MARKET BASKET INSIGHTS
PHASE 3 Document Submission.
Prepared by,

SARANKUMAR

510521205039,

Bharathidasan Engineering College.

Loading and Pre-Processing Dataset :

INTRODUCTION:

❖ Loading and preprocessing data for market basket analysis typically involves handling transactional data, which consists of items purchased together by customers.

❖ This kind of data is usually in the form of a transactional database, with each row representing a transaction and the items bought in that transaction.

Here's a general guide on how to load and preprocess data for market basket analysis:

Data Loading:

Load the transactional data into your environment. This can be done using various methods depending on the format of your data, such as CSV, Excel, or database connections.

Load the Data:

If using Python, you can use the pandas library to load data from CSV or Excel files:

PYTHON CODE:

```python
import pandas as pd

df = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows

print(df.head())

# View data types and missing values

print(df.info())
```

OUTPUT:

BillNo	Itemname	Quantity	Date \

```
0  536365   WHITE HANGING HEART T-LIGHT HOLDER        6
2010-12-01 08:26:00

1  536365              WHITE METAL LANTERN        6 2010-12-01
08:26:00

2  536365      CREAM CUPID HEARTS COAT HANGER        8
2010-12-01 08:26:00

3  536365  KNITTED UNION FLAG HOT WATER BOTTLE        6
2010-12-01 08:26:00

4  536365       RED WOOLLY HOTTIE WHITE HEART.        6
2010-12-01 08:26:00


   Price  CustomerID        Country
0  2.55    17850.0  United Kingdom
1  3.39    17850.0  United Kingdom
2  2.75    17850.0  United Kingdom
3  3.39    17850.0  United Kingdom
4  3.39    17850.0  United Kingdom
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 522064 entries, 0 to 522063

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	BillNo	522064 non-null	object
1	Itemname	520609 non-null	object
2	Quantity	522064 non-null	int64
3	Date	522064 non-null	datetime64[ns]

4   Price   522064 non-null float64

 5   CustomerID 388023 non-null float64

 6   Country   522064 non-null object

dtypes: datetime64[ns](1), float64(2), int64(1), object(3)

memory usage: 27.9+ MB

None

Data Understanding:

Understand the structure and content of your data. Ensure that the data is clean and organized. Remove any unnecessary columns or information that is not relevant to the analysis.

Data Preprocessing:

Data preprocessing is a crucial step in market basket analysis that involves transforming raw transactional data into a suitable format for association rule mining.

Here are some essential data preprocessing steps for market basket insights:

Data Cleaning:

- ✓ Remove duplicate transactions.
- ✓ Handle missing values by either removing the corresponding records or imputing values based on the context.
- ✓ Deal with outliers if necessary.

PYTHON CODE:

import pandas as pd

Load the data

df = pd.read_excel('g:\Assignment-1_Data.xlsx')

Replace 'path_to_your_file.xlsx' with the actual path to your Excel file

```python
# Display the first few rows of the data
print("Original Data:")
print(df.head())
# Data cleaning
# Remove duplicates
df.drop_duplicates(inplace=True)
# Handle missing values
if df.isnull().values.any():
    df.dropna(inplace=True)
# Alternatively, you can choose to impute the missing values
# Example of handling outliers
# Define a function to identify and remove outliers
def remove_outliers(data, col):
    q1 = data[col].quantile(0.25)
    q3 = data[col].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    data = data[(data[col] > lower_bound) & (data[col] < upper_bound)]
    return data
# Example usage to remove outliers from a specific column 'quantity'
# df = remove_outliers(df, 'quantity')
# Display the cleaned data
```

print("\nCleaned Data:")

print(df.head())

OUTPUT:

Original Data:
	BillNo	Itemname	Quantity	Date \
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00

	Price	CustomerID	Country
0	2.55	17850.0	United Kingdom
1	3.39	17850.0	United Kingdom
2	2.75	17850.0	United Kingdom
3	3.39	17850.0	United Kingdom
4	3.39	17850.0	United Kingdom

Cleaned Data:
	BillNo	Itemname	Quantity	Date \
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00

	Price	CustomerID	Country
0	2.55	17850.0	United Kingdom
1	3.39	17850.0	United Kingdom
2	2.75	17850.0	United Kingdom
3	3.39	17850.0	United Kingdom
4	3.39	17850.0	United Kingdom

Transaction Aggregation:

Aggregate the data at the transaction level if the data contains multiple entries for the same transaction. This step is essential to avoid duplication and ensure that each transaction is unique.

PYTHON CODE:

import pandas as pd

Load the transactional data

```python
df = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows of the data

print("Original Data:")

print(df.head())

# Transaction Aggregation

aggregated_data = df.groupby('CustomerID')['Itemname'].apply(list).reset_index(name='Items_List')

