Market Basket Analysis

June 14, 2024

0.0.1 1. Data Collection and Preparation

[1]:
```
import pandas as pd
```

[2]:
```
#Load dataset
df=pd.read_excel(r"C:\Users\USER\Documents\Data Portfolio␣
↪Projects\Retail\Market Basket Analysis\Jomo_Holdings_Transactions.xlsx")
df.head()
```

[2]:

	TransactionID	CustomerID	ProductID	ProductName	Category	Quantity	Price	\
0	1	101	1001	Apple	Fruits	3	0.5	
1	2	102	1002	Banana	Fruits	2	0.2	
2	3	101	1003	Milk	Dairy	1	1.0	
3	4	103	1004	Bread	Bakery	1	1.5	
4	5	104	1001	Apple	Fruits	5	0.5	

	Date	CustomerAge	CustomerGender	CustomerLocation
0	2023-01-01	34	M	Urban
1	2023-01-01	29	F	Rural
2	2023-01-02	34	M	Urban
3	2023-01-03	45	M	Suburban
4	2023-01-04	23	F	Urban

[ ]:
```

```

What is the structure of the data?

[3]:
```
# Display data types of each column
df.dtypes
```

[3]:
```
TransactionID         int64
CustomerID            int64
ProductID             int64
ProductName          object
Category             object
Quantity              int64
Price               float64
Date                 object
CustomerAge           int64
```

```
CustomerGender       object
CustomerLocation     object
dtype: object
```

[ ]:

Are there any missing values?

[4]:
```
# Check for missing values
df.isnull().sum()
```

[4]:
```
TransactionID       0
CustomerID          0
ProductID           0
ProductName         0
Category            0
Quantity            0
Price               0
Date                0
CustomerAge         0
CustomerGender      0
CustomerLocation    0
dtype: int64
```

[ ]:

What preprocessing steps are necessary?

[5]:
```
# Convert 'Date' column to datetime type
df['Date'] = pd.to_datetime(df['Date'])
```

[6]:
```
# Display the updated data types
df.dtypes
```

[6]:
```
TransactionID                int64
CustomerID                   int64
ProductID                    int64
ProductName                 object
Category                    object
Quantity                     int64
Price                      float64
Date                datetime64[ns]
CustomerAge                  int64
CustomerGender              object
CustomerLocation            object
dtype: object
```

[ ]:

[ ]:

0.0.2 2. Descriptive Analysis

What are the overall sales trends?

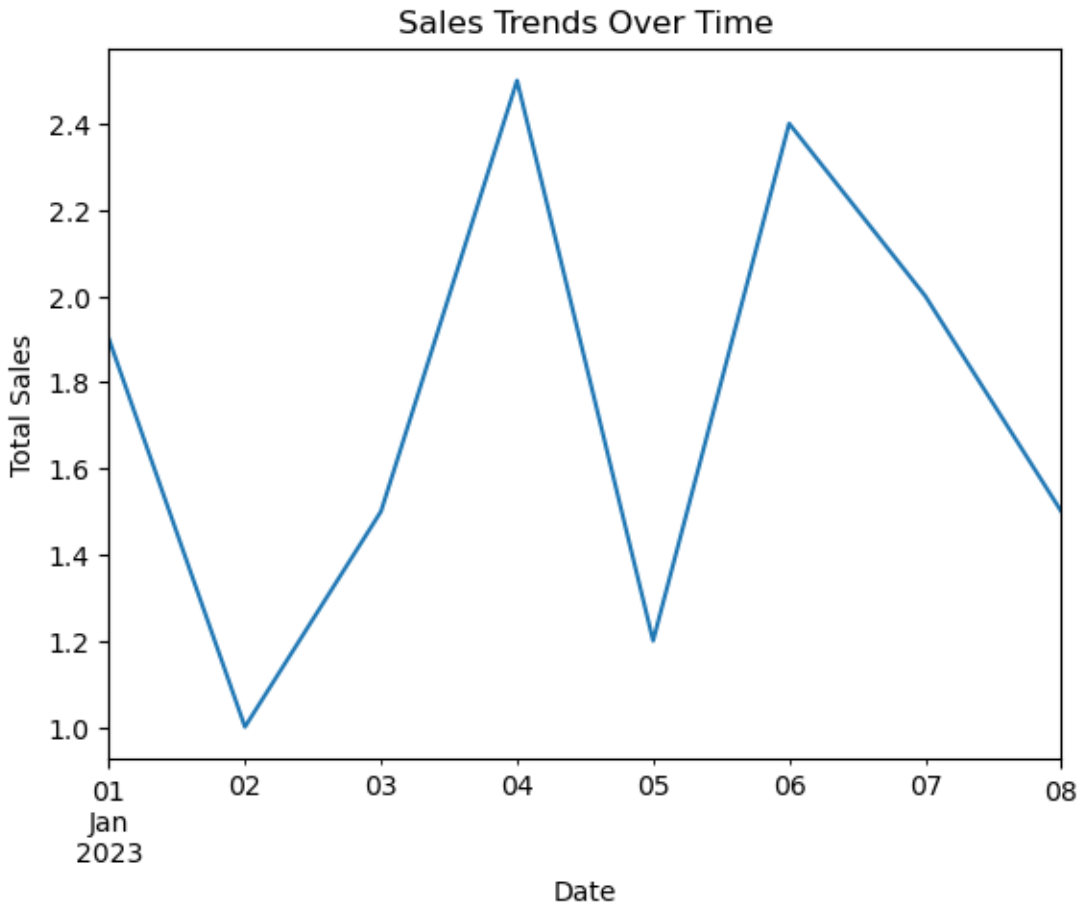
[7]:
```
import matplotlib.pyplot as plt
```

