Name: Adwait Hegde
Roll No: 2019130019
TE Comp (Batch-A)

EXPERIMENT 5

Aim:

To apply Apriori Algorithm to given dataset: Association Rule Mining with WEKA

Theory:

Association Mining is defined as finding patterns, associations, correlations, or casual structures among sets of items or objects in transaction dataset, relational database, and other information repositories. The association rule takes the form of if … then… statement of the form:

A => B (read as, if A then B)

Performance measures for association rules:

Support:

support (A => B)= P(A ∩ B)

The minimum percentage of instances in the database that contain all items listed in a given association rule.

$$\text{support } (A \Rightarrow B) = \frac{\text{number of instances containing both A and B}}{\text{Total Number of instances}}$$

Example:

5000 transaction contain milk and bread in a set of 50000

➔ Support=> 5,000/50,000=10%

Confidence:

confidence (A=> B) = P(B|A)

Given a rule of the form "if A then B", rule for confidence is the conditional probability that B is true when A is known to be True.

$$\text{confidence } (A \Rightarrow B) = \frac{\text{number of instances containing both A and B}}{\text{number of instances containing A}}$$

Example:

IF Customer purchases milk THEN they also purchase bread:

In a set of 50,000, there are 10,000 transactions that contain milk, and 5,000 of these contain also bread.

➔ Confidence => 5,000/10,000=50%

Task

Consider dataset “Groceries” and apply apriori algorithm on it. What are the first 5 rules generated when the min support is 0.001 (0.1%) and min confidence is 0.9 (90%)

Exercise 1: Basic association rule creation manually

The 'database' below has four transactions. What association rules can be found in this set, if the minimum support (i.e coverage) is 60% and the minimum confidence (i.e. accuracy) is 80% ?

Trans_id	Itemlist
T1	{K, A, D, B}
T2	{D, A C, E, B}
T3	{C, A, B, E}
T4	{B, A, D}

```
dflist = [
        [1, 1, 0, 1, 0, 1],
        [1, 1, 1, 1, 1, 0],
        [1, 1, 1, 0, 1, 0],
        [1, 1, 0, 1, 0, 0],
]

df1 = pd.DataFrame(dflist, columns=['A', 'B', 'C', 'D', 'E', 'K'])
df1
```

	A	B	C	D	E	K
0	1	1	0	1	0	1
1	1	1	1	1	1	0
2	1	1	1	0	1	0
3	1	1	0	1	0	0

```
[6] frq_items = apriori(df1, min_support = 0.6, use_colnames = True)
    print(frq_items)
    rules = association_rules(frq_items, metric ="confidence", min_threshold = 0.8)
    print(rules)
```

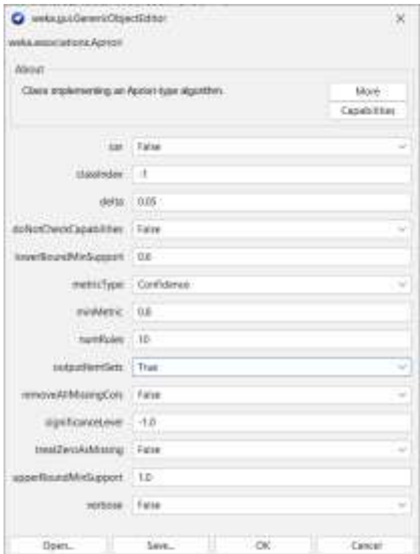
```
   support   itemsets
0     1.00        (A)
1     1.00        (B)
2     0.75        (D)
3     1.00     (A, B)
4     0.75     (A, D)
5     0.75     (D, B)
6     0.75  (A, D, B)
  antecedents consequents  antecedent support  consequent support  support  \
0         (A)         (B)                1.00                 1.0     1.00
1         (B)         (A)                1.00                 1.0     1.00
2         (D)         (A)                0.75                 1.0     0.75
3         (D)         (B)                0.75                 1.0     0.75
4      (A, D)         (B)                0.75                 1.0     0.75
5      (D, B)         (A)                0.75                 1.0     0.75
6         (D)      (A, B)                0.75                 1.0     0.75

   confidence  lift  leverage  conviction
0         1.0   1.0       0.0         inf
1         1.0   1.0       0.0         inf
2         1.0   1.0       0.0         inf
3         1.0   1.0       0.0         inf
4         1.0   1.0       0.0         inf
5         1.0   1.0       0.0         inf
6         1.0   1.0       0.0         inf
```