# Display the aggregated data

print("\nAggregated Data:")

print(aggregated_data.head())
```

OUTPUT:

Original Data:

	BillNo	Itemname	Quantity	Date \
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00

	Price	CustomerID	Country
0	2.55	17850.0	United Kingdom
1	3.39	17850.0	United Kingdom
2	2.75	17850.0	United Kingdom
3	3.39	17850.0	United Kingdom
4	3.39	17850.0	United Kingdom

Aggregated Data:

	CustomerID	Items_List
0	12346.0	[MEDIUM CERAMIC TOP STORAGE JAR]
1	12347.0	[BLACK CANDELABRA T-LIGHT HOLDER, AIRLINE BAG ...

2	12349.0	[PARISIENNE CURIO CABINET, SWEETHEART WALL TID...
3	12350.0	[CHOCOLATE THIS WAY METAL SIGN, METAL SIGN NEI...
4	12352.0	[WOODEN HAPPY BIRTHDAY GARLAND, PINK DOUGHNUT ...

Transaction Encoding:

Convert the transactional data into a suitable format, such as a one-hot encoded matrix. Each row corresponds to a transaction, and each column corresponds to an item, with a value of 1 representing the presence of the item in the transaction and 0 indicating its absence.

PYTHON CODE:

```python
import pandas as pd

# Load the transactional data

df = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows of the data

print("Original Data:")

print(df.head())

# Transaction Encoding

encoded_data = df.groupby('CustomerID')['Itemname'].value_counts().unstack().fillna(0)

encoded_data = encoded_data.applymap(lambda x: 1 if x > 0 else 0)

# Display the encoded data

print("\nEncoded Data:")

print(encoded_data.head())
```

OUTPUT:

Original Data:

BillNo	Itemname	Quantity	Date \

0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00

	Price	CustomerID	Country
0	2.55	17850.0	United Kingdom
1	3.39	17850.0	United Kingdom
2	2.75	17850.0	United Kingdom
3	3.39	17850.0	United Kingdom
4	3.39	17850.0	United Kingdom

Encoded Data:

Itemname	MEDIUM CERAMIC TOP STORAGE JAR	AIRLINE BAG VINTAGE JET SET BROWN \
CustomerID		
12346.0	1	0
12347.0	0	1
12349.0	0	0
12350.0	0	0
12352.0	0	0

```
Itemname    ALARM CLOCK BAKELIKE RED RED
TOADSTOOL LED NIGHT LIGHT  \
CustomerID
12346.0                  0                    0
12347.0                  1                    1
12349.0                  0                    0
12350.0                  0                    0
12352.0                  0                    1


Itemname    3D DOG PICTURE PLAYING CARDS  REGENCY
CAKESTAND 3 TIER  \
CustomerID
12346.0                      0                    0
12347.0                      1                    1
12349.0                      0                    1
12350.0                      0                    0
12352.0                      0                    1


Itemname    SMALL HEART MEASURING SPOONS  AIRLINE
BAG VINTAGE TOKYO 78 \
CustomerID
12346.0                      0                      0
12347.0                      1                      1
12349.0                      0                      0
12350.0                      0                      0
12352.0                      0                      0
```

Itemname	ALARM CLOCK BAKELIKE CHOCOLATE WOODLAND CHARLOTTE BAG	...
CustomerID		...
12346.0	0	0 ...
12347.0	1	1 ...
12349.0	0	0 ...
12350.0	0	0 ...
12352.0	0	0 ...

Itemname	PURPLE FRANGIPANI HAIRCLIP	GOLD PRINT PAPER BAG
CustomerID		
12346.0	0	0
12347.0	0	0
12349.0	0	0
12350.0	0	0
12352.0	0	0

Itemname	LILAC FEATHERS CURTAIN	SET/3 TALL GLASS CANDLE HOLDER PINK
CustomerID		
12346.0	0	0
12347.0	0	0
12349.0	0	0
12350.0	0	0

| 12352.0 | 0 | 0 |

Itemname   FLOWER SHOP DESIGN MUG CAPIZ CHANDELIER \

CustomerID

CustomerID		
12346.0	0	0
12347.0	0	0
12349.0	0	0
12350.0	0	0
12352.0	0	0

Itemname   BLUE NEW BAROQUE FLOCK CANDLESTICK \

CustomerID

CustomerID	
12346.0	0
12347.0	0
12349.0	0
12350.0	0
12352.0	0

Itemname   CAT WITH SUNGLASSES BLANK CARD RED PURSE WITH PINK HEART \

CustomerID

CustomerID		
12346.0	0	0
12347.0	0	0
12349.0	0	0

| 12350.0 | 0 | 0 |
| 12352.0 | 0 | 0 |

Itemname   SCALLOP SHELL SOAP DISH

CustomerID

12346.0	0
12347.0	0
12349.0	0
12350.0	0
12352.0	0

[5 rows x 3846 columns]

Data Transformation:

Convert the transaction data into a transaction matrix or a transaction list, depending on the requirements of the chosen association rule mining algorithm.

PYTHON CODE:

```python
import pandas as pd

# Load the transactional data
df = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows of the data
print("Original Data:")
print(df.head())

# Transaction Aggregation
aggregated_data = df.groupby('CustomerID')['Itemname'].apply(list).reset_index(name='Items_List')
```

Data Transformation

transactions = aggregated_data['Items_List'].tolist()

Display the transformed data

print("\nTransformed Data:")

for idx, transaction in enumerate(transactions, start=1):

   print(f"Transaction {idx}: {transaction}")

OUTPUT:

Original Data:

