S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 3

Aim: Demonstration of Association Rule Mining on dataset using Apriori Algorithm. and F-P Growth by using Weka Tool on the training data set supermarket and compare it by changing the value of support and confidence.

Name of Student: Shrutika Pradeep Bagdi

Roll No.: CS22130

Semester/Year: 6th / 3rd

Academic Session: 2024-2025

Date of Performance:

Date of Submission:

AIM: Demonstration of Association Rule mining on dataset using Apriori Algorithm and F-P Growth by using Weka Tool on the training data set supermart and compare it by changing the value of support and confidence.

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- To create the association rule between different objects
- The association rule describes how two or more objects are related to one another

HARDWARE AND SOFTWARE REQUIRMENTS:

Hardware Requirement:

Software Requirement:

THEORY:

ASSOCIATION RULE MINING

Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrences, in a database. It identifies frequent if-then associations, which themselves are the association rules.

The goal of association rule generation is to find interesting patterns and trends in transaction databases. Association rules are statistical relations between two or more items in the dataset. In a supermarket basket application, associations express the relations between items that are bought together.

Association rules are created by searching data for frequent if-then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true. A third metric, called lift, can be used to compare confidence with expected confidence, or how many times an if-then statement is expected to be found true.

Association rules are calculated from itemsets, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well-represented in data.

ASSOCIATION RULE ALGORITHMS:

With the Apriori algorithm, candidate itemsets are generated using only the large itemsets of the previous pass. The large itemset of the previous pass is joined with itself to generate all itemsets with a size that's larger by one. Each generated itemset with a subset that is not large is then deleted. The remaining itemsets are the candidates. The Apriori algorithm considers any subset of a frequent itemset to also be a frequent itemset. With this approach, the algorithm reduces the number of candidates being considered by only exploring the itemsets whose support count is greater than the minimum support count, according to Sayad.

The Apriori Algorithm:

The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

FP Growth Algorithm:

The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. For so much, it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

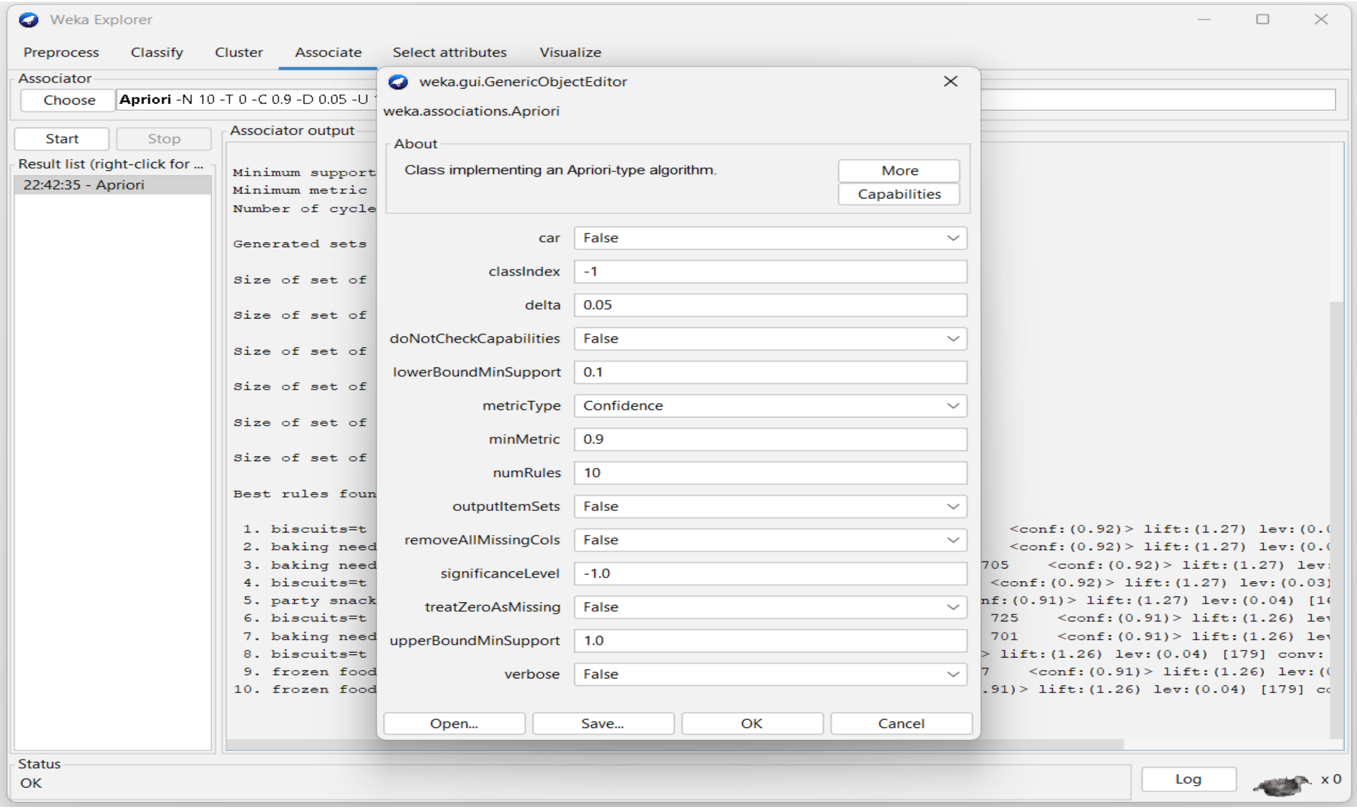
Description:

We need to use supermart data set which is already present in Weka training sets.

Procedure:

Procedure for Association Rules:

OUTPUT (SCREENSHOTS)



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U

22:42:35 - Apriori

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

car: False

classIndex: -1

delta: 0.05

doNotCheckCapabilities: False

lowerBoundMinSupport: 0.1

metricType: Confidence

minMetric: 0.9

numRules: 10

outputItemSets: False

removeAllMissingCols: False

significanceLevel: -1.0

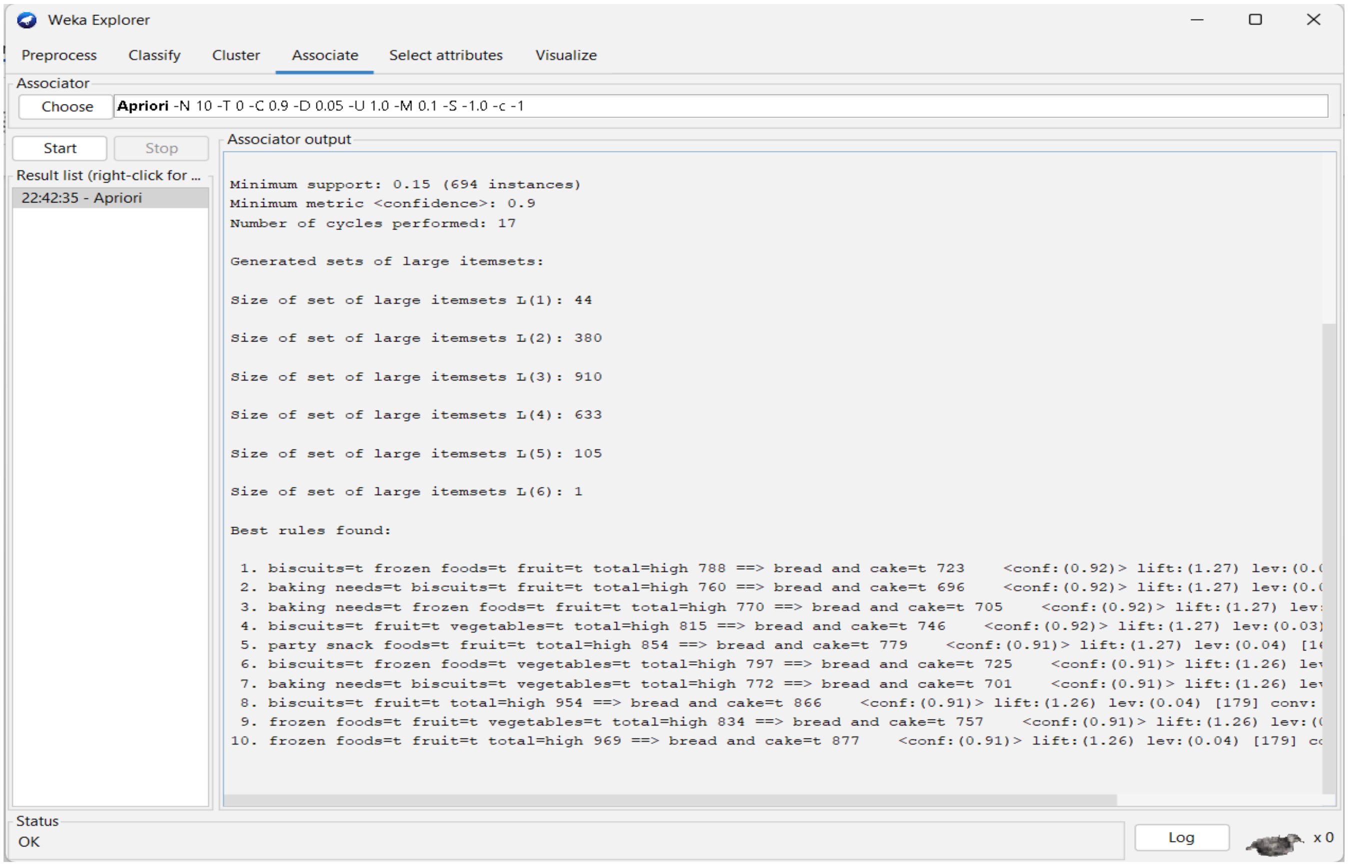
treatZeroAsMissing: False

upperBoundMinSupport: 1.0

verbose: False

```
Minimum support
Minimum metric
Number of cycle

Generated sets

Size of set of

Size of set of

Size of set of

Size of set of

Size of set of

Size of set of

Best rules foun

 1. biscuits=t          <conf:(0.92)> lift:(1.27) lev:(0.(
 2. baking need         <conf:(0.92)> lift:(1.27) lev:(0.(
 3. baking need      705    <conf:(0.92)> lift:(1.27) lev:
 4. biscuits=t          <conf:(0.92)> lift:(1.27) lev:(0.03]
 5. party snack      nf:(0.91)> lift:(1.27) lev:(0.04) [1(
 6. biscuits=t        725    <conf:(0.91)> lift:(1.26) lev
 7. baking need       701    <conf:(0.91)> lift:(1.26) lev
 8. biscuits=t       > lift:(1.26) lev:(0.04) [179] conv:
 9. frozen food      7    <conf:(0.91)> lift:(1.26) lev:((
10. frozen food      .91)> lift:(1.26) lev:(0.04) [179] co
```