[8]:
```
# Calculate total sales and number of transactions
total_sales = df['Quantity'] * df['Price']
df['TotalSales'] = total_sales
num_transactions = df['TransactionID'].nunique()
average_transaction_value = total_sales.sum() / num_transactions
```

[9]:
```
# Print the summary statistics
summary_stats = {
    "Total Sales": total_sales.sum(),
    "Number of Transactions": num_transactions,
    "Average Transaction Value": average_transaction_value
}
summary_stats
```

[9]:
```
{'Total Sales': 14.0,
 'Number of Transactions': 10,
 'Average Transaction Value': 1.4}
```

[10]:
```
# Plot sales trends over time
sales_trends = df.groupby('Date')['TotalSales'].sum()
sales_trends.plot(kind='line', title='Sales Trends Over Time', xlabel='Date',␣
 ↪ylabel='Total Sales')
plt.show()
```



[ ]:

Who are the customers?

[11]:
```
# Display customer demographics summary
customer_demographics = df[['CustomerID', 'CustomerAge', 'CustomerGender',␣
↪'CustomerLocation']].drop_duplicates()
customer_demographics_summary = customer_demographics.describe(include='all')
customer_demographics_summary
```

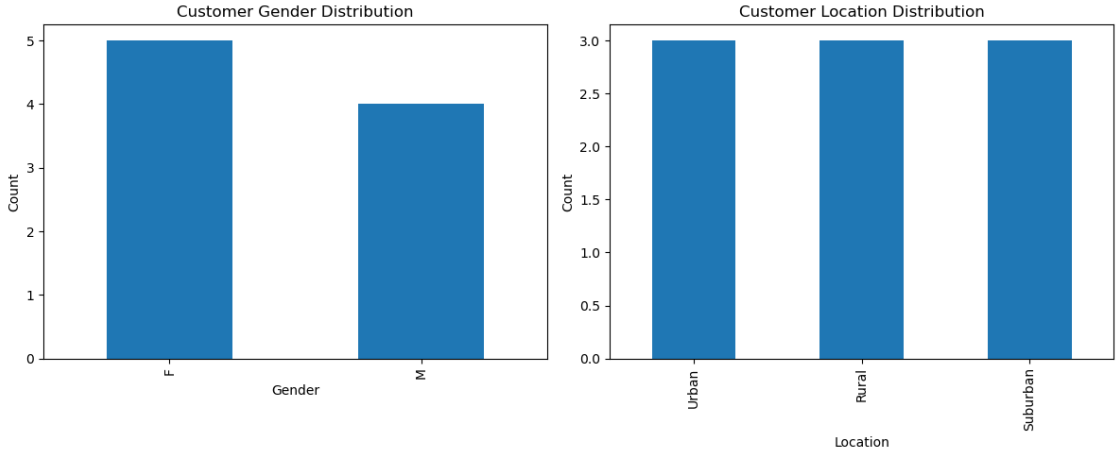
[11]:

	CustomerID	CustomerAge	CustomerGender	CustomerLocation
count	9.000000	9.000000	9	9
unique	NaN	NaN	2	3
top	NaN	NaN	F	Urban
freq	NaN	NaN	5	3
mean	105.000000	33.000000	NaN	NaN
std	2.738613	6.745369	NaN	NaN
min	101.000000	23.000000	NaN	NaN

```
25%     103.000000    29.000000          NaN          NaN
50%     105.000000    31.000000          NaN          NaN
75%     107.000000    37.000000          NaN          NaN
max     109.000000    45.000000          NaN          NaN
```

[12]:
```
# Visualize customer distribution by gender and location
gender_distribution = customer_demographics['CustomerGender'].value_counts()
location_distribution = customer_demographics['CustomerLocation'].value_counts()

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
gender_distribution.plot(kind='bar', ax=axes[0], title='Customer Gender␣
 ↪Distribution', xlabel='Gender', ylabel='Count')
location_distribution.plot(kind='bar', ax=axes[1], title='Customer Location␣
 ↪Distribution', xlabel='Location', ylabel='Count')
plt.tight_layout()
plt.show()
```



[ ]:

[ ]:

0.0.3 3. Market Basket Analysis

What are the frequent itemsets?