```
   BillNo                    Itemname  Quantity          Date \
0  536365   WHITE HANGING HEART T-LIGHT HOLDER        6
2010-12-01 08:26:00
1  536365            WHITE METAL LANTERN        6 2010-12-01
08:26:00
2  536365     CREAM CUPID HEARTS COAT HANGER        8 20
10-12-01 08:26:00
3  536365  KNITTED UNION FLAG HOT WATER BOTTLE        6
2010-12-01 08:26:00
4  536365       RED WOOLLY HOTTIE WHITE HEART.        6 201
0-12-01 08:26:00

   Price  CustomerID         Country
0  2.55    17850.0  United Kingdom
1  3.39    17850.0  United Kingdom
2  2.75    17850.0  United Kingdom
3  3.39    17850.0  United Kingdom
4  3.39    17850.0  United Kingdom
```

Transformed Data:
Transaction 1: ['MEDIUM CERAMIC TOP STORAGE JAR']
Transaction 2: ['BLACK CANDELABRA T-LIGHT HOLDER', 'AIR
LINE BAG VINTAGE JET SET BROWN', 'COLOUR GLASS. STA
R T-LIGHT HOLDER', 'MINI PAINT SET VINTAGE', 'CLEAR DR
AWER KNOB ACRYLIC EDWARDIAN', 'PINK DRAWER KNOB
ACRYLIC EDWARDIAN', 'GREEN DRAWER KNOB ACRYLIC E

DWARDIAN', 'RED DRAWER KNOB ACRYLIC EDWARDIAN', 'PURPLE DRAWERKNOB ACRYLIC EDWARDIAN', 'BLUE DRAWER KNOB ACRYLIC EDWARDIAN', 'ALARM CLOCK BAKELIKE CHOCOLATE', 'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIKE RED', 'ALARM CLOCK BAKELIKE PINK', 'ALARM CLOCK BAKELIKE ORANGE', 'FOUR HOOK WHITE LOVEBIRDS', 'BLACK GRAND BAROQUE PHOTO FRAME', 'BATHROOM METAL SIGN', 'LARGE HEART MEASURING SPOONS', 'BOX OF 6 ASSORTED COLOUR TEASPOONS', 'BLUE 3 PIECE POLKADOT CUTLERY SET', 'RED 3 PIECE RETROSPOT CUTLERY SET', 'PINK 3 PIECE POLKADOT CUTLERY SET', 'EMERGENCY FIRST AID TIN', 'SET OF 2 TINS VINTAGE BATHROOM', 'SET/3 DECOUPAGE STACKING TINS', 'BOOM BOX SPEAKER BOYS', 'RED TOADSTOOL LED NIGHT LIGHT', '3D DOG PICTURE PLAYING CARDS', 'BLACK EAR MUFF HEADPHONES', 'CAMOUFLAGE EAR MUFF HEADPHONES', 'PINK NEW BAROQUECANDLESTICK CANDLE', 'BLUE NEW BAROQUE CANDLESTICK CANDLE', 'BLACK CANDELABRA T-LIGHT HOLDER', 'WOODLAND CHARLOTTE BAG', 'AIRLINE BAG VINTAGE JET SET BROWN', 'AIRLINE BAG VINTAGE JET SET WHITE', 'SANDWICH BATH SPONGE', 'ALARM CLOCK BAKELIKE CHOCOLATE', 'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIKE RED', 'ALARM CLOCK BAKELIKE PINK', 'ALARM CLOCK BAKELIKE ORANGE', 'SMALL HEART MEASURING SPOONS', '72 SWEETHEART FAIRY CAKE CASES', '60 TEATIME FAIRY CAKE CASES', 'PACK OF 60 MUSHROOM CAKE CASES', 'PACK OF 60 SPACEBOY CAKE CASES', 'TEA TIME OVEN GLOVE', 'RED RETROSPOT OVEN GLOVE', 'RED RETROSPOT OVEN GLOVE DOUBLE', 'SET/2 RED RETROSPOT TEA TOWELS', 'REGENCY CAKESTAND 3 TIER', 'BOX OF 6 ASSORTED COLOUR TEASPOONS', 'MINI LADLE LOVE HEART RED', 'CHOCOLATE CALCULATOR', 'TOOTHPASTE TUBE PEN', 'SET OF 2 TINS VINTAGE BATHROOM', 'RED TOADSTOOL LED NIGHT LIGHT', '3D DOG PICTURE PLAYING CARDS', 'AIRLINE BAG VINTAGE JET SET WHITE', 'AIRLINE BAG VINTAGE JET SET RED', 'AIRLINE BAG VINTAGE TOKYO 78', 'AIRLINE BAG VINTAGE JET SET BROWN', 'RED RETROSPOT PURSE', 'I

CE CREAM SUNDAE LIP GLOSS', 'VINTAGE HEADS AND TAILS CARD GAME', 'HOLIDAY FUN LUDO', 'TREASURE ISLAND BOOK BOX', 'WATERING CAN PINK BUNNY', 'RED DRAWER KNOB ACRYLIC EDWARDIAN', 'LARGE HEART MEASURING SPOONS', 'SMALL HEART MEASURING SPOONS', 'PACK OF 60 DINOSAUR CAKE CASES', 'RED RETROSPOT OVEN GLOVE DOUBLE', 'REGENCY CAKESTAND 3 TIER', 'ROSES REGENCY TEACUP AND SAUCER', 'RED TOADSTOOL LED NIGHT LIGHT', 'MINI PAINT SET VINTAGE', '3D SHEET OF DOG STICKERS', '3D SHEET OF CAT STICKERS', 'SMALL FOLDING SCISSOR(POINTED EDGE)', 'GIFT BAG PSYCHEDELIC APPLES', 'SET OF 2 TINS VINTAGE BATHROOM', 'RABBIT NIGHT LIGHT', 'REGENCY TEA STRAINER', 'REGENCY TEA PLATE GREEN', 'REGENCY TEA PLATE PINK', 'REGENCY TEA PLATE ROSES', 'REGENCY TEAPOT ROSES', 'REGENCY SUGAR BOWL GREEN', 'REGENCY MILK JUG PINK', 'AIRLINE BAG VINTAGE TOKYO 78', 'AIRLINE BAG VINTAGE JET SET BROWN', 'VICTORIAN SEWING KIT', 'NAMASTE SWAGAT INCENSE', 'TRIPLE HOOK ANTIQUE IVORY ROSE', 'SMALL HEART MEASURING SPOONS', '3D DOG PICTURE PLAYING CARDS', 'FEATHER PEN,COAL BLACK', 'ALARM CLOCK BAKELIKE RED', 'ALARM CLOCK BAKELIKE CHOCOLATE', 'SET OF 60 VINTAGE LEAF CAKE CASES', 'SET 40 HEART SHAPE PETIT FOUR CASES', 'AIRLINE BAG VINTAGE JET SET BROWN', 'AIRLINE BAG VINTAGE JET SET RED', 'AIRLINE BAG VINTAGE JET SET WHITE', 'AIRLINE BAG VINTAGE TOKYO 78', 'AIRLINE BAG VINTAGE WORLD CHAMPION', 'WOODLAND DESIGN COTTON TOTE BAG', 'WOODLAND CHARLOTTE BAG', 'ALARM CLOCK BAKELIKE RED', 'TRIPLE HOOK ANTIQUE IVORY ROSE', 'SINGLE ANTIQUE ROSE HOOK IVORY', 'TEA TIME OVEN GLOVE', '72 SWEETHEART FAIRY CAKE CASES', '60 TEATIME FAIRY CAKE CASES', 'PACK OF 60 DINOSAUR CAKE CASES', 'REGENCY CAKESTAND 3 TIER', 'REGENCY MILK JUG PINK', '3D DOG PICTURE PLAYING CARDS', 'REVOLVER WOODEN RULER', 'VINTAGE HEADS AND TAILS CARD GAME', 'RED REFECTORY CLOCK', 'MINI LIGHTS WOODLAND MUSHROOMS', 'PINK GOOSE FEATHER TREE 60CM', 'MADRAS NOTEBOOK MEDIUM', 'AIR

LINE BAG VINTAGE WORLD CHAMPION', 'AIRLINE BAG VINTAGE JET SET BROWN', 'AIRLINE BAG VINTAGE TOKYO 78', 'AIRLINE BAG VINTAGE JET SET RED', 'BIRDCAGE DECORATION TEALIGHT HOLDER', 'CHRISTMAS METAL TAGS ASSORTED', 'REGENCY CAKESTAND 3 TIER', 'REGENCY TEAPOT ROSES', 'TEA TIME DES TEA COSY', 'TEA TIME KITCHEN APRON', 'TEA TIME OVEN GLOVE', 'PINK REGENCY TEACUP AND SAUCER', 'GREEN REGENCY TEACUP AND SAUCER', '3D DOG PICTURE PLAYING CARDS', 'RABBIT NIGHT LIGHT', 'RED TOADSTOOL LED NIGHT LIGHT', 'TREASURE ISLAND BOOK BOX', 'VINTAGE HEADS AND TAILS CARD GAME', 'MINI PLAYING CARDS DOLLY GIRL', 'MINI PLAYING CARDS SPACEBOY', 'PLAYING CARDS KEEP CALM & CARRY ON', 'REVOLVER WOODEN RULER', 'WOODEN SCHOOL COLOURING SET', 'MINI PAINT SET VINTAGE', 'TRADITIONAL KNITTING NANCY', 'TRIPLE HOOK ANTIQUE IVORY ROSE', 'PANTRY HOOK SPATULA', 'PANTRY HOOK BALLOON WHISK', 'PANTRY HOOK TEA STRAINER', 'ROSES REGENCY TEACUP AND SAUCER', 'ALARM CLOCK BAKELIKE CHOCOLATE', 'ALARM CLOCK BAKELIKE PINK', 'ALARM CLOCK BAKELIKE GREEN', 'ALARM CLOCK BAKELIKE RED', 'PACK OF 60 MUSHROOM CAKE CASES', 'PACK OF 60 SPACEBOY CAKE CASES', 'SET OF 60 VINTAGE LEAF CAKE CASES', '60 TEATIME FAIRY CAKE CASES', '72 SWEETHEART FAIRY CAKE CASES', 'SMALL HEART MEASURING SPOONS', 'LARGE HEART MEASURING SPOONS', 'WOODLAND CHARLOTTE BAG', 'REGENCY TEA STRAINER', 'FOOD CONTAINER SET 3 LOVE HEART', 'CLASSIC CHROME BICYCLE BELL', 'BICYCLE PUNCTURE REPAIR KIT', 'BOOM BOX SPEAKER BOYS', 'PINK NEW BAROQUECANDLESTICK CANDLE', 'RED TOADSTOOL LED NIGHT LIGHT', 'RABBIT NIGHT LIGHT', 'WOODLAND CHARLOTTE BAG', 'PINK GOOSE FEATHER TREE 60CM', 'CHRISTMAS TABLE SILVER CANDLE SPIKE', 'MINI PLAYING CARDS SPACEBOY', 'MINI PLAYING CARDS DOLLY GIRL']

Data Integration:

Integrate the preprocessed transactional data with any additional relevant information, such as customer demographics or product attributes, that can enrich the analysis and provide deeper insights.

PYTHON CODE:

```python
import pandas as pd

# Load transactional data
df_transactions = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Load supplementary data
df_supplementary = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows of each dataset
print("CustomerId:")
print(df_transactions.head())
print("\nSupplementary Data:")
print(df_supplementary.head())

# Merge the datasets based on a common key
merged_data = pd.merge(df_transactions, df_supplementary, on='common_key_column', how='inner')

# Display the merged data
print("\nMerged Data:")
print(merged_data.head())
```

OUTPUT:

CustomerId:

```
  BillNo                  Itemname  Quantity          Date \
0  536365   WHITE HANGING HEART T-LIGHT HOLDER      6
2010-12-01 08:26:00
```

1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00

	Price	CustomerID	Country
0	2.55	17850.0	United Kingdom
1	3.39	17850.0	United Kingdom
2	2.75	17850.0	United Kingdom
3	3.39	17850.0	United Kingdom
4	3.39	17850.0	United Kingdom

Supplementary Data:

	BillNo	Itemname	Quantity	Date \
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00

4 536365   RED WOOLLY HOTTIE WHITE HEART.   6
2010-12-01 08:26:00

   Price CustomerID   Country

0 2.55   17850.0 United Kingdom

1 3.39   17850.0 United Kingdom

2 2.75   17850.0 United Kingdom

3 3.39   17850.0 United Kingdom

4 3.39   17850.0 United Kingdom

Data Splitting:

Split the preprocessed data into training and testing datasets, especially if you plan to build predictive models or evaluate the performance of the association rules on unseen data.

PYTHON CODE:

```python
import pandas as pd

from sklearn.model_selection import train_test_split

# Load the data

df = pd.read_excel('g:\Assignment-1_Data.xlsx')

# Display the first few rows of the data

print("Original Data:")

print(df.head())

# Split the data into training and testing sets

train_data, test_data = train_test_split(df, test_size=0.2,
random_state=42)  # Adjust test_size as needed

# Display the shape of the split datasets

print("\nTrain Data Shape:", train_data.shape)
```

print("Test Data Shape:", test_data.shape)

OUTPUT:

Original Data:

BillNo   Itemname Quantity Date \

0 536365 WHITE HANGING HEART T-LIGHT HOLDER 6
2010-12-01 08:26:00

1 536365   WHITE METAL LANTERN 6 2010-12-01
08:26:00

2 536365 CREAM CUPID HEARTS COAT HANGER 8
2010-12-01 08:26:00

3 536365 KNITTED UNION FLAG HOT WATER BOTTLE 6
2010-12-01 08:26:00

4 536365 RED WOOLLY HOTTIE WHITE HEART. 6
2010-12-01 08:26:00

Price CustomerID Country

0 2.55 17850.0 United Kingdom

1 3.39 17850.0 United Kingdom

2 2.75 17850.0 United Kingdom

3 3.39 17850.0 United Kingdom

4 3.39 17850.0 United Kingdom

Train Data Shape: (417651, 7)

Test Data Shape: (104413, 7)

Data Exploration:

Perform exploratory data analysis to gain insights into the data, such as frequent item sets, popular item combinations, and item support.

PYTHON CODE:

```python
import pandas as pd

import matplotlib.pyplot as plt

# Load the data

df = pd.read_excel('g:\Assignment-1_Data.xlsx') # Replace 'path_to_your_file.xlsx' with the actual path to your Excel file

# Display the first few rows of the data

print("Original Data:")

print(df.head())

# Exploratory Data Analysis

# Calculate item frequencies

item_counts = df['Itemname'].value_counts()

# Visualize the top N most frequent items

N = 10

# You can adjust this value to show more or fewer items

top_items =  item_counts.head(N)