OK

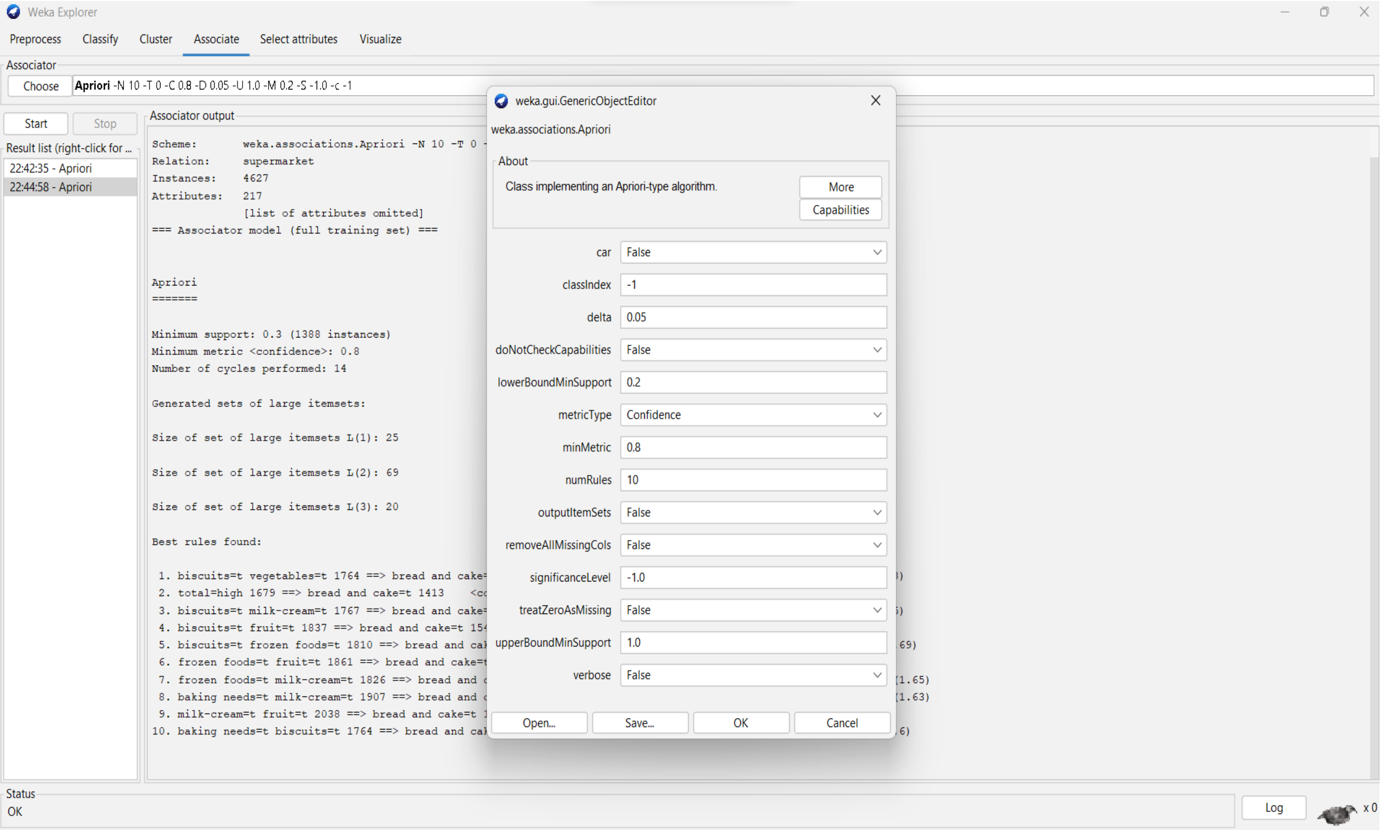


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

22:42:35 - Apriori

```
Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

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.(
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.(
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [1(
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:((
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] cc
```

Status
OK



Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c -1

22:42:35 - Apriori
22:44:58 - Apriori

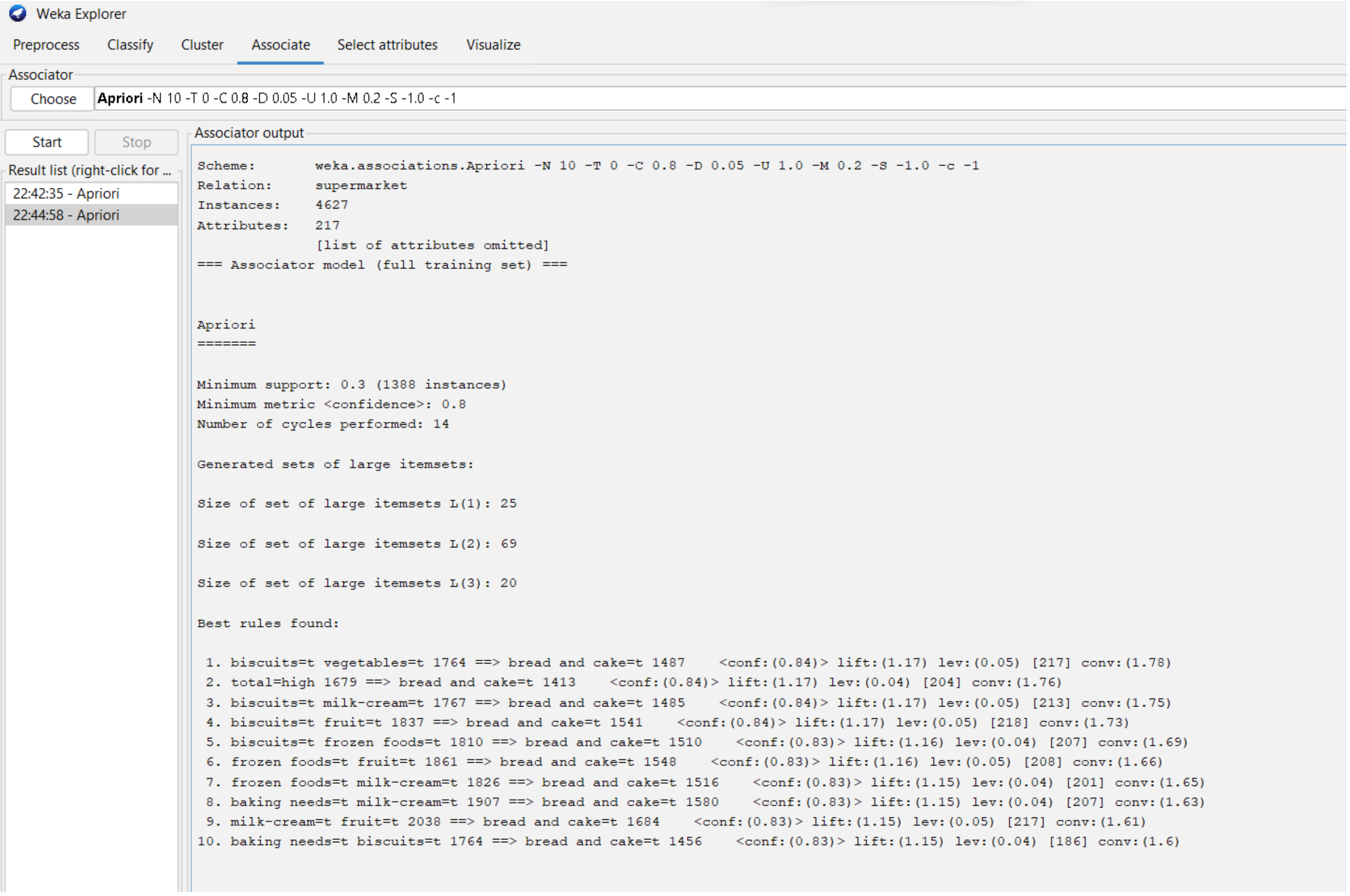
```
Scheme:       weka.associations.Apriori -N 10 -T 0 -
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===


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

Size of set of large itemsets L(2): 69

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

Best rules found:

 1. biscuits=t vegetables=t 1764 ==> bread and cake=                                  3)
 2. total=high 1679 ==> bread and cake=t 1413    <cc
 3. biscuits=t milk-cream=t 1767 ==> bread and cake=                                  5)
 4. biscuits=t fruit=t 1837 ==> bread and cake=t 154
 5. biscuits=t frozen foods=t 1810 ==> bread and cal                                  .69)
 6. frozen foods=t fruit=t 1861 ==> bread and cake=t
 7. frozen foods=t milk-cream=t 1826 ==> bread and c                                  (1.65)
 8. baking needs=t milk-cream=t 1907 ==> bread and c                                  (1.63)
 9. milk-cream=t fruit=t 2038 ==> bread and cake=t 1
10. baking needs=t biscuits=t 1764 ==> bread and cal                                  .6)
```

weka.associations.Apriori

Class implementing an Apriori-type algorithm.

car: False
classIndex: -1
delta: 0.05
doNotCheckCapabilities: False
lowerBoundMinSupport: 0.2
metricType: Confidence
minMetric: 0.8
numRules: 10
outputItemSets: False
removeAllMissingCols: False
significanceLevel: -1.0
treatZeroAsMissing: False
upperBoundMinSupport: 1.0
verbose: False

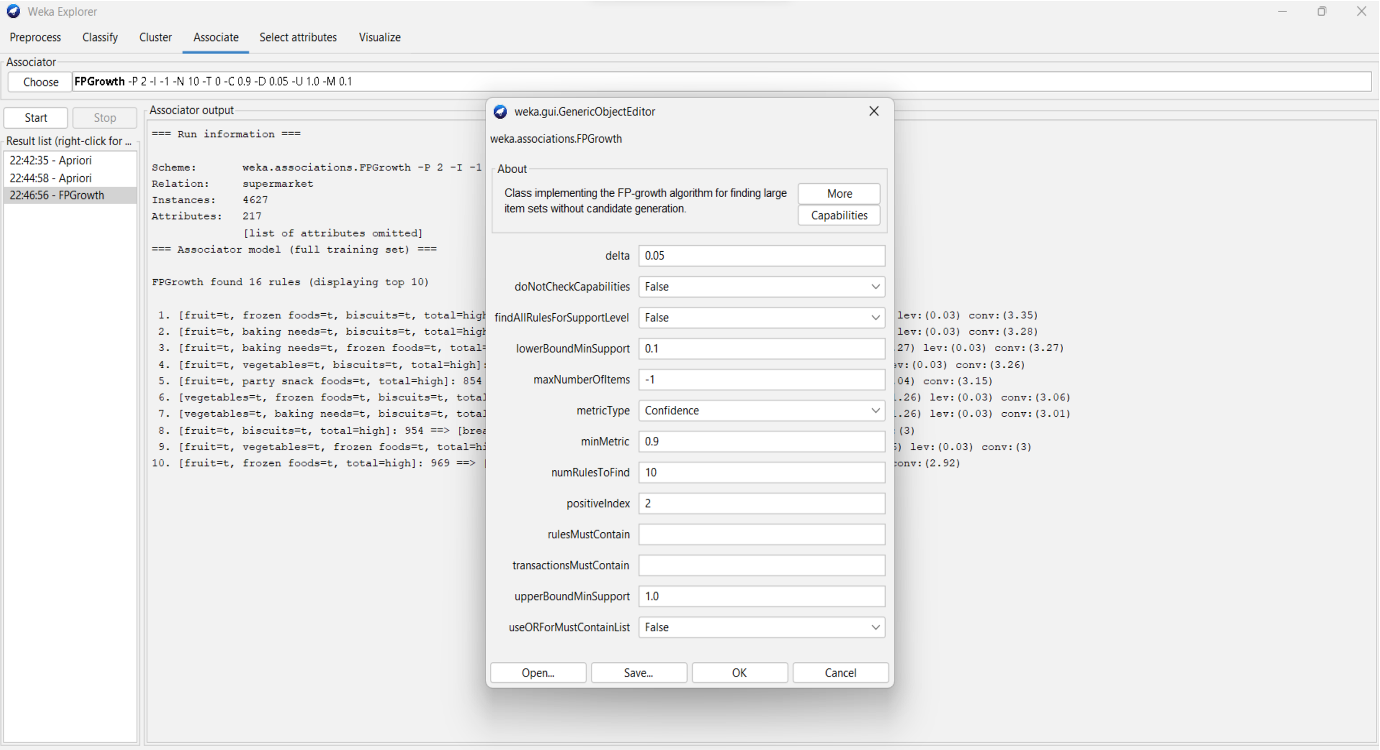
Status
OK



Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c -1

22:42:35 - Apriori
22:44:58 - Apriori

```
Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2 -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

Size of set of large itemsets L(2): 69

