Experiment No. 8

Expt No. 8	Implementation of Association Rule Mining
Date :	

Aim:

To implement and understand the working of association rule mining algorithm, used to discover interesting relationships between variables in large datasets in Machine Learning.

System Requirements:

- Operating System: Windows 8 or above / Linux / macOS
- Memory (RAM): Minimum 4 GB
- Processor: Minimum 2.33 GHz (Dual Core or higher)

Software/Tools Required:

Jupyter Notebook / Anaconda Navigator / Google Colaboratory / Spyder
Python 3.x with the following libraries:

Expected Outcomes:

- Understand the concept and fundamentals of association rule mining algorithm and its relevance to unsupervised learning
- To understand the importance of discovering meaningful patterns and correlations from transactional data
- To apply the algorithms to identify frequent itemsets and generate association rules
- To calculate and interpret the key metrics used in association rule mining.

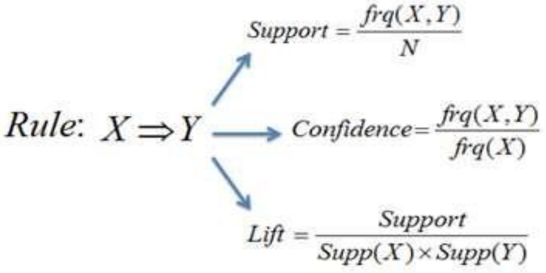
Theory:

The Apriori Algorithm, used for the first phase of the Association Rules, is the most popular and classical algorithm in the frequent old parts. These algorithm properties and data are evaluated with Boolean Association Rules. In this algorithm, there are product clusters that pass frequently, and then strong relationships between these products and other products are sought.

The importance of an Association Rules can be determined by 3 parameters that are used to identify the strength of the algorithm. Namely –

- Support
- Confidence
- Lift

Let X and Y represent the products in the market and N represents the total number of products.


$Support = \frac{frq(X,Y)}{N}$
$Rule: X \Rightarrow Y$
$Confidence = \frac{frq(X,Y)}{frq(X)}$
$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$

Support: It is the probability of an event to occur. Fraction of transactions that contain an itemset.

For example, the support of item I is defined as the number of transactions containing I divided by the total number of transactions

$$support(I) = \frac{Number\ of\ transactions\ containing\ I}{Total\ number\ of\ transactions}$$

Confidence: It is a measure of conditional probability
Measures how often items in Y appear in transactions that contain X
Confidence is the likelihood that item Y is also bought if item X is bought. It's calculated as the number of transactions containing X and Y divided by the number of transactions containing X.

$$confidence(X \rightarrow Y) = \frac{Number\ of\ transactions\ containing\ X\ and\ Y}{Number\ of\ transactions\ containing\ X}$$

Lift: It is the probability of all items occurring together divided by the product of antecedent and consequent occurring as if they are independent of each other

These measures show how much more often the antecedent and consequence of the rule occur together than we would expect if they were statistically independent (This measures the strength of the association between X and Y.). A lift value greater than 1 implies that the antecedent and consequent are dependent. (Lift evaluates the strength of the rule over the randomness. A lift greater than 1 indicates a strong rule)

Concept of Itemset –

n 'item' is a collection of one or more items found within a dataset. For example, consider a dataset containing various groceries. An item could be a combination like {Cheese, Tomato}

The 'length' of an item set is the number of items it contains. Thus, {Cheese, Tomato} is a 2-itemset.