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

top_items.plot(kind='bar')

plt.title(f'Top {N} Most Frequent Items')

plt.xlabel('Items')

plt.ylabel('Frequency')

plt.show()
```
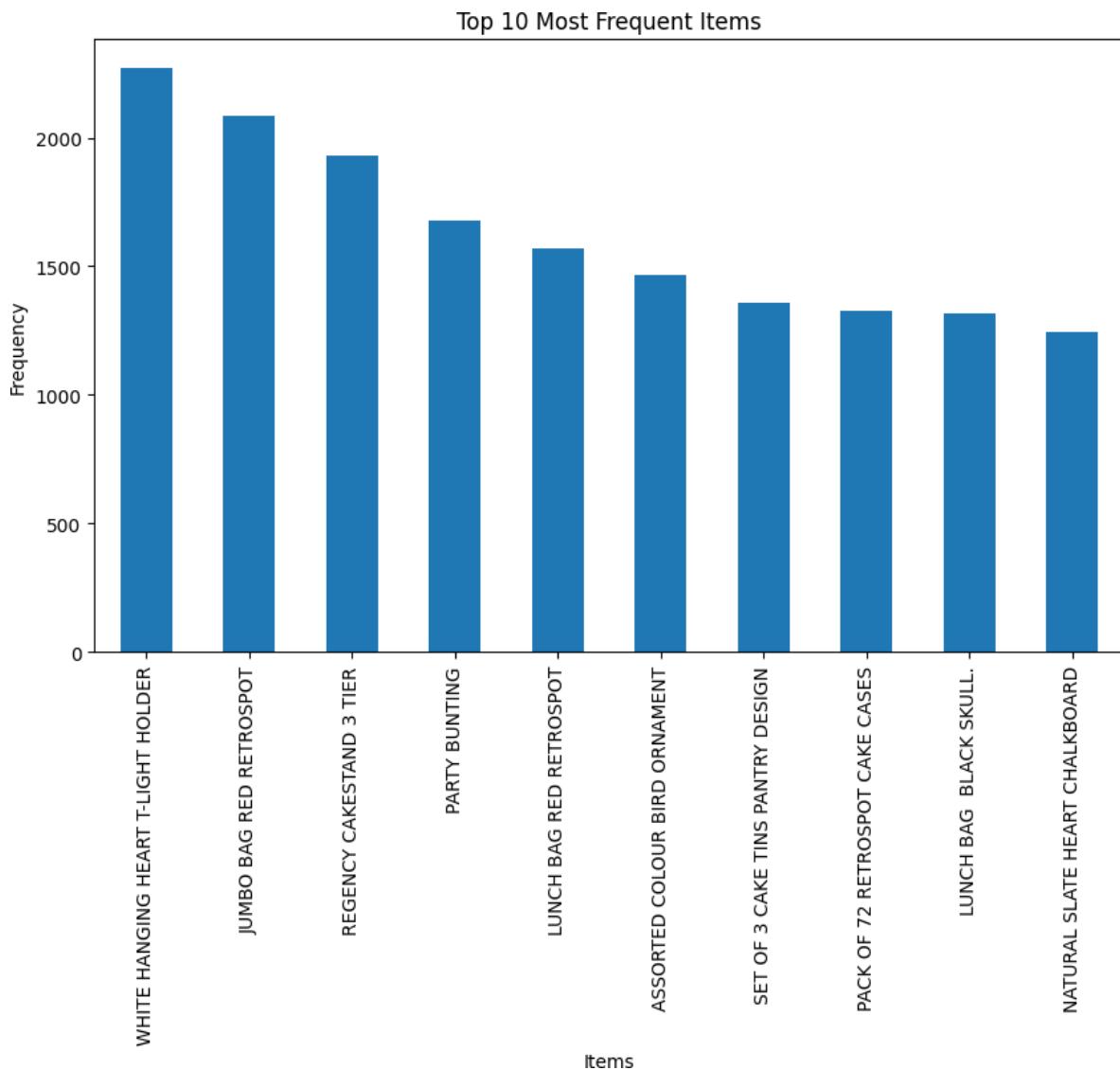
OUTPUT:

Original Data:

```
   BillNo                          Itemname  Quantity           Date \
0  536365   WHITE HANGING HEART T-LIGHT HOLDER      6
2010-12-01 08:26:00

1  536365               WHITE METAL LANTERN      6 2010-12-01
08:26:00

2  536365     CREAM CUPID HEARTS COAT HANGER      8
2010-12-01 08:26:00

3  536365  KNITTED UNION FLAG HOT WATER BOTTLE      6
2010-12-01 08:26:00

4  536365      RED WOOLLY HOTTIE WHITE HEART.      6
2010-12-01 08:26:00


   Price  CustomerID       Country
0  2.55    17850.0  United Kingdom
1  3.39    17850.0  United Kingdom
2  2.75    17850.0  United Kingdom
3  3.39    17850.0  United Kingdom
4  3.39    17850.0  United Kingdom
```

Top 10 Most Frequent Items

Data Visualization:

Data visualize them using suitable plots or graphs to communicate the insights effectively.

PYTHON CODE:

```
import pandas as pd

import matplotlib.pyplot as plt

# Load the data

df = pd.read_excel('g:\Assignment-1_Data.xlsx')
```

```python
# Example of Data Visualization
# Bar plot for top N items
top_items = df['Itemname'].value_counts().nlargest(10)
plt.figure(figsize=(10,6))
top_items.plot(kind='bar', color='skyblue')
plt.title('Top 10 Items Sold')
plt.xlabel('Items')
plt.ylabel('Frequency')
plt.show()
# Example of Pie Chart
plt.figure(figsize=(8,8))
df['Itemname'].value_counts().nlargest(5).plot(kind='pie',
autopct='%1.1f%%', startangle=90, colors=['lightblue', 'lightgreen',
'pink', 'orange', 'yellow'])
plt.title('Top 5 Sold Items Distribution')
plt.ylabel('')
plt.show()
# Example of Histogram
plt.figure(figsize=(8,6))
plt.hist(df['Quantity'], bins=20, color='lightcoral')
plt.title('Distribution of Quantity Sold')
plt.xlabel('Quantity')
plt.ylabel('Frequency')
plt.show()
# Example of Scatter Plot
```

