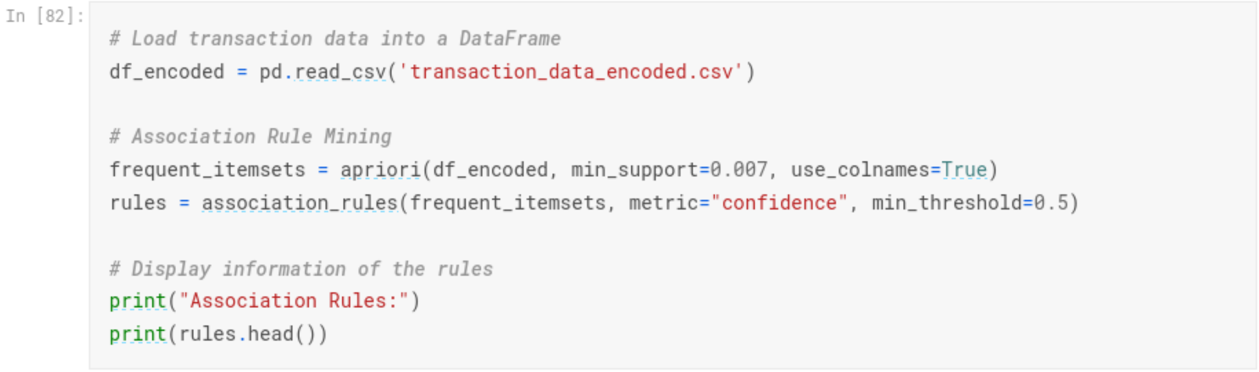
```
In [75]:  # Split the 'Itemname' column into individual items
          items_df = transaction_data['Itemname'].str.split(', ', expand=True)

          # 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:

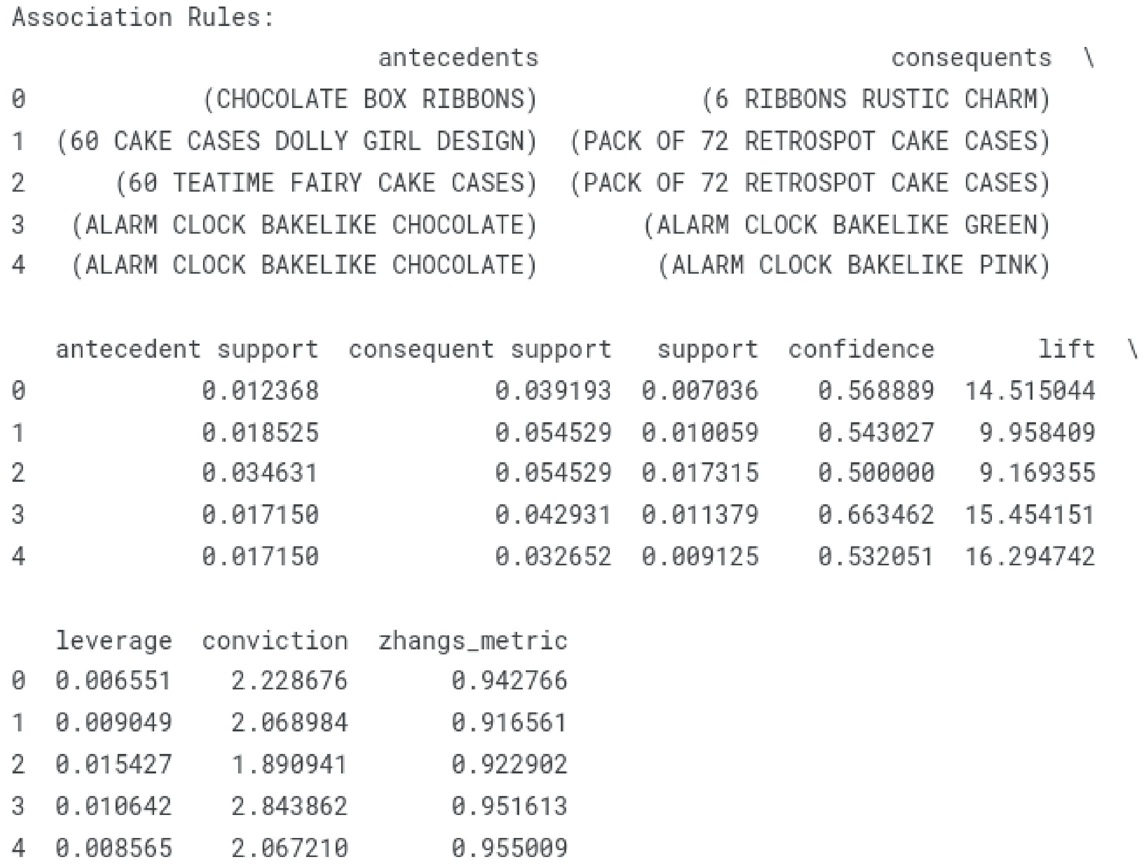
Once you've identified frequent itemsets using Apriori, you can generate association rules. This code will generate rules with a minimum confidence of 0.5 (adjust as needed):

rules = association_rules(frequent_itemsets, metric='lift', min_threshold=1.0)

The 'lift' metric measures the strength of association between items. You can use other metrics like 'confidence' or 'support' depending on your needs.