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

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) [218] 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) [186] conv:(1.6)
```



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

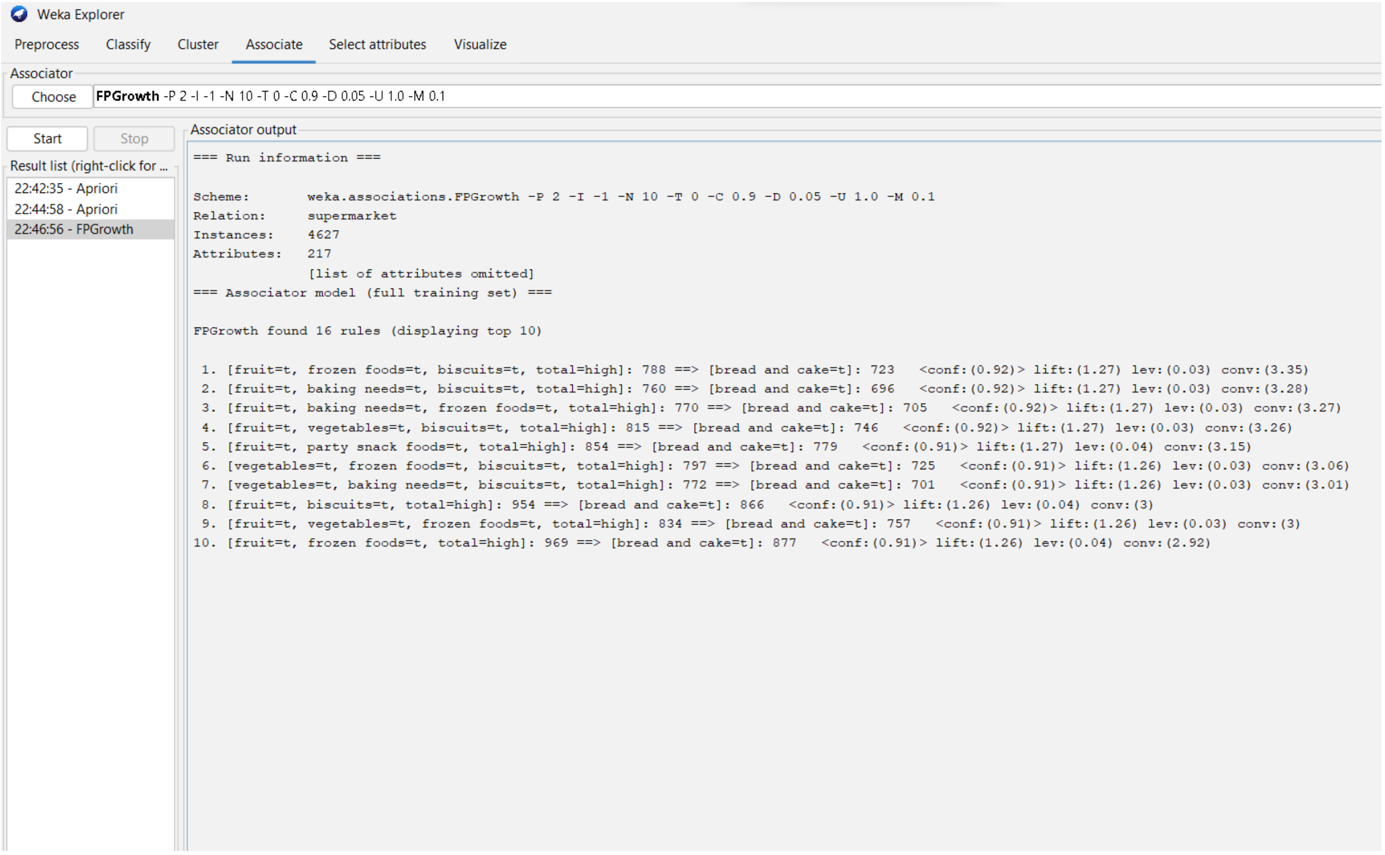
22:42:35 - Apriori
22:44:58 - Apriori
22:46:56 - FPGrowth

