Name: Jigar Siddhpura **SAPID:** 60004210155

DIV: C/C2 **Branch:** Computer Engineering

DMW - EXPERIMENT 6 - ASSOCIATION RULE MINING

Jigar Siddhpura
60004210155

DMW - Experiment 6

Aim: To implement association rule mining using:
1) Apriori
2) FP Tree

Theory: 1. Association rule learning is a type of unsupervised learning technique that checks for dependency of one data item on other & maps accordingly so that it can be profitable.

2. It is based on different rules to discover the interesting relations btw variables in the database.

3. It is employed in Market Basket Analysis, web usage mining, continuous production, etc. being one of the important concepts in Machine Learning.

4. We can understand it by taking an example of supermarket, as in a supermarket, all products that are purchased together are kept together.

5. For example, if a customer buys bread; he most likely buys butter, eggs or milk, so these are stored nearby.

6. Association rule mining can be div into 3 types of algos:
a) Apriori
b) F-P Growth algo
c) Eclat

7. Association rule learning works on the concept of If & Else statements, such as If A then B.

8. These type of relations where we can find out some association or relation btw 2 items are known as single cardinality.

9. If the no. of item increases, cardinality also increases.

10. To measure, the association rules btw lots of data, metrics

used are : a) Support b) Lift c) Confidence.

Apriori : 1. It is an array based algorithm
2. It uses join & prune algos technique
3. It uses BFS algo.
4. It utilizes level wise approach where it generates patterns containing 1 item, then 2, then 3 & so on.
5. Candidate generation is extremely slow. Runtime increases exp. depending on no. of diff. items.
6. Candidate generation is parallelization.
7. Requires large memory space.
8. Scan db multiple times

FP Growth : 1. It is a tree based algo.
2. It constructs conditional freq. pattern tree & conditional pattern base from db which satisfy min. sp.
3. It uses DFS algo.
4. It utilizes pattern growth approach i.e. only considers patterns actually existing in db.
5. Runtime increases linearly, depending on no. of transactions & items.
6. Data is very interdependent, each node needs the root.
7. Requires less memory ~~as~~ due to compact structure & no candidate generation.
8. Scans db only twice for constructing frequent pattern tree.

Part A:

Read min_support and confidence from the user

Program Apriori algorithm using inbuilt functions.

Print the association rules

Code:

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder

df  =
pd.read_csv('/content/gdrive/MyDrive/DMW/datasets/GroceryStoreDataSet.csv',sep=',',names
=['products'])
df.head()

#one hot encoding
data = list(df['products'].apply(lambda x:x.split(',')))
encoder = TransactionEncoder()
encoded_data = encoder.fit_transform(data)
df2 = pd.DataFrame(encoded_data,columns=encoder.columns_)
df2.replace(True,1,inplace=True)
df2.replace(False,0,inplace=True)

frq_items = apriori(df2,min_support=min_support,use_colnames=True)
rules = association_rules(frq_items,metric='confidence',min_threshold=min_conf)
print(f"Enter minimum support : 0.2")
print(f"Enter minimum confidence : 0.6")
rules
```

Output:

```
Enter minimum support : 0.2
Enter minimum confidence : 0.6
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.75	0.250000
1	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.05	0.035714
2	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80	0.571429
3	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.80	0.571429
4	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.25	0.750000

Part B:

Program FP tree using inbuilt functions for the following dataset

TID	*Items bought*
100	{*f, a, c, d, g, i, m, p*}
200	{*a, b, c, f, l, m, o*}
300	{*b, f, h, j, o, w*}
400	{*b, c, k, s, p*}
500	{*a, f, c, e, l, p, m, n*}

Print the frequent patterns generated.

Code:

```
from mlxtend.frequent_patterns.fpgrowth import fpgrowth

dataset = [['f', 'a', 'c', 'd', 'g', 'i', 'm', 'p'],
['a', 'b', 'c', 'f', 'l', 'm', 'o'],
['b', 'f', 'h', 'j', 'o', 'w'],
['b', 'c', 'k', 's', 'p'],
['a', 'f', 'c', 'e', 'l', 'p', 'm', 'n']]

encoder = TransactionEncoder()
encoded_data = encoder.fit_transform(dataset)
fp_df = pd.DataFrame(encoded_data,columns=encoder.columns_)

pattern = fpgrowth(fp_df,min_support=min_conf_fp,use_colnames=True)
print(f"FPTree with minimum confidence = {min_conf_fp*100}%")
Pattern

rules = association_rules(pattern,metric='confidence',min_threshold=min_conf_fp)
print(f"Association rules are as follows")
rules
```

Output:

FPTree with minimum confidence = 60.0%

	support	itemsets
0	0.8	(f)
1	0.8	(c)
2	0.6	(p)
3	0.6	(m)
4	0.6	(a)
5	0.6	(b)
6	0.6	(c, f)
7	0.6	(p, c)
8	0.6	(c, m)
9	0.6	(f, m)
10	0.6	(c, f, m)
11	0.6	(a, m)
12	0.6	(a, c)
13	0.6	(a, f)
14	0.6	(a, m, c)
15	0.6	(a, f, m)
16	0.6	(a, f, c)
17	0.6	(a, m, c, f)

Association rules are as follows

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(c)	(f)	0.8	0.8	0.6	0.75	0.937500	-0.04	0.8	-0.25
1	(f)	(c)	0.8	0.8	0.6	0.75	0.937500	-0.04	0.8	-0.25
2	(p)	(c)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
3	(c)	(p)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
4	(c)	(m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
5	(m)	(c)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
6	(f)	(m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
7	(m)	(f)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
8	(c, f)	(m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
9	(c, m)	(f)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
10	(f, m)	(c)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
11	(c)	(f, m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
12	(f)	(c, m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00

13	(m)	(c, f)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
14	(a)	(m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
15	(m)	(a)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
16	(a)	(c)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
17	(c)	(a)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
18	(a)	(f)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
19	(f)	(a)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
20	(a, m)	(c)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
21	(a, c)	(m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
22	(c, m)	(a)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
23	(a)	(c, m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
24	(m)	(a, c)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
25	(c)	(a, m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00
26	(a, f)	(m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
27	(a, m)	(f)	0.6	0.8	0.6	1.00	1.250000	0.12	inf	0.50
28	(f, m)	(a)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
29	(a)	(f, m)	0.6	0.6	0.6	1.00	1.666667	0.24	inf	1.00
30	(f)	(a, m)	0.8	0.6	0.6	0.75	1.250000	0.12	1.6	1.00

Conclusion:

Implemented Apriori and algorithm for a market basket analysis dataset and made and FP Tree for the given dataset. Apriori is a Join-Based algorithm and FP-Growth is Tree-Based algorithm for frequent itemset mining or frequent pattern mining for market basket analysis.