```
In [82]:  # Load transaction data into a DataFrame
          df_encoded = pd.read_csv('transaction_data_encoded.csv')

          # 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
```

Model Training - Apriori Algorithm:

- **Algorithm Selection**: Discuss the choice of the association rule mining algorithm. Explain why you selected a particular algorithm and how it aligns with the project goals.
- **Data Split**: Split the dataset into training and testing sets (if applicable). Describe your approach to data partitioning.
- **Model Training**: Train the association rule mining model on the training data.

The Apriori algorithm is commonly used for association rule mining. You can use the mlxtend library to apply this algorithm:

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Perform market basket analysis using Apriori
frequent_itemsets = apriori(basket.drop('TransactionID', axis=1),
min_support=0.01, use_colnames=True)
```

In this code, min_support specifies the minimum support threshold for itemsets. Adjust this value according to your dataset and analysis requirements.

Model Evaluation:

To evaluate the quality of the association rules, you can consider various metrics like lift, confidence, and support. You might also filter the rules based on these metrics to retain only those that are most interesting or relevant

Performance Metrics: Report the model's performance based on the selected metrics. Present numerical results and provide interpretations. Discuss the implications of high or low values for support, confidence, and lift.
Quality of Rules: Evaluate the quality and relevance of the generated association rules. Discuss the significance of the rules, such as identifying strong associations or discovering unexpected patterns. Consider the potential for actionable insights.

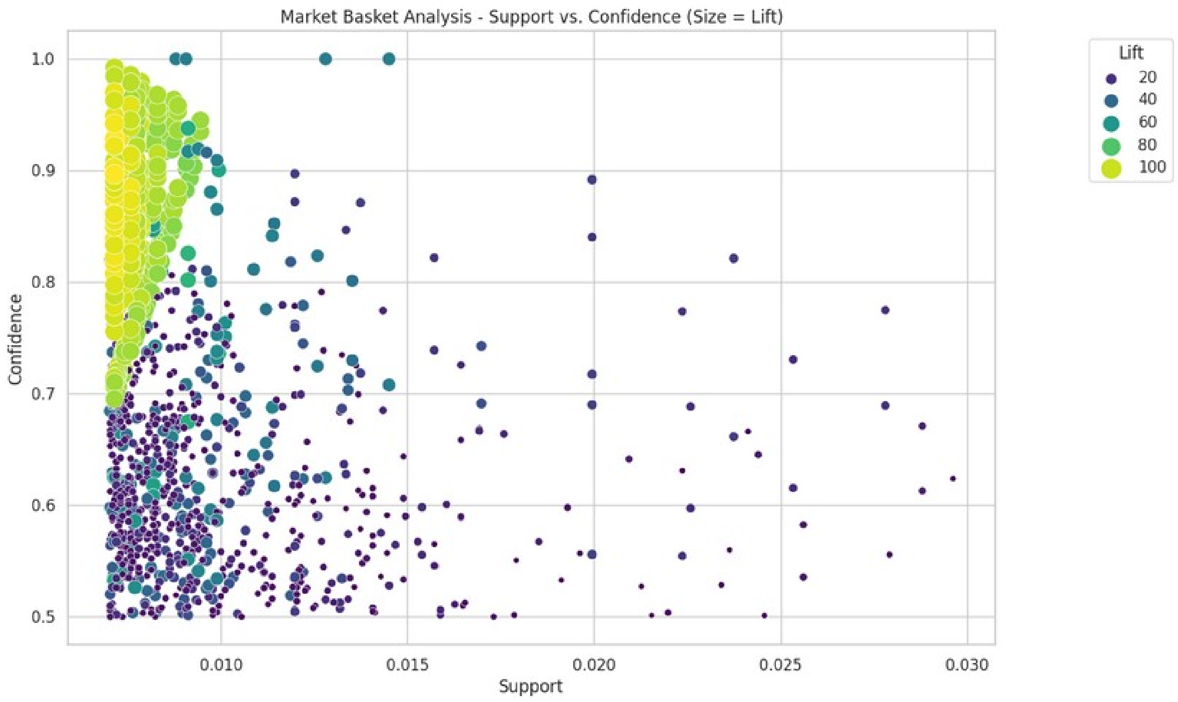
Visualizations: If appropriate, create visualizations to support your findings. Examples include bar charts showing rule metrics, network graphs illustrating item associations, or heatmaps depicting rule relationships.

Visualization:

Create visualizations to represent the discovered association rules. For instance, you can visualize support vs. confidence or lift vs. support to identify significant rules.

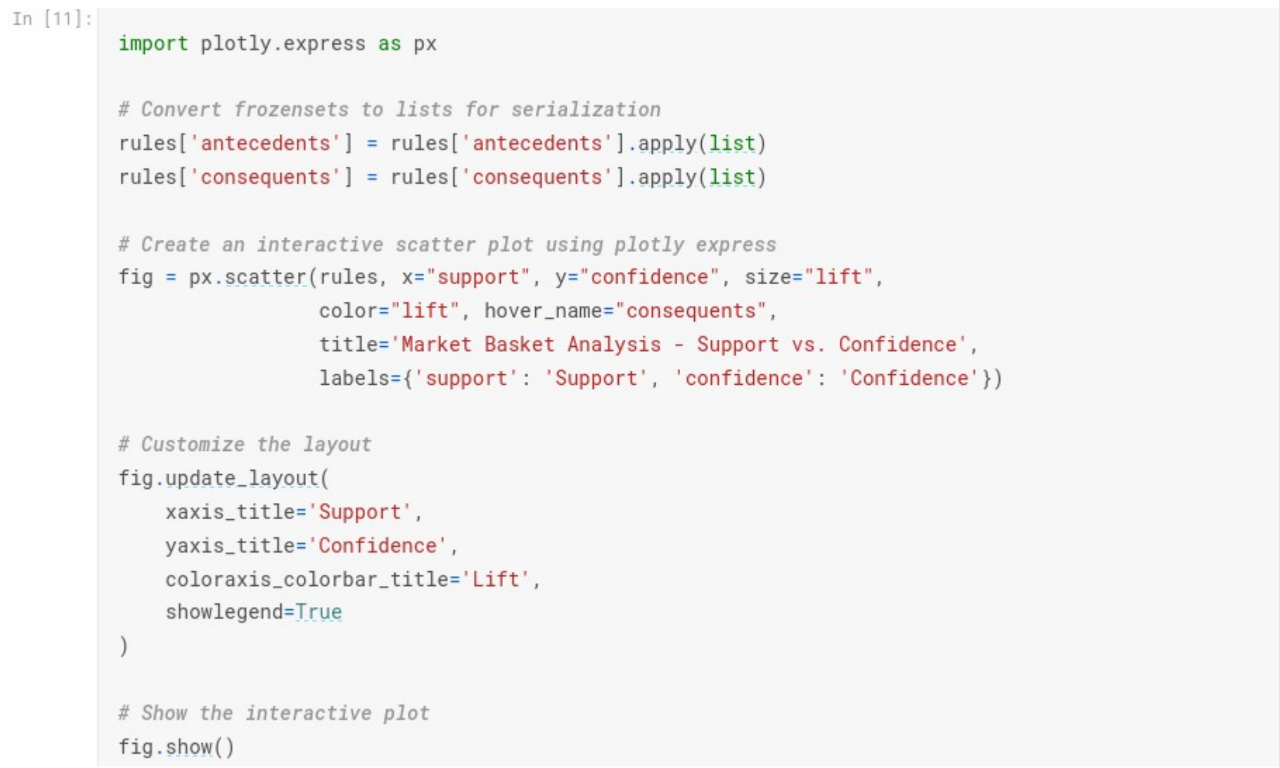
In [78]:

```
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="viridi
s", 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()
```

