Name: Yash Brid

Roll No.: 2019130008

Batch: TE COMPS Batch A

Course Code: DA (Data Analytics)

Experiment No. 5

Aim: Apply Apriori Algorithm to given dataset. Perform Association Rule Mining with WEKA.

Theory:

Association Mining is defined as finding patterns, associations, correlations, or casual structuresamong 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 givenassociation rule.

support (A => B)= (number of instances containing both A and B/ 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 truewhen A is known to be True.

confidence (A => B)= (number of instances containing both A and B/ number of instances containing A)

Example:

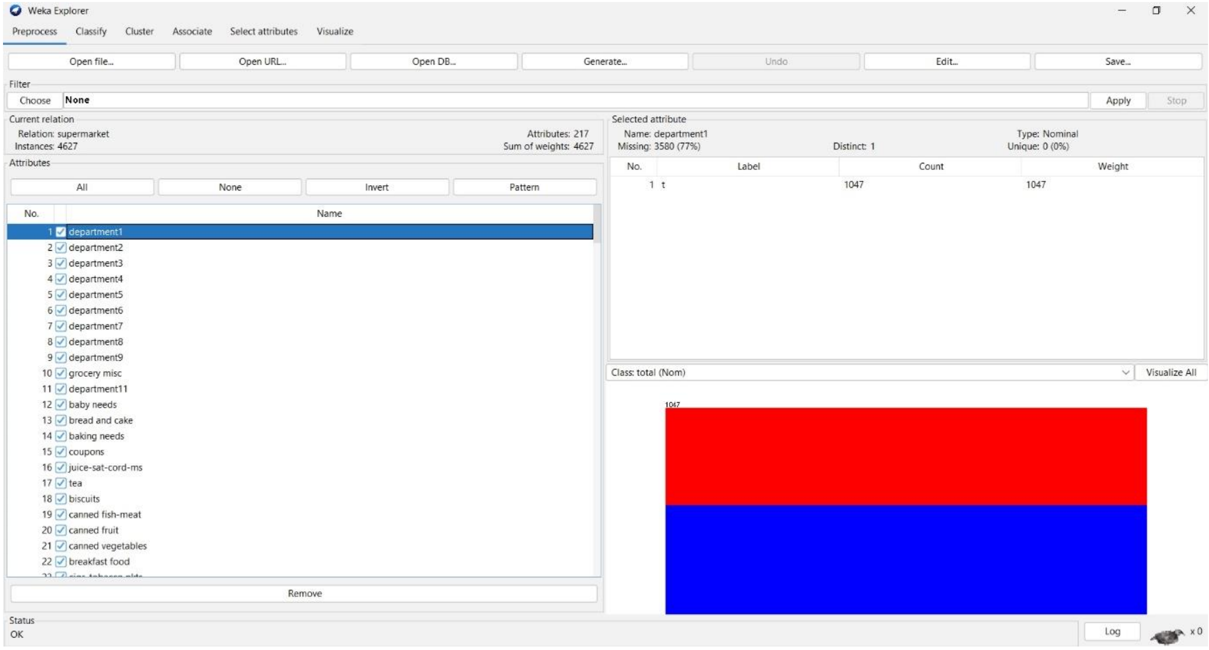
IF Customer purchases milk THEN they also purchase bread:

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

Confidence => 5,000/10,000=50%

Procedure:

1. Open the concerned CSV file in WEKA. It will look like this after opening



Filter: None

Current relation
Relation: supermarket
Instances: 4627
Attributes: 217
Sum of weights: 4627

Attributes

No.		Name
1	☑	department1
2	☑	department2
3	☑	department3
4	☑	department4
5	☑	department5
6	☑	department6
7	☑	department7
8	☑	department8
9	☑	department9
10	☑	grocery misc
11	☑	department11
12	☑	baby needs
13	☑	bread and cake
14	☑	baking needs
15	☑	coupons
16	☑	juice-sat-cord-ms
17	☑	tea
18	☑	biscuits
19	☑	canned fish-meat
20	☑	canned fruit
21	☑	canned vegetables
22	☑	breakfast food

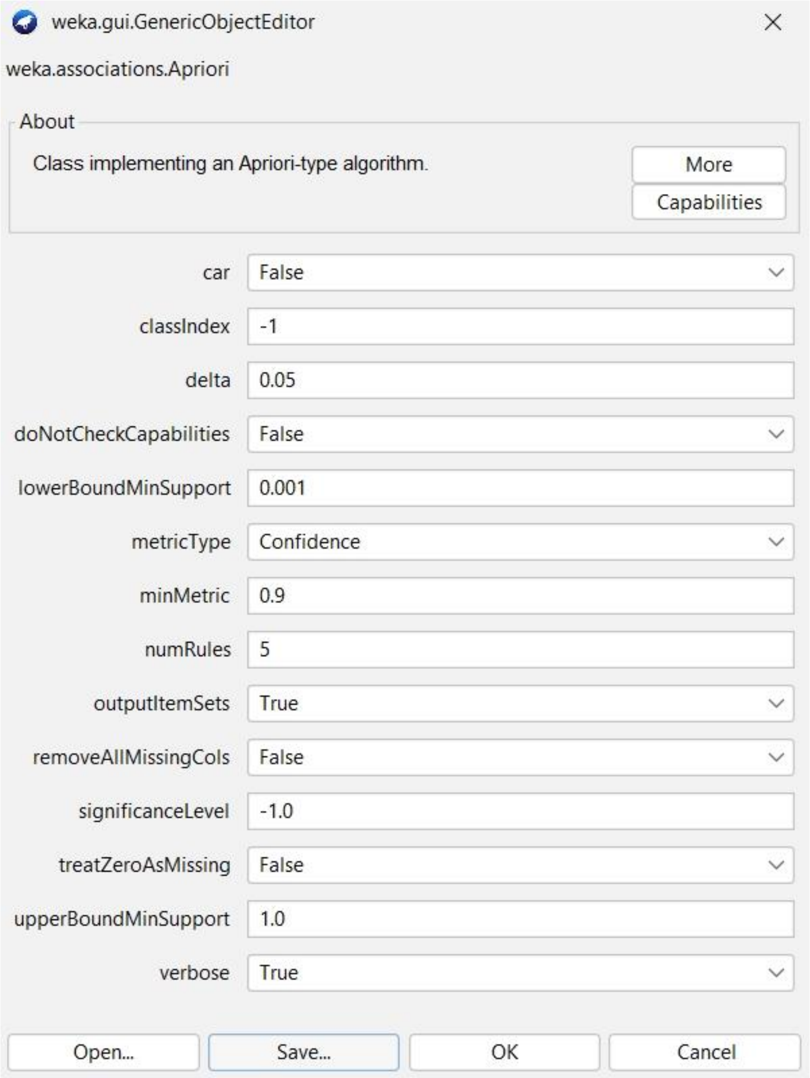
Selected attribute
Name: department1
Missing: 3580 (77%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	t	1047	1047

Class: total (Nom)

Status
OK

2. Click on Associate tab and select ‘Apriori’ from drop down appeared after clicking ‘Choose’ button
3. Click on Choose and select Apriori algorithm
4. Now double click apriori algorithm to open option menu pop up where you have to set up appropriate values.



weka.gui.GenericObjectEditor

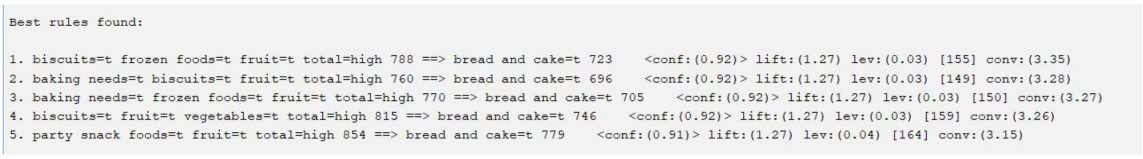
weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.001
metricType	Confidence
minMetric	0.9
numRules	5
outputItemSets	True
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	True

5. Now click start and WEKA will automatically process the data and return us the required output



```
Best rules found:

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
```

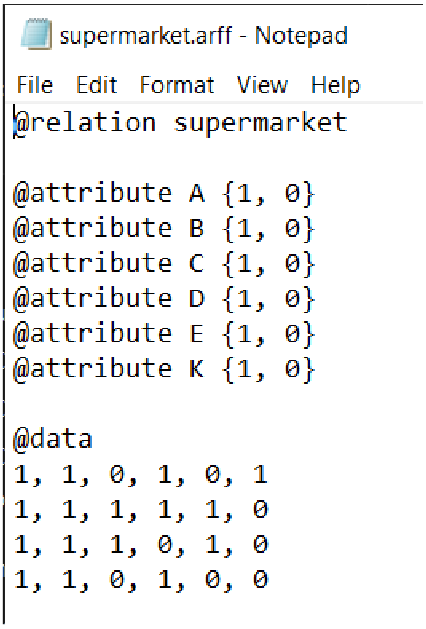
Here, the output of WEKA shows the 5 association rules. These, indicates that bread and cake are always bought by buyers when total is generally high. Rules state that if a buyer buys biscuits, frozen foods, fruits and total is high then it is very likely that buyer will buy bread and cake. If a buyer buys baking needs, biscuits, fruits and total is high then it is very likely that buyer will buy bread and cake. If a buyer buys baking needs, frozen foods, fruits and total is high then it is very likely that buyer will buy bread and cake. These are all frequently bought items.

Exercise 1: Basic association rule creation manually.

COLAB LINK:
https://colab.research.google.com/drive/115ufMm1D_ZYmremic1tjyP9DBspch-ax?usp=sharing

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

1. Create a .arff file for given dataset.

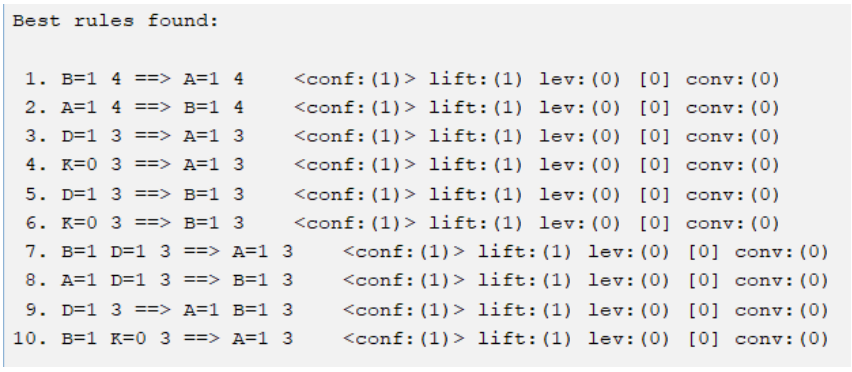


```
@relation supermarket

@attribute A {1, 0}
@attribute B {1, 0}
@attribute C {1, 0}
@attribute D {1, 0}
@attribute E {1, 0}
@attribute K {1, 0}

@data
1, 1, 0, 1, 0, 1
1, 1, 1, 1, 1, 0
1, 1, 1, 0, 1, 0
1, 1, 0, 1, 0, 0
```

2. Load into WEKA and perform association rule mining.

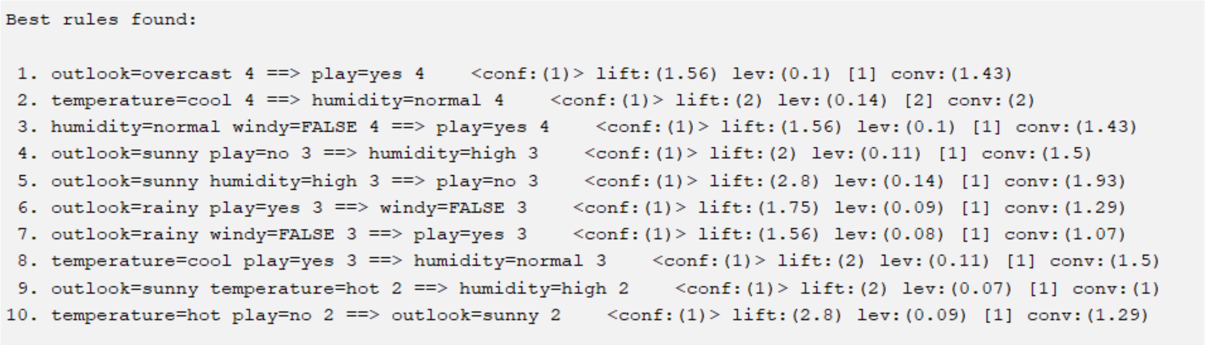


```
Best rules found:

 1. B=1 4 ==> A=1 4    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. A=1 4 ==> B=1 4    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. D=1 3 ==> A=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. K=0 3 ==> A=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. D=1 3 ==> B=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. K=0 3 ==> B=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. B=1 D=1 3 ==> A=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. A=1 D=1 3 ==> B=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. D=1 3 ==> A=1 B=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. B=1 K=0 3 ==> A=1 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

We observe that association rules that we determined using manual method, are exactly same as that of given by WEKA. I created and used a .arff file (See point 1)

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



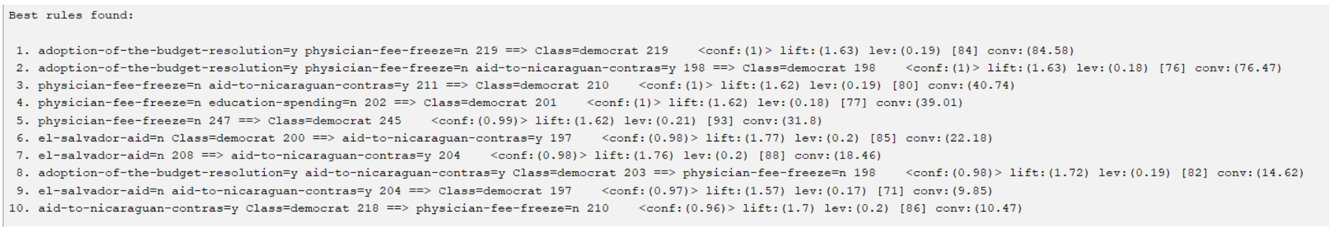
```
Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. outlook=sunny temperature=hot 2 ==> humidity=high 2    <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2    <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```

