**Name:-Ansari Nooras Fatima Roll No:- 16co01**

**Exp No:- 06 Batch No:- 01**

**Aim:- Implementation of Association Rule Mining algorithm(Apriori).**

**Theory:-**

**Apriori Algorithm:-**

Apriori algorithm, a classic algorithm, is useful in mining frequent itemsets and relevant association rules. Usually, you operate this algorithm on a database containing a large number of transactions. One such example is the items customers buy at a supermarket.

It helps the customers buy their items with ease, and enhances the sales performance of the departmental store.

This algorithm has utility in the field of healthcare as it can help in detecting adverse drug reactions (ADR) by producing association rules to indicate the combination of medications and patient characteristics that could lead to ADRs. Three significant components comprise the apriori algorithm. They are as follows.

* Support
* Confidence
* Lift

This example will make things easy to understand.

As mentioned earlier, you need a big database. Let us suppose you have 2000 customer transactions in a supermarket. You have to find the Support, Confidence, and Lift for two items, say bread and jam. It is because people frequently bundle these two items together.

Out of the 2000 transactions, 200 contain jam whereas 300 contain bread. These 300 transactions include a 100 that includes bread as well as jam. Using this data, we shall find out the support, confidence, and lift.

#### **Support**

Support is the default popularity of any item. You calculate the Support as a quotient of the division of the number of transactions containing that item by the total number of transactions. Hence, in our example,

Support (Jam) = (Transactions involving jam) / (Total Transactions)

= 200/2000 = 10%

#### **Confidence**

In our example, Confidence is the likelihood that customer bought both bread and jam. Dividing the number of transactions that include both bread and jam by the total number of transactions will give the Confidence figure.

Confidence = (Transactions involving both bread and jam) / (Total Transactions involving jam)

                    = 100/200 = 50%

It implies that 50% of customers who bought jam bought bread as well.

#### **Lift**

According to our example, Lift is the increase in the ratio of the sale of bread when you sell jam. The mathematical formula of Lift is as follows.

Lift = (Confidence (Jam͢͢ – Bread)) / (Support (Jam))

      = 50 / 10 = 5

It says that the likelihood of a customer buying both jam and bread together is 5 times more than the chance of purchasing jam alone. If the Lift value is less than 1, it entails that the customers are unlikely to buy both the items together. Greater the value, the better is the combination.

**Algorithm:-**

Below are the apriori algorithm steps:

1. Scan the transaction data base to get the support ‘S’ each 1-itemset, compare ‘S’ with min\_sup, and get a support of 1-itemsets,
2. Use join to generate a set of candidate k-item set. Use apriori property to prune the unfrequented k-item sets from this set.
3. Scan the transaction database to get the support ‘S’ of each candidate k-item set in the given set, compare ‘S’ with min\_sup, and get a set of frequent k-item set
4. If the candidate set is NULL, for each frequent item set 1, generate all nonempty subsets of 1.
5. For every nonempty subsets of 1, output the rule “s=>(1-s)” if confidence C of the rule “s=>(1-s)” min\_conf
6. If the candidate set is not NULL, go to step 2.

**Program:-**

**#apriori.py**

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#Importing the dataset

dataset = pd.read\_csv('Market\_Basket\_Optimisation.csv', header = None)

transactions = []

for i in range(0,7501):

transactions.append([str(dataset.values[i,j]) for j in range(0,20)])

#training apriorion the dataset

from apyori import apriori

rules = apriori(transactions, min\_support = 0.003,min\_confidence = 0.2,min\_lift = 3,min\_length = 2)

#visualising results

results = list(rules)

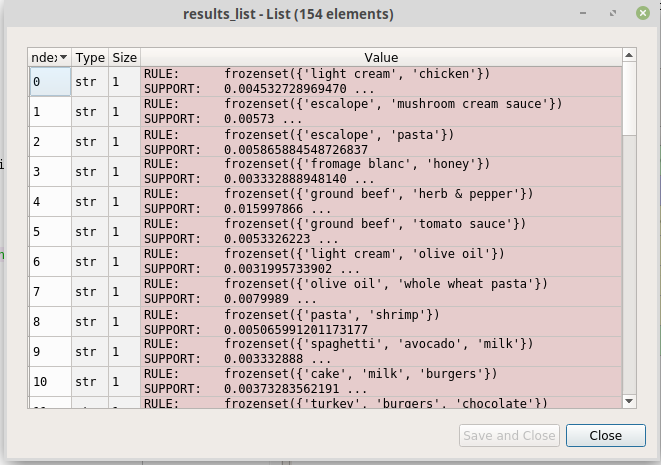
results\_list = []

for i in range(0, len(results)):

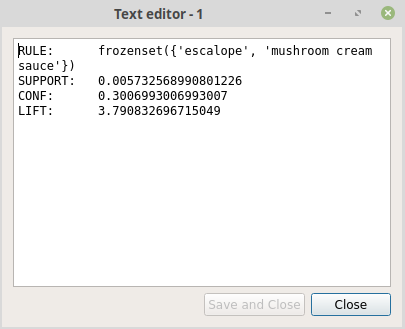
results\_list.append('RULE:\t' + str(results[i][0]) + '\nSUPPORT:\t' + str(results[i][1]) + '\nCONF:\t' + str(results[i][2][0][2]) + '\nLIFT:\t' + str(results[i][2][0][3]))

**Output:-**

**#getting the list of result\_list.**



**#selecting one rule.**



**Conclusion:-**

In this experiment, we have seen the Association mining algorithm and have seen various definition related to this such as support, confidence and lift and we have learned how apriori algorithm works.