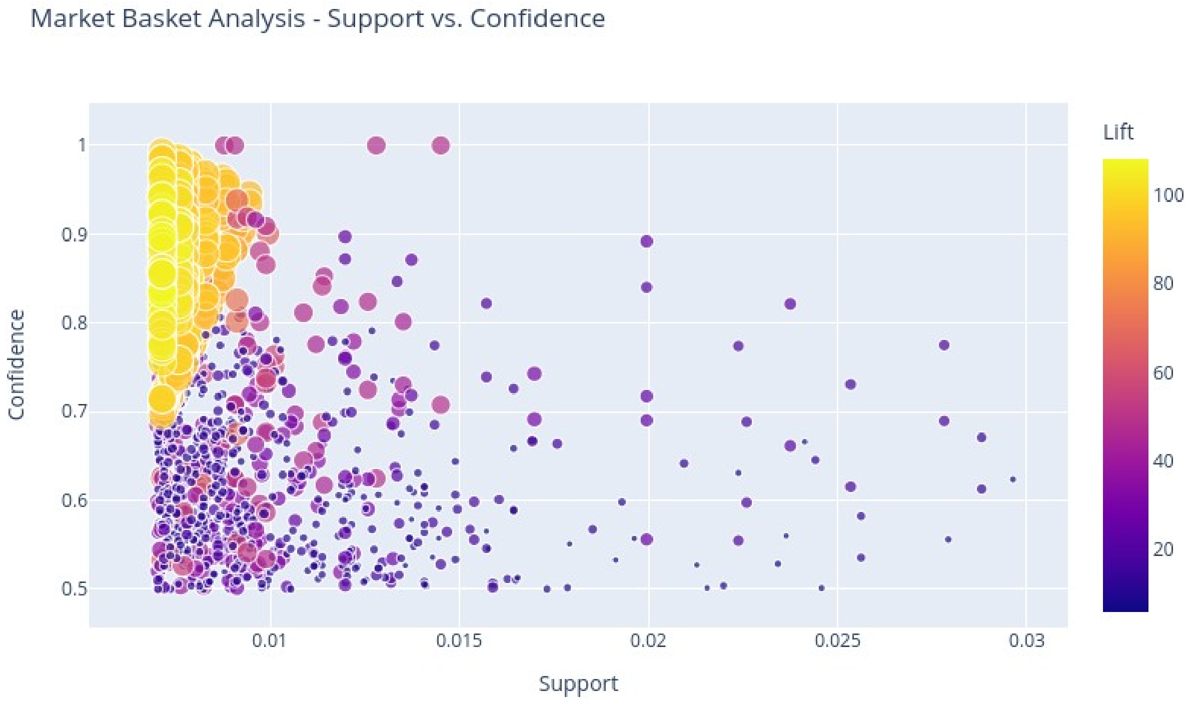


Interactive Market Basket Analysis Visualization

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.

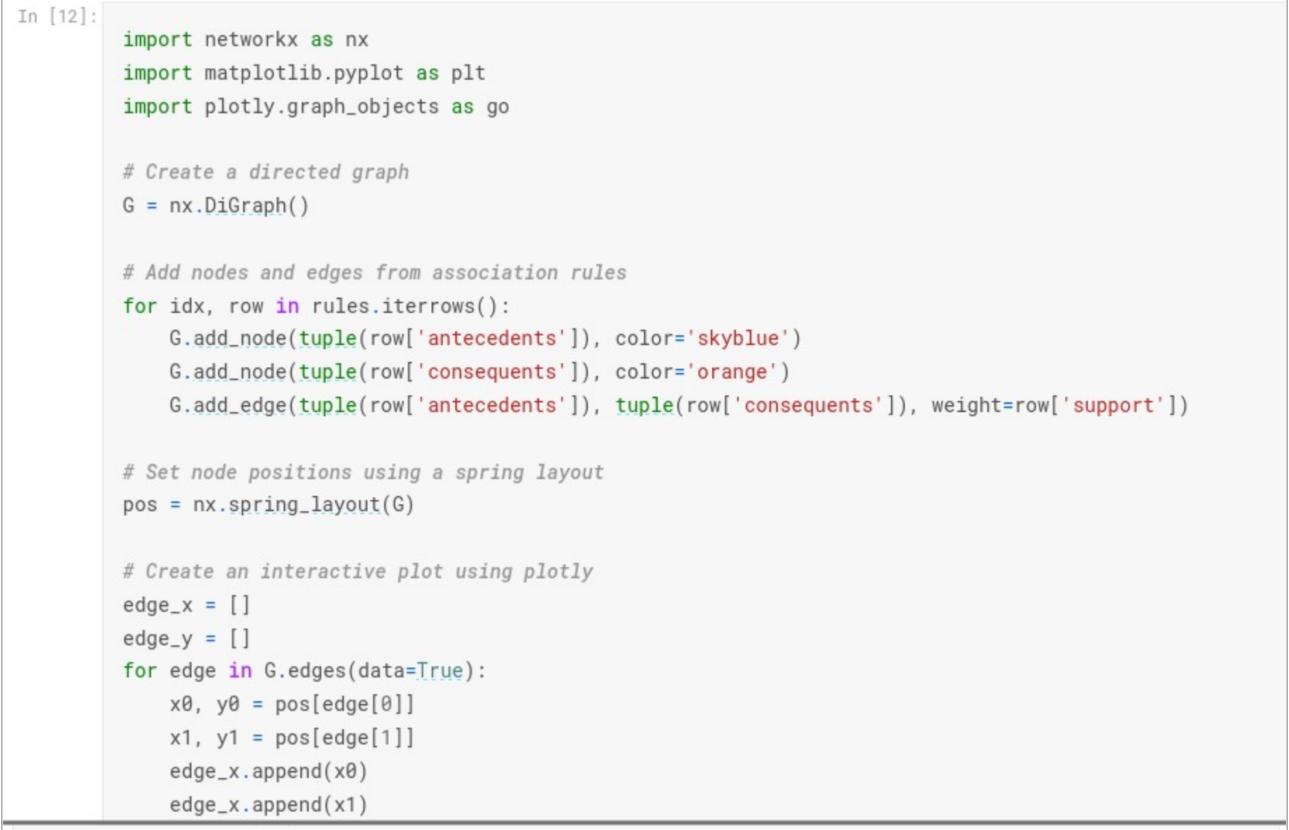


```
In [11]:
import plotly.express as px

# 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()
```



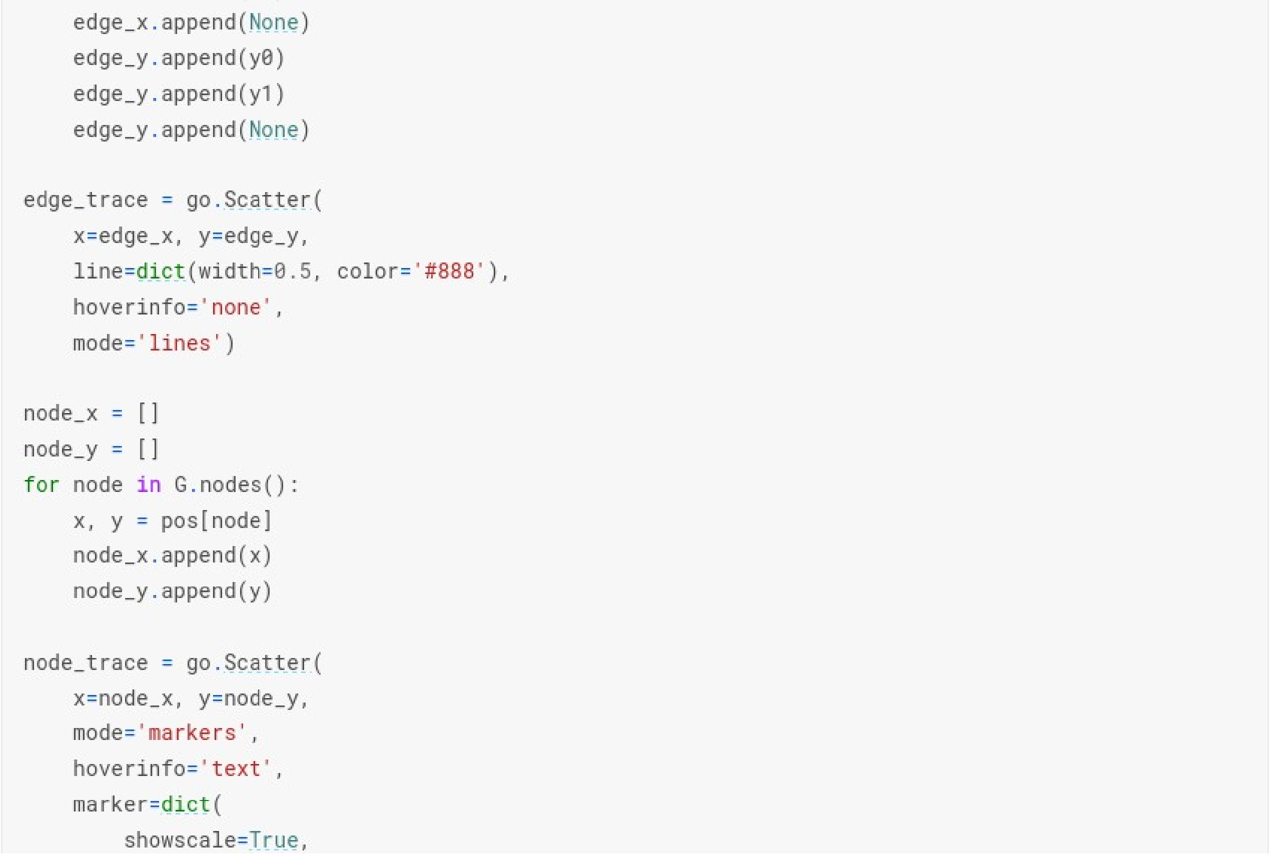
Interactive Network Visualization for Association Rules

We utilize the NetworkX and Plotly libraries to create an interactive network graph visualizing the association rules. This graph represents relationships between antecedent and consequent items, showcasing support as edge weights.