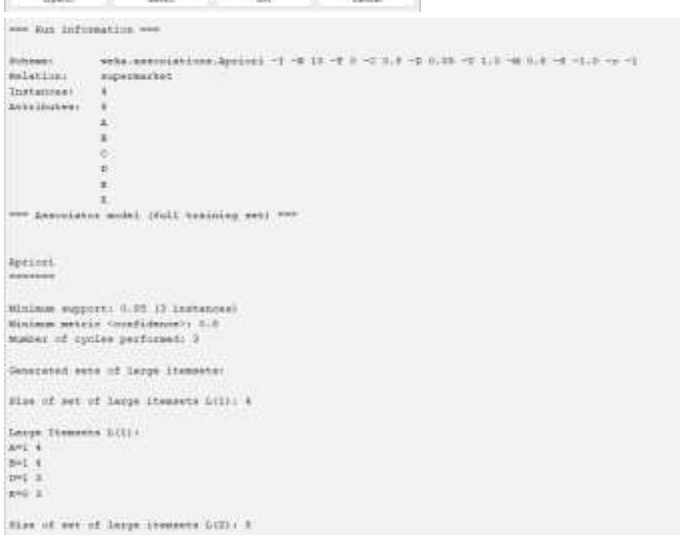
Association rules found:

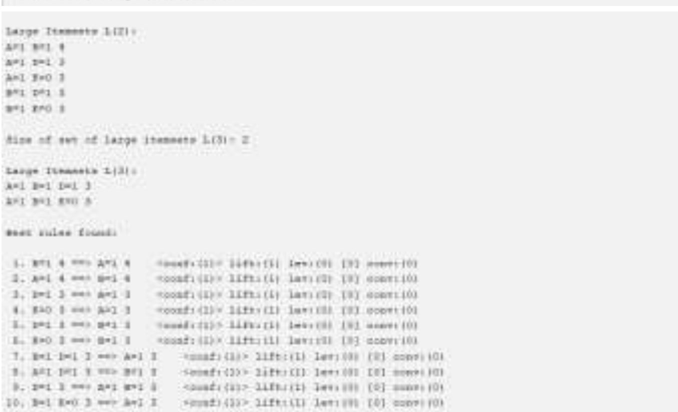
(A) => (B)
(B) => (A)
(D) => (A)
(D) => (B)
(D, B) => (A)
(D, A) => (B)
(D) => (B, A)

Hint: Make a tabular and binary representation of the data in order to better see the relationship between Items. First generate all item sets with minimum support of 60%. Then form rules and calculate their confidence base on the conditional probability P(B|A) = |B∩A| / |A|. Remember to only take the item sets from the previous phase whose support is 60% or more.

Exercise 2: Input file generation and Initial experiments with Weka's association rule discovery







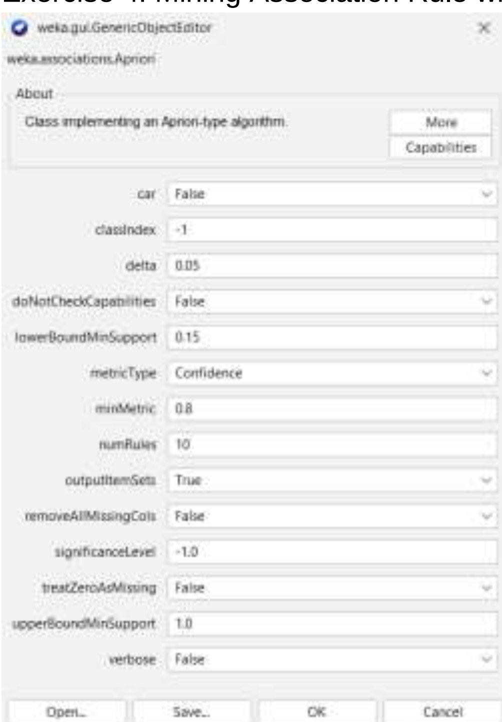
One can notice that the association rules we determined via the manual method are identical to those provided by WEKA. I made an .arff file and utilised it.

Exercise 3: Mining Association Rule with WEKA Explorer – Weather dataset

I used an open weather dataset to perform association rule mining.

It does not have a target property like a decision tree. Instead, it tries to link all of the columns together. The values of the play column in the decision tree are anticipated based on the values of the outlook, temp, humidity, and windy columns. The values of the play column are also taken into account in the association rule, and the remainder columns can be predicted.

Exercise 4: Mining Association Rule with WEKA Explorer – Vote



```
physician-fee-freeze=n 247
religious-groups-in-schools=y 272
anti-satellite-test-ban=y 239
aid-to-nicaraguan-contras=y 242
synfuels-corporation-cutback=n 264
education-spending=n 233
crime=y 248
duty-free-exports=n 233
export-administration-act-south-africa=y 269
Class=democrat 267

Size of set of large itemsets L(2): 4

Large Itemsets L(2):
adoption-of-the-budget-resolution=y physician-fee-freeze=n 219
adoption-of-the-budget-resolution=y Class=democrat 231
physician-fee-freeze=n Class=democrat 245
aid-to-nicaraguan-contras=y Class=democrat 218

Size of set of large itemsets L(3): 1

Large Itemsets L(3):
adoption-of-the-budget-resolution=y physician-fee-freeze=n Class=democrat 219

Best rules found:

 1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219    <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.50)
 2. physician-fee-freeze=n 247 ==> Class=democrat 245    <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8)
 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219    <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68)
 4. Class=democrat 267 ==> physician-fee-freeze=n 245    <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02)
 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231    <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25)
 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218    <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74)
 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219    <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8)
 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219    <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56)
 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219    <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99)
10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219    <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
```

Members of the Democratic Party are more prevalent than members of the Republican Party, which enhances the likelihood of their presence in the most often occurring item sets. As a result, there are no members of the republican party listed in the rules. If the number of Republic party members grows, we may see fewer entries in the rules.

Exercise 5: Let’s run Apriori on another real-world dataset.

```
=== Run information ===

Scheme:       weka.associations.Apriori -I -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.3 -S -1.0 -c -1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.3 (1388 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 25

Large Itemsets L(1):
bread and cake=t 3330
baking needs=t 2795
juice-sat-cord-ms=t 2463
biscuits=t 2605
canned vegetables=t 1577
breakfast food=t 1862
sauces-gravy-pkle=t 2201
confectionary=t 1690
frozen foods=t 2717
pet foods=t 1867
laundry needs=t 1563
party snack foods=t 2330
tissues-paper prd=t 2247
soft drinks=t 1888
cheese=t 1879
milk-cream=t 2939
```

```
Size of set of large itemsets L(3): 20

Large Itemsets L(3):
bread and cake=t baking needs=t biscuits=t 1456
bread and cake=t baking needs=t frozen foods=t 1405
bread and cake=t baking needs=t milk-cream=t 1580
bread and cake=t baking needs=t fruit=t 1564
bread and cake=t baking needs=t vegetables=t 1586
bread and cake=t biscuits=t frozen foods=t 1510
bread and cake=t biscuits=t milk-cream=t 1485
bread and cake=t biscuits=t fruit=t 1541
bread and cake=t biscuits=t vegetables=t 1487
bread and cake=t frozen foods=t milk-cream=t 1516
bread and cake=t frozen foods=t fruit=t 1548
bread and cake=t frozen foods=t vegetables=t 1548
bread and cake=t milk-cream=t fruit=t 1684
bread and cake=t milk-cream=t vegetables=t 1658
bread and cake=t fruit=t vegetables=t 1791
baking needs=t milk-cream=t vegetables=t 1392
baking needs=t fruit=t vegetables=t 1489
biscuits=t fruit=t vegetables=t 1404
frozen foods=t fruit=t vegetables=t 1451
milk-cream=t fruit=t vegetables=t 1571

Best rules found:

 1. biscuits=t vegetables=t 1764 ==> bread and cake=t 1487    <conf:(0.84)> lift:(1.17) lev:(0.05) [217] conv:(1.78)
 2. total=high 1679 ==> bread and cake=t 1413    <conf:(0.84)> lift:(1.17) lev:(0.04) [204] conv:(1.76)
 3. biscuits=t milk-cream=t 1767 ==> bread and cake=t 1485    <conf:(0.84)> lift:(1.17) lev:(0.05) [213] conv:(1.75)
 4. biscuits=t fruit=t 1837 ==> bread and cake=t 1541    <conf:(0.84)> lift:(1.17) lev:(0.05) [219] conv:(1.73)
 5. biscuits=t frozen foods=t 1810 ==> bread and cake=t 1510    <conf:(0.83)> lift:(1.16) lev:(0.04) [207] conv:(1.69)
 6. frozen foods=t fruit=t 1861 ==> bread and cake=t 1548    <conf:(0.83)> lift:(1.16) lev:(0.05) [208] conv:(1.66)
 7. frozen foods=t milk-cream=t 1826 ==> bread and cake=t 1516    <conf:(0.83)> lift:(1.15) lev:(0.04) [201] conv:(1.65)
 8. baking needs=t milk-cream=t 1907 ==> bread and cake=t 1580    <conf:(0.83)> lift:(1.15) lev:(0.04) [207] conv:(1.63)
 9. milk-cream=t fruit=t 2038 ==> bread and cake=t 1684    <conf:(0.83)> lift:(1.15) lev:(0.05) [217] conv:(1.61)
10. baking needs=t biscuits=t 1764 ==> bread and cake=t 1456    <conf:(0.83)> lift:(1.15) lev:(0.04) [196] conv:(1.6)
```

Implementation:

GitHub Link:
https://github.com/adwait-hegde/DataAnalytics-Lab/tree/main/Exp5

Conclusion:
The discovery of interesting associations and links among vast sets of data objects is made possible by association rule mining. This rule indicates how often a particular itemset appears in a transaction. We can find rules that forecast the occurrence of an item based on the occurrences of other items in the transaction given a set of transactions. We can use the Apriori algorithm to mine the frequent itemset and construct association rules between them. The biggest drawback is the amount of time it takes to hold a large number of candidate sets with frequent item sets, low minimum support, or huge item sets, making it inefficient for large datasets.