PART A

(PART A : TO BE REFFERED BY STUDENTS)

Experiment No.09

Aim: **Implementation of Association Rule Mining Algorithm (Apriori) using Java/Python.**

Outcome:

After successful completion of this experiment students will be able to

1. Demonstrate an understanding of the importance of data mining
2. Organize and Prepare the data needed for data mining using pre preprocessing techniques
3. Perform exploratory analysis of the data to be used for mining.
4. Implement the appropriate data mining methods like Frequent Pattern mining on large data sets.

Theory:

Apriori Algorithm

The apriori algorithm has become one of the most widely used algorithms for frequent itemset mining and association rule learning. It has been applied to a variety of applications, including market basket analysis, recommendation systems, and fraud detection, and has inspired the development of many other algorithms for similar tasks.

Algorithm Details

The apriori algorithm starts by setting the minimum support threshold. This is the minimum number of times an item must occur in the database in order for it to be considered a frequent itemset. The algorithm then filters out any candidate itemsets that do not meet the minimum support threshold.

The algorithm then generates a list of all possible combinations of frequent itemsets and counts the number of times each combination appears in the database. The algorithm then generates a list of association rules based on the frequent itemset combinations.

An association rule is a statement of the form "if item A is present in a transaction, then item B is also likely to be present". The strength of the association is measured using the confidence of the rule, which is the probability that item B is present given that item A is present.

The algorithm then filters out any association rules that do not meet a minimum confidence threshold. These rules are referred to as strong association rules. Finally, the algorithm then returns the list of strong association rules as output.

Apriori uses a "bottom-up" approach, starting with individual items and gradually combining them into larger and larger itemsets as it searches for frequent patterns. It also uses a "delete-relabel" approach to efficiently prune the search space by eliminating infrequent itemsets from consideration.

Metrics for Evaluating Association Rules

In association rule mining, several metrics are commonly used to evaluate the quality and importance of the discovered association rules.

These metrics can be used to evaluate the quality and importance of association rules and to select the most relevant rules for a given application. It is important to note that the appropriate choice of metric will depend on the specific goals and requirements of the application.

Interpreting the results of association rule mining metrics requires understanding the meaning and implications of each metric, as well as how to use them to evaluate the quality and importance of the discovered association rules. Here are some guidelines for interpreting the results of the main association rule mining metrics:

Support

Support is a measure of how frequently an item or itemset appears in the dataset. It is calculated as the number of transactions containing the item(s) divided by the total number of transactions in the dataset. High support indicates that an item or itemset is common in the dataset, while low support indicates that it is rare.

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

Confidence

Confidence is a measure of the strength of the association between two items. It is calculated as the number of transactions containing both items divided by the number of transactions containing the first item. High confidence indicates that the presence of the first item is a strong predictor of the presence of the second item.

$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

Lift

Lift is a measure of the strength of the association between two items, taking into account the frequency of both items in the dataset. It is calculated as the confidence of the association divided by the support of the second item. Lift is used to compare the strength of the association between two items to the expected strength of the association if the items were independent.

A lift value greater than 1 indicates that the association between two items is stronger than expected based on the frequency of the individual items. This suggests that the association may be meaningful and worth further investigation. A lift value less than 1 indicates that the association is weaker than expected and may be less reliable or less significant.