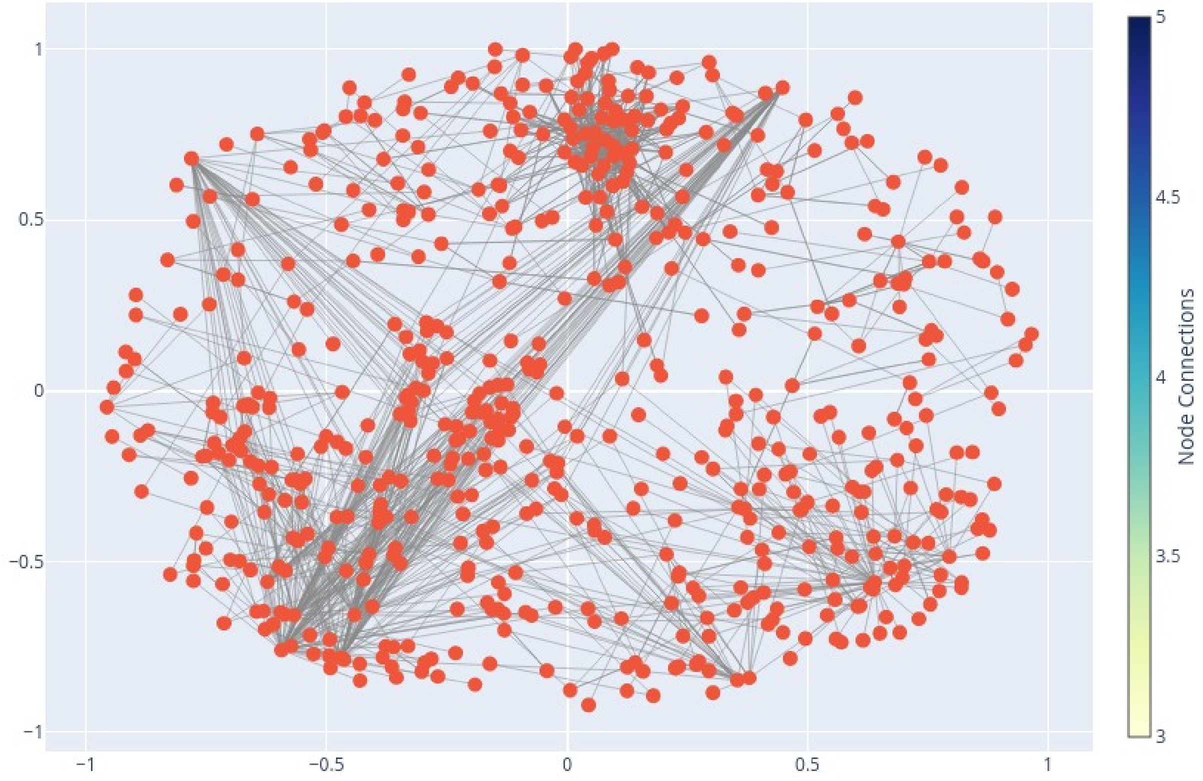


In [12]:

```
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)
```

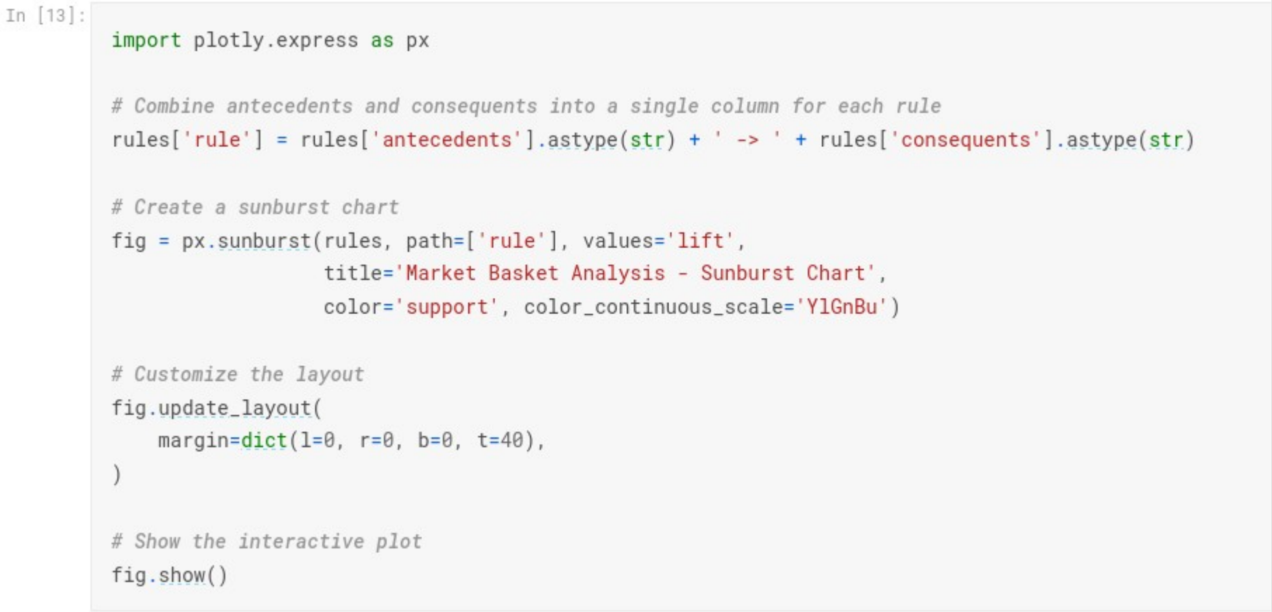


```
    edge_x.append(None)
    edge_y.append(y0)
    edge_y.append(y1)
    edge_y.append(None)

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()
```