- Single item itemsets: {Milk}, {Bread}, {Butter}, {Diapers}, {Sauce}, {Cola}
- Two-item itemsets: {Milk, Bread}, {Bread, Butter}, {Diapers, Sauce}, etc.
- Three-item itemsets: {Milk, Bread, Butter}, {Bread, Diapers, Sauce}, etc

Procedure to implement association rule mining algorithm:

1. **Data Collection:** Use a sample dataset of market transactions. For this experiment, use a grocery store dataset where each row represents a customer's purchase.
2. **Data Preprocessing:** Convert the dataset into a transactional format, such as a list of items purchased in each transaction.
3. **Implementing association rule mining:** Use the apriori function from the library to find frequent itemsets with minimum support.
4. **Generate Association Rules:** Use the association_rules function to generate rules from the frequent itemsets with thresholds for confidence and lift.
5. **Evaluate and Analyze results:** Interpret the generated rule and analyze relationships among items

Sample Code:

```
# Suppress DeprecationWarnings (including those from jupyter_client)

import warnings
warnings.filterwarnings('ignore', category=DeprecationWarning)

# import necessary libraries

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



# Step 1: Data Collection

data = {'TID': [1, 2, 3, 4, 5],
        'Milk': [1, 0, 1, 1, 0],
        'Bread': [1, 1, 1, 0, 1],
        'Butter': [0, 1, 0, 1, 1],
        'Cheese': [1, 0, 1, 1, 0]}

# Creating DataFrame and setting TID as index

df = pd.DataFrame(data).set_index('TID')



# Convert 0/1 to boolean (True/False) to avoid the warning

df = df.astype(bool)



print("Dataset:\n", df)




# Step 2: Apply Apriori Algorithm
```

```
frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)
print("\nFrequent Itemsets:\n", frequent_itemsets)


# Step 3: Generate Association Rules

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

# Only keep the essential columns

rules = rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]

print("\nAssociation Rules:\n", rules)


# Step 4: Analyzing Results
# Filter for strong rules with Lift > 1

strong_rules = rules[rules['lift'] > 1]

print("\nStrong Rules (Lift > 1):\n", strong_rules)
```

Observations -

- List the frequent itemsets and their respective support values
- Analyze the association rules, noting the confidence and lift values for each rule
- Identify strong rules based on high confidence and lift

Real-World Applications of Apriori algorithm:

1. Market Basket Analysis: Used by retailers to identify products that are frequently purchased together, enabling better product placement and promotions.
2. Recommendation Systems: E-commerce platforms (e.g., Amazon) use Apriori to recommend products based on customer purchase patterns, increasing sales.
3. Fraud Detection: Financial institutions use Apriori to identify suspicious transaction patterns that may indicate fraudulent behavior.

4. Healthcare: The algorithm helps in discovering patterns in medication prescriptions, aiding in drug interaction detection and improving patient safety.

Conclusion –

Hence, the model summarizes the key relationships discovered from the dataset and discuss practical applications such as Market Basket Analysis, Recommendation Systems or Inventory Management.

Viva Questions

Sr.No	Question	CO
1.	What is association rule mining?	**1ICPC406_4**
2.	Explain the concepts of "support", "confidence" and "lift" in association rule mining?	**1ICPC406_4**
3.	What is the difference between frequent itemsets and association rules?	**1ICPC406_4**
4.	Why is support important in association rule mining?	**1ICPC406_4**
5.	What is Apriori algorithm and how does it work?	**1ICPC406_4**
6.	Explain the limitations of the Apriori algorithm	**1ICPC406_4**
7.	What are some common applications of association rule mining in real-world scenarios?	**1ICPC406_4**
8.	What is the purpose of association rule mining?	**1ICPC406_4**
9.	How does association rule mining help in data analysis?	**1ICPC406_4**
10.	What is the difference between frequent itemsets and association rules? How are they related?	**1ICPC406_4**

References –

a. Textbook –

i. Machine Learning with Python – An approach to Applied ML – Abhishek Vijayvargiya, BPB Publications, 1st Edition 2018

ii. Machine Learning, Tom Mitchell, McGraw Hill Education, 1st Edition 1997

b. Online references –

i. https://www.kaggle.com/code/mervetorkan/association-rules-with-python

ii. https://github.com/ymoch/apyori

iii. https://thinkingneuron.com/how-to-do-association-rule-mining-using-apriori-in-python/