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

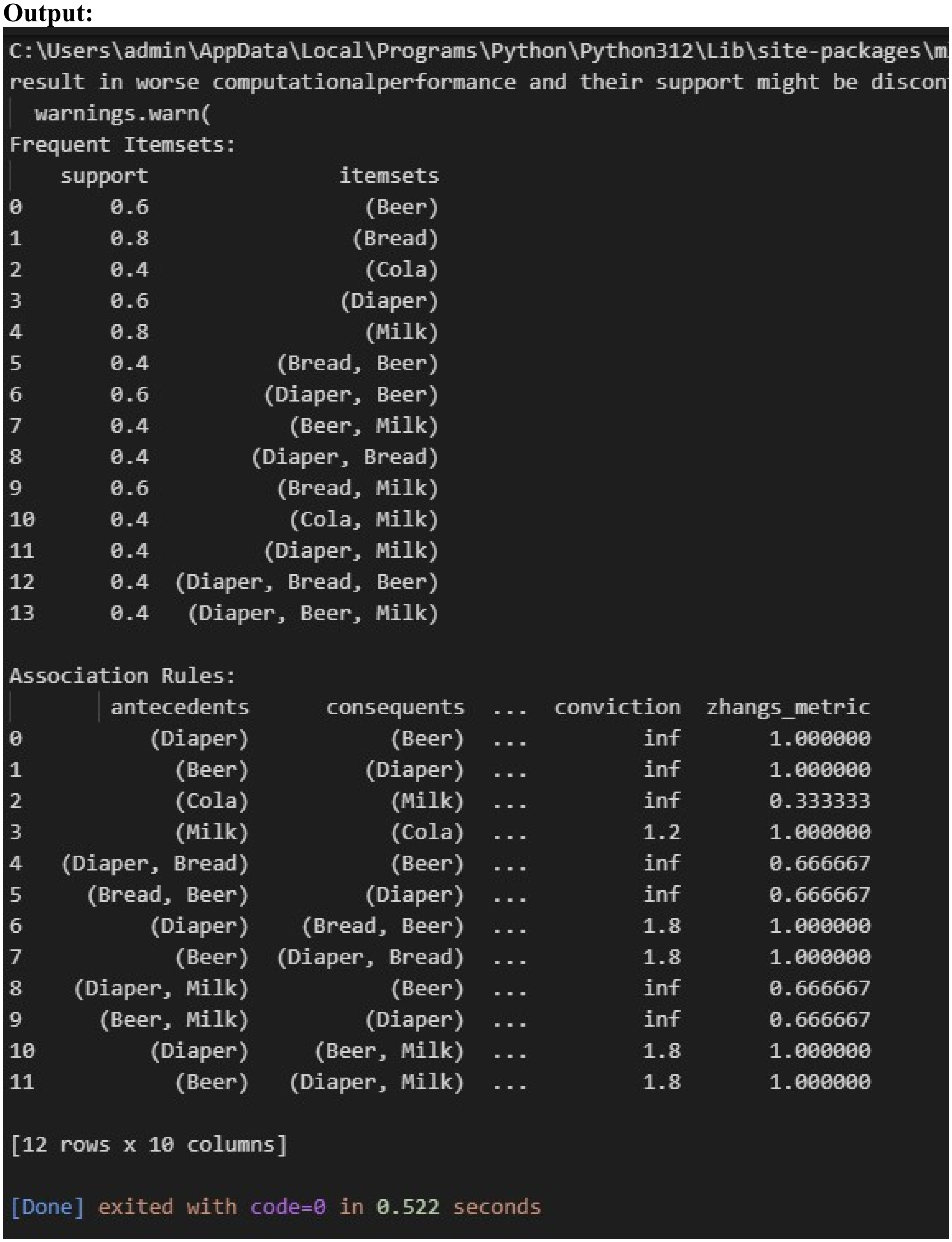
(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded at the end of the practical)

Roll. No: A48	Name: Vedang kajari
Class: T.E(AI&DS)	Batch: A3
Date of Experiment: 07/10/2024	Date of Submission: 10/10/2024
Grade:	

Input and Output:

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

# Sample
transaction data
data = {
    'TransactionID': [1, 2, 3, 4, 5],
    'Items': [['Bread', 'Milk'],
              ['Bread', 'Diaper', 'Beer', 'Eggs'],
              ['Milk', 'Diaper', 'Beer', 'Cola'],
              ['Bread', 'Milk', 'Diaper', 'Beer'],
              ['Bread', 'Milk', 'Cola']]
}