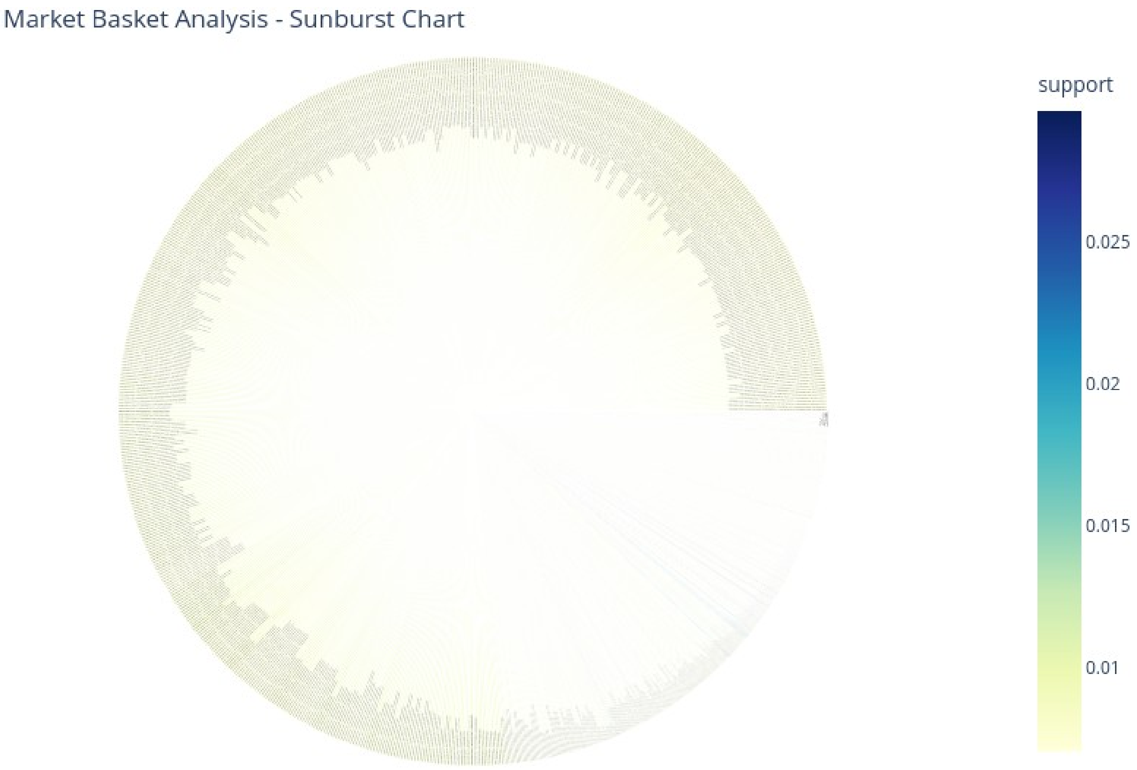


Interactive Sunburst Chart for Association Rules

We use Plotly Express to create an interactive sunburst chart visualizing association rules. This chart represents the relationships between antecedent and consequent items, showcasing lift as well as support through color intensity.



```
In [13]:
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()
```



Actionability and Recommendations:

- **Interpretation**: Interpret the findings from your analysis in a business context. Explain what the association rules reveal about customer purchasing behavior. Offer insights into common purchasing patterns, complementary products, or cross-selling opportunities.
- **Business Impact**: Discuss how the results can impact the business. Highlight potential improvements in marketing strategies, inventory management, or customer experience. Quantify the expected benefits or changes.
- **Recommendations**: Provide actionable recommendations based on the associations discovered. For example, if certain products frequently co-occur in transactions, recommend bundling or promoting them together. Suggest specific strategies for product placement or cross-selling.

Conclusion

In the conclusion phase of your Market Basket Analysis project, you should aim to provide a comprehensive and insightful summary of the work you've done and the significance of your findings. This is where you tie together all the pieces of your analysis and explain the implications for the business.