The Apriori algorithm will be used to find frequent itemsets.

[13]:
```
from mlxtend.frequent_patterns import apriori, association_rules
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[13], line 1
```

```
----> 1 from mlxtend.frequent_patterns import apriori, association_rules

ModuleNotFoundError: No module named 'mlxtend'
```

[14]:
```
!pip install mlxtend
```

```
Collecting mlxtend
  Obtaining dependency information for mlxtend from https://files.pythonhosted.o
rg/packages/1c/07/512f6a780239ad6ce06ce2aa7b4067583f5ddcfc7703a964a082c706a070/m
lxtend-0.23.1-py3-none-any.whl.metadata
  Downloading mlxtend-0.23.1-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: scipy>=1.2.1 in c:\users\user\anaconda3\lib\site-
packages (from mlxtend) (1.10.1)
Requirement already satisfied: numpy>=1.16.2 in
c:\users\user\anaconda3\lib\site-packages (from mlxtend) (1.24.3)
Requirement already satisfied: pandas>=0.24.2 in
c:\users\user\anaconda3\lib\site-packages (from mlxtend) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.2 in
c:\users\user\anaconda3\lib\site-packages (from mlxtend) (1.3.0)
Requirement already satisfied: matplotlib>=3.0.0 in
c:\users\user\anaconda3\lib\site-packages (from mlxtend) (3.7.1)
Requirement already satisfied: joblib>=0.13.2 in
c:\users\user\anaconda3\lib\site-packages (from mlxtend) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\user\anaconda3\lib\site-
packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(23.0)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\user\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\user\anaconda3\lib\site-
```

```
packages (from pandas>=0.24.2->mlxtend) (2022.7)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\user\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend)
(2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Downloading mlxtend-0.23.1-py3-none-any.whl (1.4 MB)
   ---------------------------------------- 0.0/1.4 MB ? eta -:--:--
    --------------------------------------- 0.0/1.4 MB ? eta -:--:--
   -- ------------------------------------- 0.1/1.4 MB 1.3 MB/s eta 0:00:02
   --- ------------------------------------ 0.1/1.4 MB 1.3 MB/s eta 0:00:02
   -------- ------------------------------- 0.3/1.4 MB 2.0 MB/s eta 0:00:01
   ---------- ----------------------------- 0.4/1.4 MB 2.1 MB/s eta 0:00:01
   ------------- -------------------------- 0.5/1.4 MB 2.3 MB/s eta 0:00:01
   ----------------- ---------------------- 0.7/1.4 MB 2.6 MB/s eta 0:00:01
   -------------------- ------------------- 0.8/1.4 MB 2.9 MB/s eta 0:00:01
   ------------------------- -------------- 1.0/1.4 MB 3.0 MB/s eta 0:00:01
   ------------------------------ --------- 1.2/1.4 MB 3.2 MB/s eta 0:00:01
   ---------------------------------- ----- 1.3/1.4 MB 3.1 MB/s eta 0:00:01
   ---------------------------------------  1.4/1.4 MB 3.3 MB/s eta 0:00:01
   ---------------------------------------- 1.4/1.4 MB 3.1 MB/s eta 0:00:00
Installing collected packages: mlxtend
Successfully installed mlxtend-0.23.1
```

[15]:
```
from mlxtend.frequent_patterns import apriori, association_rules
```

[16]:
```
# Prepare data for market basket analysis
basket = df.groupby(['TransactionID', 'ProductName'])['Quantity'].sum().
 ↪unstack().fillna(0)
basket = basket.applymap(lambda x: 1 if x > 0 else 0)
```

[17]:
```
# Find frequent itemsets
frequent_itemsets = apriori(basket, min_support=0.1, use_colnames=True)
frequent_itemsets.sort_values(by='support', ascending=False, inplace=True)

frequent_itemsets
```

```
C:\Users\USER\anaconda3\Lib\site-
packages\mlxtend\frequent_patterns\fpcommon.py:109: DeprecationWarning:
DataFrames with non-bool types result in worse computationalperformance and
their support might be discontinued in the future.Please use a DataFrame with
bool type
  warnings.warn(
```

[17]:

	support	itemsets
0	0.2	(Apple)
1	0.2	(Banana)
2	0.2	(Bread)

```
4      0.2    (Milk)
3      0.1    (Eggs)
5      0.1  (Orange)
```

[ ]:

[ ]: