**Lab 2**

**Association rule mining**

1. Write a program to find frequent itemsets and strong association rules from the sample dataset of your own by using Apriori algorithm.

Finding frequent itemsets and strong association rules is a key task in data mining, used to uncover hidden patterns in large datasets. The Apriori algorithm is a popular method for mining frequent itemsets, which can then be used to generate association rules. This program applies the Apriori algorithm to a sample dataset to identify the most frequent itemsets and derive meaningful rules that describe relationships between items.

Program:

import numpy as np

import pandas as pd

from apyori import apriori

data\_frame = pd.read\_csv('market\_basket.csv', header =None)

data\_frame.head()

lsts = []

for i in range (0,22):

lsts.append([str(data\_frame.values[i,j]) for j in range (0,6)])

asscsn\_rules = apriori(lsts, min\_support =0.50, min\_confidence = 0.7, min\_lift = 1.2, min\_length = 2)

asscsn\_results = list(asscsn\_rules)

formatted\_rules = []

for result in asscsn\_results:

for ordered\_stat in result.ordered\_statistics:

formatted\_rules.append({

'Base': ', '.join(list(ordered\_stat.items\_base)),

'Add': ', '.join(list(ordered\_stat.items\_add)),

'Support': result.support,

'Confidence': ordered\_stat.confidence,

'Lift': ordered\_stat.lift

})

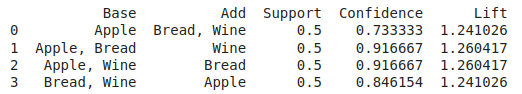
# Convert to a DataFrame for better readability

rules\_df = pd.DataFrame(formatted\_rules)

print(rules\_df)

Explanation:

This code performs market basket analysis using the Apriori algorithm to discover association rules from a transaction dataset. It starts by loading the dataset (market\_basket.csv) and converts it into a list of lists, where each list represents a transaction. The Apriori algorithm is applied with specified parameters: a minimum support of 50%, minimum confidence of 70%, and minimum lift of 1.2, considering itemsets of at least two items. The resulting association rules are formatted into a readable structure, including base items, added items, support, confidence, and lift values. Finally, the results are displayed as a DataFrame for easy interpretation of the discovered rules.

Output:  


The output displays association rules extracted from market basket data using the Apriori algorithm, which identifies relationships between items based on support, confidence, and lift. The "Base" column lists the item(s) found in the antecedent (left-hand side), while the "Add" column shows the item(s) in the consequent (right-hand side) of the rule. Support represents the proportion of transactions containing the items in the rule, confidence indicates the likelihood of the consequent being bought when the base items are bought, and lift measures the strength of the association (higher lift implies a stronger association). For example, the first rule indicates that when "Apple" is bought, there's a 73.33% chance that "Bread" and "Wine" will also be purchased together, with a lift value of 1.24, suggesting a slight positive correlation.

1. Write a program to find frequent itemsets and strong association rules from the sample dataset of your own by using FP growth algorithm.

The FP-growth algorithm is an efficient method for discovering frequent itemsets and strong association rules in large datasets. Unlike the Apriori algorithm, FP-growth uses a tree structure to compress the dataset and mine frequent itemsets without generating candidate sets. This program implements the FP-growth algorithm on a sample dataset to uncover frequent itemsets and generate association rules that highlight relationships between items.

Program:

import pandas as pd

from mlxtend.frequent\_patterns import fpgrowth, association\_rules

from mlxtend.preprocessing import TransactionEncoder

data = pd.read\_csv('market\_basket.csv', header=None)

transactions = []

for i in range(len(data)):

transactions.append([str(data.values[i, j]) for j in range(data.shape[1]) if str(data.values[i, j]) != 'nan'])

num\_transactions = data.shape[0]

# Use TransactionEncoder to convert the list of transactions into a format suitable for the FP-Growth algorithm.

te = TransactionEncoder()

te\_ary = te.fit(transactions).transform(transactions)

df = pd.DataFrame(te\_ary, columns=te.columns\_)

frequent\_items = fpgrowth(df, min\_support=0.5, use\_colnames=True)

print(frequent\_items)

rules = association\_rules(frequent\_items, metric="confidence", min\_threshold=0.7, num\_itemsets=22)

rules = rules[rules['lift'] >= 1.2]

rules = rules[rules['antecedents'].apply(lambda x: len(x)) + rules['consequents'].apply(lambda x: len(x)) >= 2]

# Reset the index and start numbering from 1

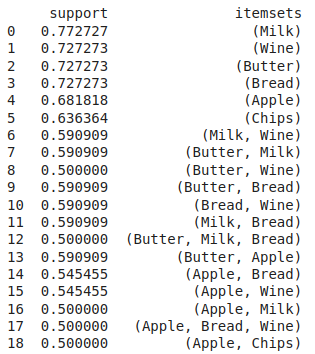
rules.reset\_index(drop=True, inplace=True)

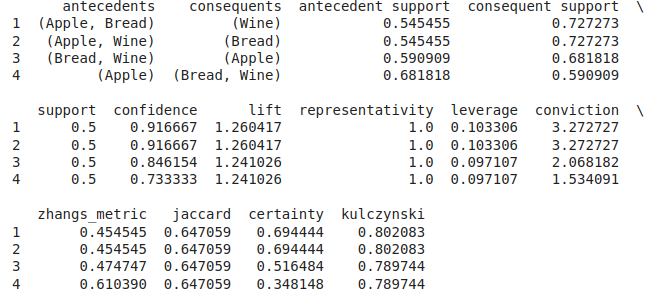
rules.index += 1 # Shift the index to start from 1

print(rules)

Explanation:  
This code performs market basket analysis using the FP-Growth algorithm to discover frequent itemsets and generate association rules. First, it loads the dataset (market\_basket.csv) and processes it into a list of transactions, where each transaction contains non-null items. The TransactionEncoder is then used to convert the transaction data into a binary matrix suitable for the FP-Growth algorithm. Frequent itemsets are generated with a minimum support of 50%, and association rules are derived using a minimum confidence of 70% and a lift threshold of 1.2. The rules are filtered to include only those with at least two items in both the antecedents and consequents, and the index is reset for clarity before printing the results.

Output:





The output shows four association rules derived from frequent itemsets, with each rule specifying relationships between items in transactions. The antecedents are the items found on the left-hand side of the rule, while the consequents are the items on the right-hand side. For instance, the first rule suggests that if a customer buys both "Apple" and "Bread," there is a 91.67% chance they will also purchase "Wine," with a lift of 1.26, indicating a positive association between these items. The confidence represents the probability that the consequent is purchased when the antecedent is bought, while lift measures the strength of the association. Support reflects the proportion of transactions containing the entire rule. Additional metrics such as leverage, conviction, and Zhang’s metric further evaluate the rule's strength and relevance, with Jaccard and certainty providing measures of similarity and certainty in the association.