Performed association rule mining on the weather dataset that I found online. Unlike decision tree, it has no target attribute. Instead, it tries to associate all the columns. In decision tree, depending upon values of outlook, temp, humidity and windy columns, values of play column is predicted. In association rule, values of play column is also considered and rest columns can also be predicted.

For example, in rule 1, it states that if outlook is overcast then we can play. But in rule 4, it states that, if outlook is sunny and you are playing then humidity must be high.

Exercise 4: Mining Association Rule with WEKA Explorer – Vote



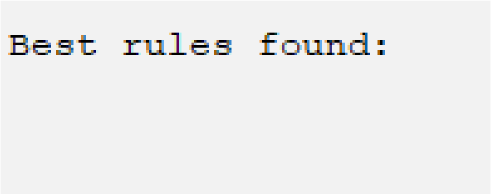
```
Best rules found:

 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:(84.58)
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198    <conf:(1)> lift:(1.63) lev:(0.18) [76] conv:(76.47)
 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210    <conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74)
 4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201    <conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01)
 5. physician-fee-freeze=n 247 ==> Class=democrat 245    <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8)
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18)
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46)
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198    <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:(14.62)
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85)
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210    <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47)
```

Here, number of members of Democratic party are more in number as compared to members of Republic party which ultimately increases the probability of their appearance in the most frequent item sets. Hence, we see no member of republic party in the rules. Probably if we increase the number of members of Republic party, we may find few entries in rules.

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

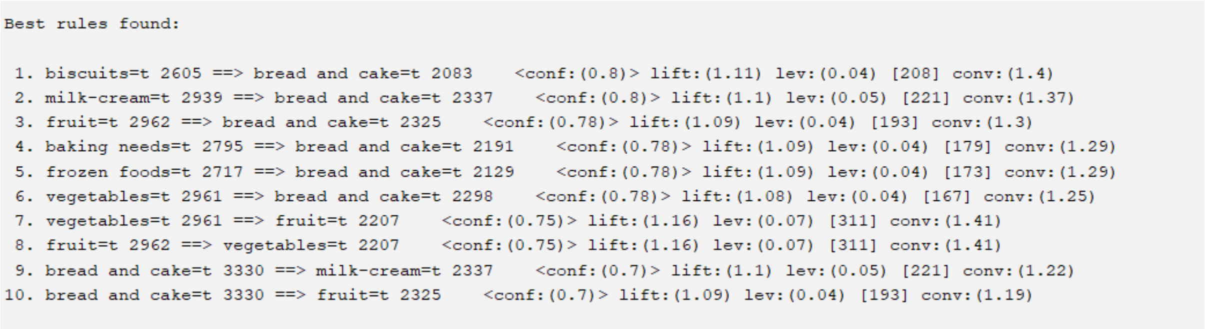
1. minConfidence 0.9:



```
Best rules found:
```

At minConf = 0.9 with minsupport 0.3, no rule is generated.

2. minConfidence 0.6:



```
Best rules found:

 1. biscuits=t 2605 ==> bread and cake=t 2083    <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4)
 2. milk-cream=t 2939 ==> bread and cake=t 2337    <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37)
 3. fruit=t 2962 ==> bread and cake=t 2325    <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3)
 4. baking needs=t 2795 ==> bread and cake=t 2191    <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29)
 5. frozen foods=t 2717 ==> bread and cake=t 2129    <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29)
 6. vegetables=t 2961 ==> bread and cake=t 2298    <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25)
 7. vegetables=t 2961 ==> fruit=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41)
 8. fruit=t 2962 ==> vegetables=t 2207    <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41)
 9. bread and cake=t 3330 ==> milk-cream=t 2337    <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22)
10. bread and cake=t 3330 ==> fruit=t 2325    <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
```

At minConf = 0.6 with minSupport 0.3, we can see generated rules

Conclusion:

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Apriori algorithm allows us to mine the frequent itemset in order to generate association rule between them. The main limitation is time required to hold a vast number of candidate sets with much frequent