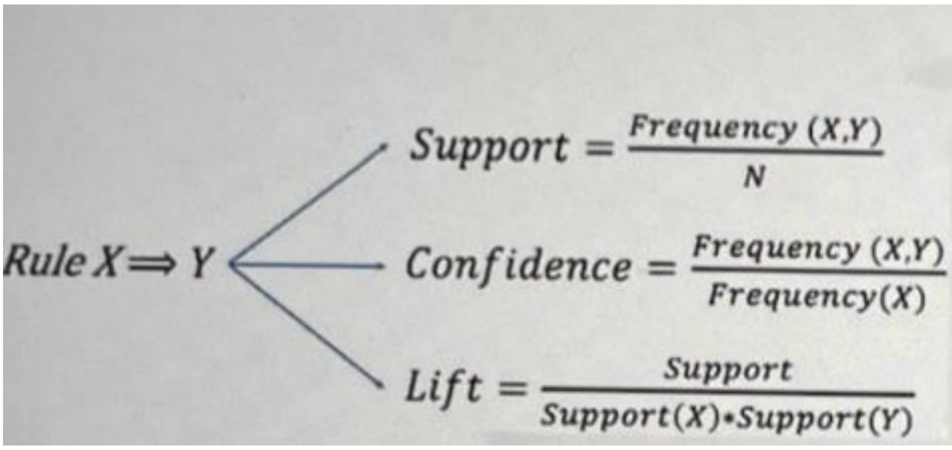
EXERCISE	9. ASSOCIATION RULE LEARNING USING APRIORI ALGORITHM
DATE	05.11.2024

AIM:

For the given dataset, generate association rule learning using Apriori Algorithm with support threshold = 8% and confidence threshold = 50%.

DESCRIPTION:

Association rule learning works on the concept of If and Else Statement, such as if A then B. If element is called antecedent, and then statement is called as Consequent. To measure the associations between thousands of data items, there are several metrics.



$$Rule\ X \Rightarrow Y$$

$$Support = \frac{Frequency\ (X,Y)}{N}$$

$$Confidence = \frac{Frequency\ (X,Y)}{Frequency(X)}$$

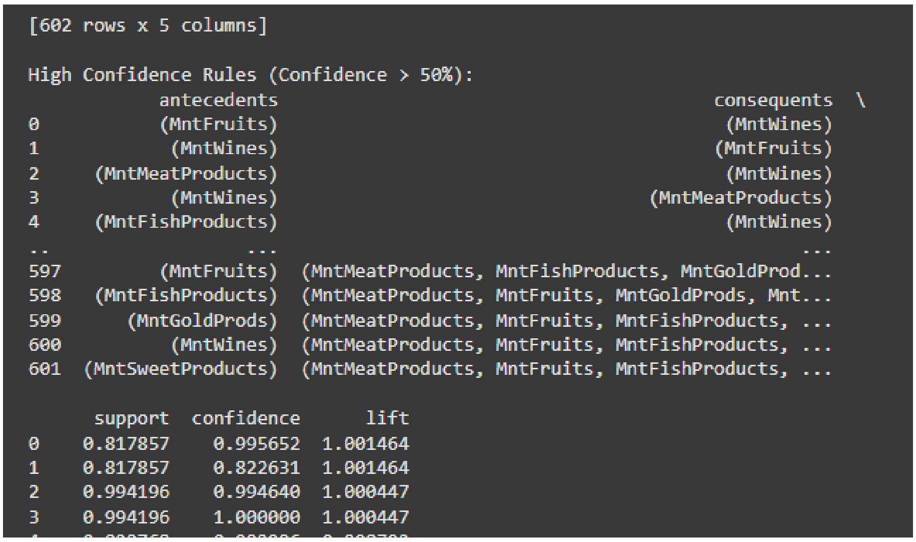
$$Lift = \frac{Support}{Support(X) \cdot Support(Y)}$$

ALGORITHM STEP:

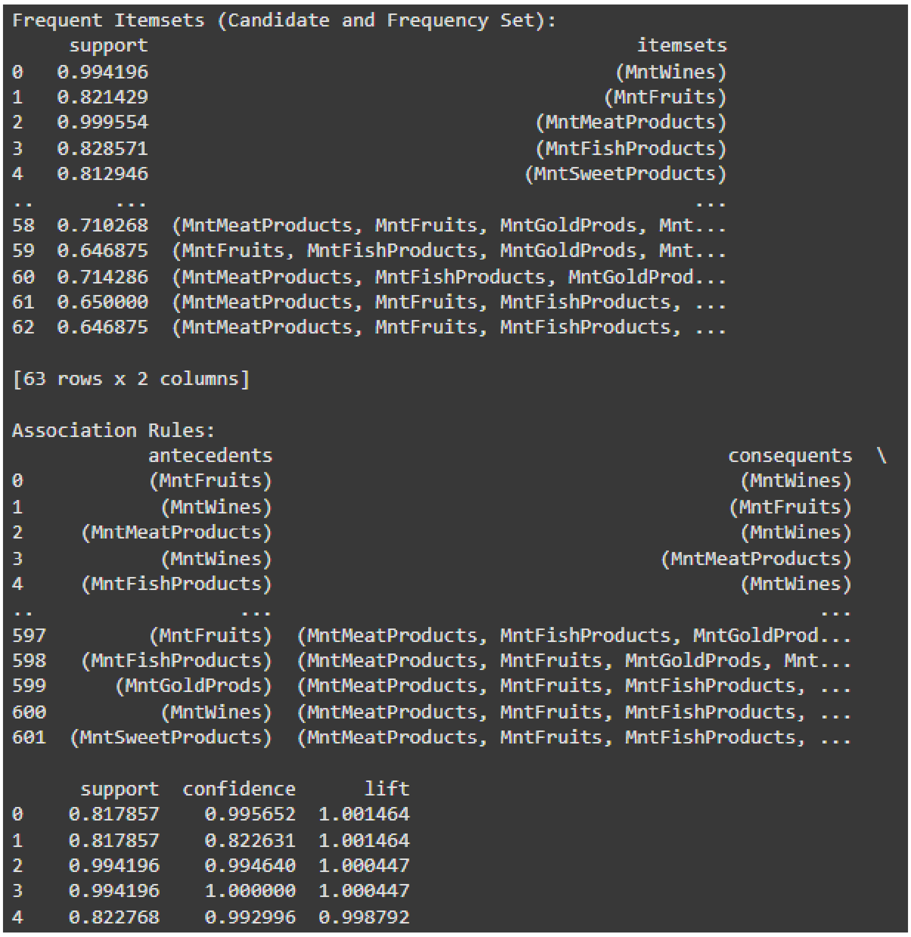
1: Select the minimum support and confidence.
2: Calculate support for itemsets of size 1.
3: Apply the minimum support threshold and prune itemsets that do not meet the threshold.
4: Move on to itemsets of size 2 and repeat steps one and two.
5: Continue the same process until no additional itemsets satisfying the minimum threshold can be found.

PROGRAM:

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules

# Step 1: Load the dataset
data = pd.read_csv('/content/customer_segmentation.csv')

# Step 2: Convert spending columns to boolean (True if spent > 0, otherwise False)
# Here, each column related to product spending will be treated as an item in the transaction
items = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']
data[items] = data[items].apply(lambda x: x > 0)  # Converts directly to boolean (True/False)

# Step 3: Use only the boolean columns for the apriori algorithm
df = data[items]

# Step 4: Generate frequent itemsets with a minimum support of 8%
frequent_itemsets = apriori(df, min_support=0.08, use_colnames=True)

# Display the candidate set and frequency set for each iteration
print("Frequent Itemsets (Candidate and Frequency Set):")
print(frequent_itemsets)

# Step 5: Generate association rules with a minimum confidence of 50%
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)

# Display the association rules
print("\nAssociation Rules:")
print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])

# Step 6: Find rules with confidence greater than 50%
high_confidence_rules = rules[rules['confidence'] > 0.5]

# Display high confidence rules
print("\nHigh Confidence Rules (Confidence > 50%):")
print(high_confidence_rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```



```
[602 rows x 5 columns]

High Confidence Rules (Confidence > 50%):
            antecedents                                        consequents  \
0           (MntFruits)                                        (MntWines)
1            (MntWines)                                       (MntFruits)
2     (MntMeatProducts)                                        (MntWines)
3            (MntWines)                                 (MntMeatProducts)
4     (MntFishProducts)                                        (MntWines)
..                  ...                                               ...
597         (MntFruits)  (MntMeatProducts, MntFishProducts, MntGoldProd...
598   (MntFishProducts)  (MntMeatProducts, MntFruits, MntGoldProds, Mnt...
599      (MntGoldProds)  (MntMeatProducts, MntFruits, MntFishProducts, ...
600          (MntWines)  (MntMeatProducts, MntFruits, MntFishProducts, ...
601  (MntSweetProducts)  (MntMeatProducts, MntFruits, MntFishProducts, ...

      support  confidence      lift
0    0.817857    0.995652  1.001464
1    0.817857    0.822631  1.001464
2    0.994196    0.994640  1.000447
3    0.994196    1.000000  1.000447
```



```
Frequent Itemsets (Candidate and Frequency Set):
     support                                          itemsets
0   0.994196                                        (MntWines)
1   0.821429                                       (MntFruits)
2   0.999554                                 (MntMeatProducts)
3   0.828571                                 (MntFishProducts)
4   0.812946                                (MntSweetProducts)
..       ...                                               ...
58  0.710268  (MntMeatProducts, MntFruits, MntGoldProds, Mnt...
59  0.646875  (MntFruits, MntFishProducts, MntGoldProds, Mnt...
60  0.714286  (MntMeatProducts, MntFishProducts, MntGoldProd...
61  0.650000  (MntMeatProducts, MntFruits, MntFishProducts, ...
62  0.646875  (MntMeatProducts, MntFruits, MntFishProducts, ...

[63 rows x 2 columns]

Association Rules:
            antecedents                                        consequents  \
0           (MntFruits)                                        (MntWines)
1            (MntWines)                                       (MntFruits)
2     (MntMeatProducts)                                        (MntWines)
3            (MntWines)                                 (MntMeatProducts)
4     (MntFishProducts)                                        (MntWines)
..                  ...                                               ...
597         (MntFruits)  (MntMeatProducts, MntFishProducts, MntGoldProd...
598   (MntFishProducts)  (MntMeatProducts, MntFruits, MntGoldProds, Mnt...
599      (MntGoldProds)  (MntMeatProducts, MntFruits, MntFishProducts, ...
600          (MntWines)  (MntMeatProducts, MntFruits, MntFishProducts, ...
601  (MntSweetProducts)  (MntMeatProducts, MntFruits, MntFishProducts, ...

      support  confidence      lift
0    0.817857    0.995652  1.001464
1    0.817857    0.822631  1.001464
2    0.994196    0.994640  1.000447
3    0.994196    1.000000  1.000447
4    0.822768    0.992996  0.998792
```

RESULT:

The above code is executed successfully using Association Rule Learning Using Apriori Algorithm.