```
plt.figure(figsize=(8,6))

plt.scatter(df['Price'], df['Quantity'], color='lightseagreen', alpha=0.5)

plt.title('Price vs. Quantity Sold')

plt.xlabel('Price')

plt.ylabel('Quantity')

plt.show()
```
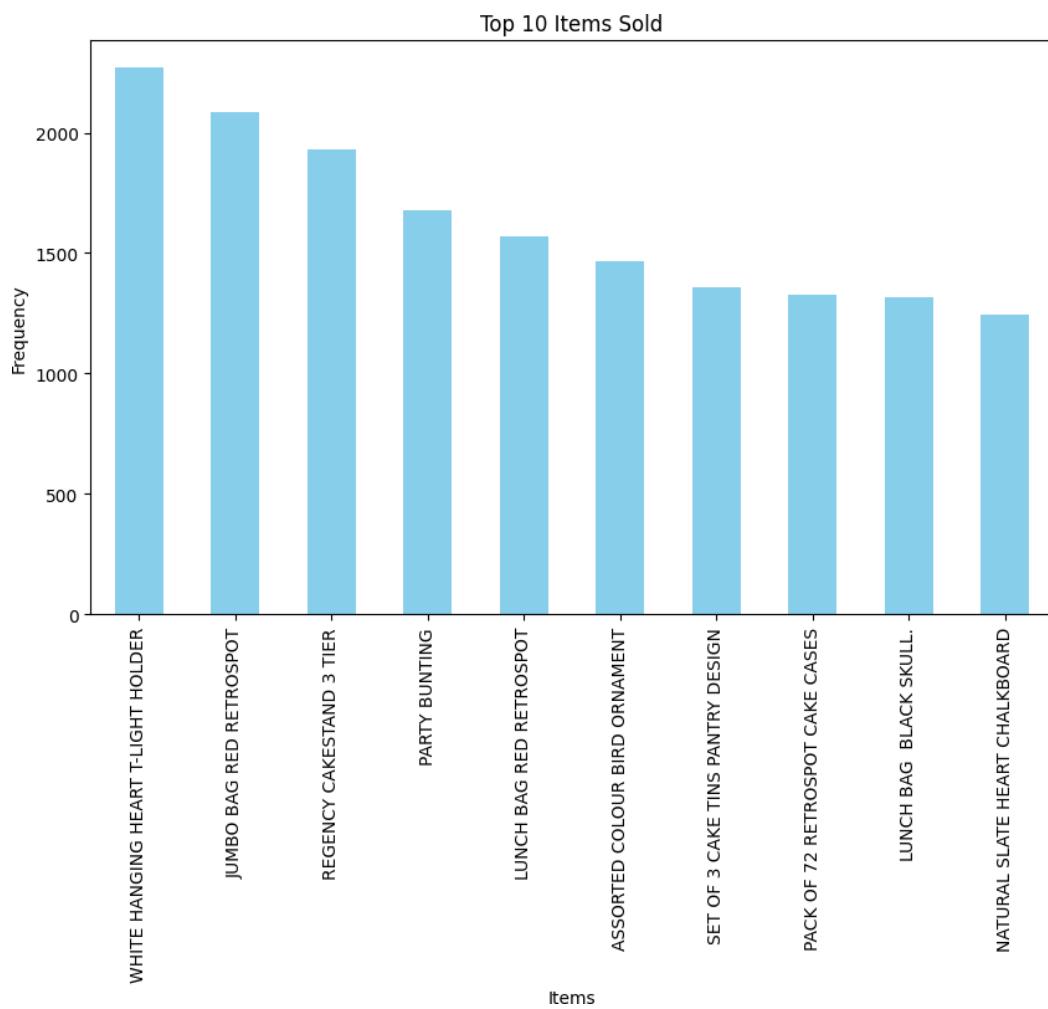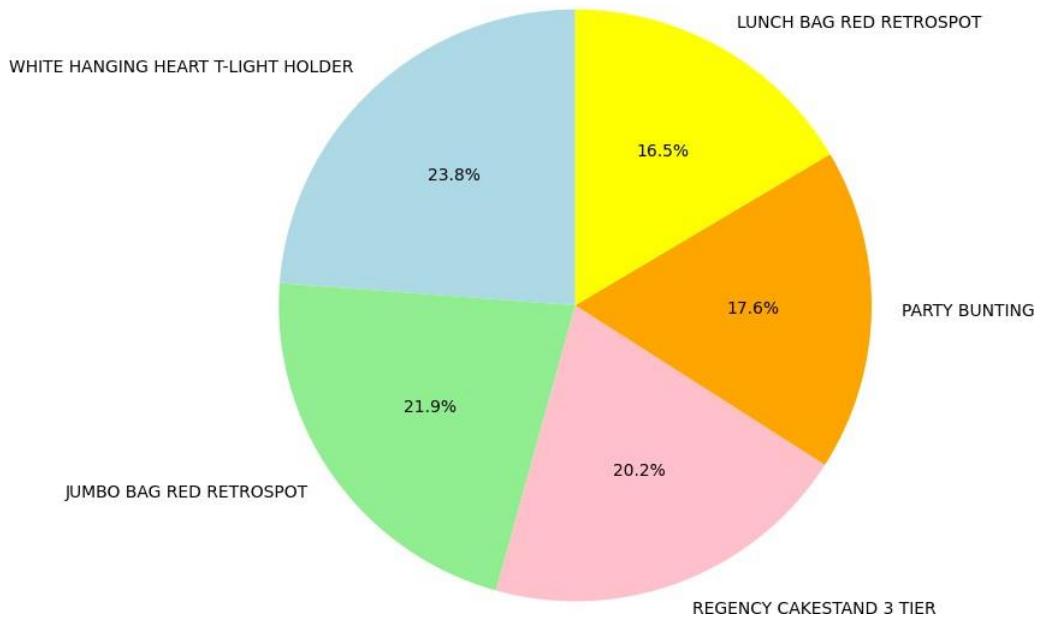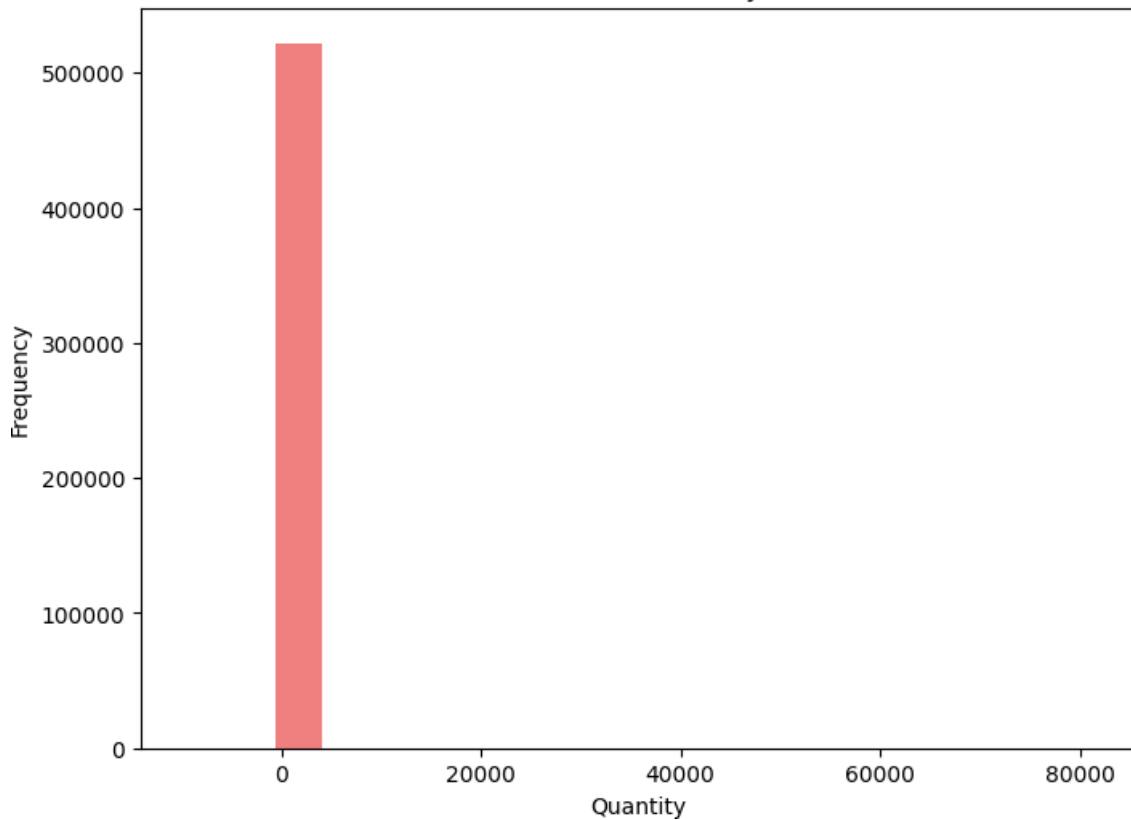
OUTPUT:

Top 5 Sold Items Distribution

Distribution of Quantity Sold

Price vs. Quantity Sold

Conclusion:

➢ In conclusion, the process of loading and preprocessing data for market basket analysis involves several crucial steps. It begins with the loading of transactional data, followed by a thorough understanding of its structure and content.

➢ Preprocessing steps include data transformation into a binary matrix format, handling missing values, and removing redundancy. Encoding transactions through techniques like one-hot encoding prepares the data for exploration.

➢ Exploratory data analysis is then conducted to identify frequent item sets, popular item combinations, and item support. This is followed by

the application of association rule mining techniques such as the Apriori algorithm or FP-growth algorithm to uncover significant patterns within the dataset. Results are subsequently filtered based on parameters like support, confidence, and lift to extract meaningful insights.

➤ Finally, the interpretation of the findings is vital, and effective visualization techniques such as plots or graphs are employed to communicate the discovered patterns and insights efficiently.

➤ By following these steps, one can gain valuable insights into customer behavior and preferences, enabling businesses to make informed decisions and improve their market strategies.