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 16 rules (displaying top 10)

 1. [fruit=t, frozen foods=t, biscuits=t, total=high          lev:(0.03) conv:(3.35)
 2. [fruit=t, baking needs=t, biscuits=t, total=high          lev:(0.03) conv:(3.28)
 3. [fruit=t, baking needs=t, frozen foods=t, total=          .27) lev:(0.03) conv:(3.27)
 4. [fruit=t, vegetables=t, biscuits=t, total=high]:          v:(0.03) conv:(3.26)
 5. [fruit=t, party snack foods=t, total=high]: 854           04) conv:(3.15)
 6. [vegetables=t, frozen foods=t, biscuits=t, total          .26) lev:(0.03) conv:(3.06)
 7. [vegetables=t, baking needs=t, biscuits=t, total          .26) lev:(0.03) conv:(3.01)
 8. [fruit=t, biscuits=t, total=high]: 954 ==> [brea          (3)
 9. [fruit=t, vegetables=t, frozen foods=t, total=hi          ) lev:(0.03) conv:(3)
10. [fruit=t, frozen foods=t, total=high]: 969 ==>            onv:(2.92)
```

weka.associations.FPGrowth

Class implementing the FP-growth algorithm for finding large item sets without candidate generation.

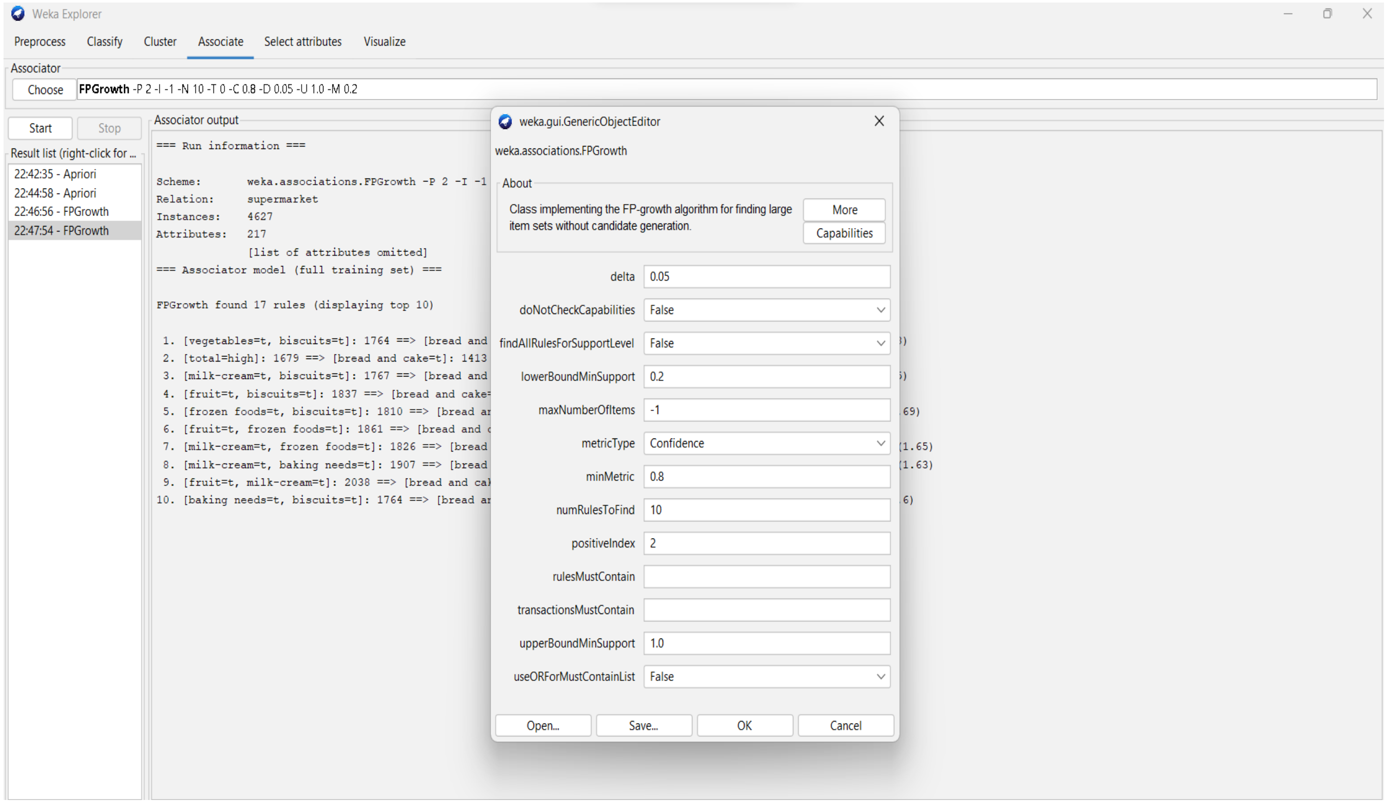
delta: 0.05
doNotCheckCapabilities: False
findAllRulesForSupportLevel: False
lowerBoundMinSupport: 0.1
maxNumberOfItems: -1
metricType: Confidence
minMetric: 0.9
numRulesToFind: 10
positiveIndex: 2
rulesMustContain:
transactionsMustContain:
upperBoundMinSupport: 1.0
useORForMustContainList: False



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

22:42:35 - Apriori
22:44:58 - Apriori
22:46:56 - FPGrowth

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 16 rules (displaying top 10)