# Convert to
DataFrame df =
pd.DataFrame(data
)

# One-hot encoding of items
oht = df['Items'].str.join('|').str.get_dummies()

# Generate frequent itemsets
frequent_itemsets = apriori(oht, min_support=0.4, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
```

Output:



```
C:\Users\admin\AppData\Local\Programs\Python\Python312\Lib\site-packages\m
result in worse computationalperformance and their support might be discon
  warnings.warn(
Frequent Itemsets:
    support               itemsets
0       0.6                 (Beer)
1       0.8                (Bread)
2       0.4                 (Cola)
3       0.6               (Diaper)
4       0.8                 (Milk)
5       0.4          (Bread, Beer)
6       0.6         (Diaper, Beer)
7       0.4           (Beer, Milk)
8       0.4        (Diaper, Bread)
9       0.6          (Bread, Milk)
10      0.4           (Cola, Milk)
11      0.4         (Diaper, Milk)
12      0.4  (Diaper, Bread, Beer)
13      0.4   (Diaper, Beer, Milk)

Association Rules:
        antecedents      consequents  ...  conviction  zhangs_metric
0          (Diaper)           (Beer)  ...         inf       1.000000
1            (Beer)         (Diaper)  ...         inf       1.000000
2            (Cola)           (Milk)  ...         inf       0.333333
3            (Milk)           (Cola)  ...         1.2       1.000000
4   (Diaper, Bread)           (Beer)  ...         inf       0.666667
5     (Bread, Beer)         (Diaper)  ...         inf       0.666667
6          (Diaper)    (Bread, Beer)  ...         1.8       1.000000
7            (Beer)  (Diaper, Bread)  ...         1.8       1.000000
8    (Diaper, Milk)           (Beer)  ...         inf       0.666667
9      (Beer, Milk)         (Diaper)  ...         inf       0.666667
10         (Diaper)     (Beer, Milk)  ...         1.8       1.000000
11           (Beer)   (Diaper, Milk)  ...         1.8       1.000000

[12 rows x 10 columns]

[Done] exited with code=0 in 0.522 seconds
```

Observations and learning:

- **Pattern Discovery**: The Apriori algorithm is effective for discovering interesting relationships and patterns in large datasets, making it widely used in market basket analysis.
- **Support and Confidence**: Key metrics like support (how often itemsets appear) and confidence (how often items in a rule appear together) are fundamental for evaluating the strength of associations.
- **Efficiency**: The algorithm's performance relies heavily on the minimum support threshold. Higher thresholds reduce the number of candidate itemsets, improving efficiency, but may miss significant associations.
- **Data Preprocessing**: Proper data preprocessing, including cleaning and transforming data into the right format, is crucial for successful implementation of the Apriori algorithm.
- **Limitations**: Apriori can be computationally expensive for large datasets due to the need to generate all possible itemsets, leading to a preference for more advanced algorithms like FP-Growth in many applications.

Conclusion:

The Apriori algorithm is a fundamental technique in association rule mining, primarily used for uncovering patterns and relationships within large datasets. It excels in market basket analysis, enabling businesses to understand consumer behavior and make informed marketing decisions. Despite its effectiveness, the algorithm faces challenges such as computational inefficiency with large datasets and sensitivity to the choice of support thresholds.

Question of Curiosity:

1. What are the main metrics used to evaluate the strength of association rules in the Apriori algorithm?

A: The main metrics are support (how often itemsets appear) and confidence (the likelihood that items in a rule appear together).

2. What is a key limitation of the Apriori algorithm when dealing with large datasets?

A: A key limitation is its computational inefficiency due to the need to generate all possible itemsets, which can lead to long processing times.

3. In what application is the Apriori algorithm commonly used?

A: The Apriori algorithm is commonly used in market basket analysis to identify items frequently purchased together by customers.

****