 1. [fruit=t, frozen foods=t, biscuits=t, total=high]: 788 ==> [bread and cake=t]: 723   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.35)
 2. [fruit=t, baking needs=t, biscuits=t, total=high]: 760 ==> [bread and cake=t]: 696   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.28)
 3. [fruit=t, baking needs=t, frozen foods=t, total=high]: 770 ==> [bread and cake=t]: 705   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.27)
 4. [fruit=t, vegetables=t, biscuits=t, total=high]: 815 ==> [bread and cake=t]: 746   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.26)
 5. [fruit=t, party snack foods=t, total=high]: 854 ==> [bread and cake=t]: 779   <conf:(0.91)> lift:(1.27) lev:(0.04) conv:(3.15)
 6. [vegetables=t, frozen foods=t, biscuits=t, total=high]: 797 ==> [bread and cake=t]: 725   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.06)
 7. [vegetables=t, baking needs=t, biscuits=t, total=high]: 772 ==> [bread and cake=t]: 701   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.01)
 8. [fruit=t, biscuits=t, total=high]: 954 ==> [bread and cake=t]: 866   <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(3)
 9. [fruit=t, vegetables=t, frozen foods=t, total=high]: 834 ==> [bread and cake=t]: 757   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3)
10. [fruit=t, frozen foods=t, total=high]: 969 ==> [bread and cake=t]: 877   <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(2.92)
```



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2

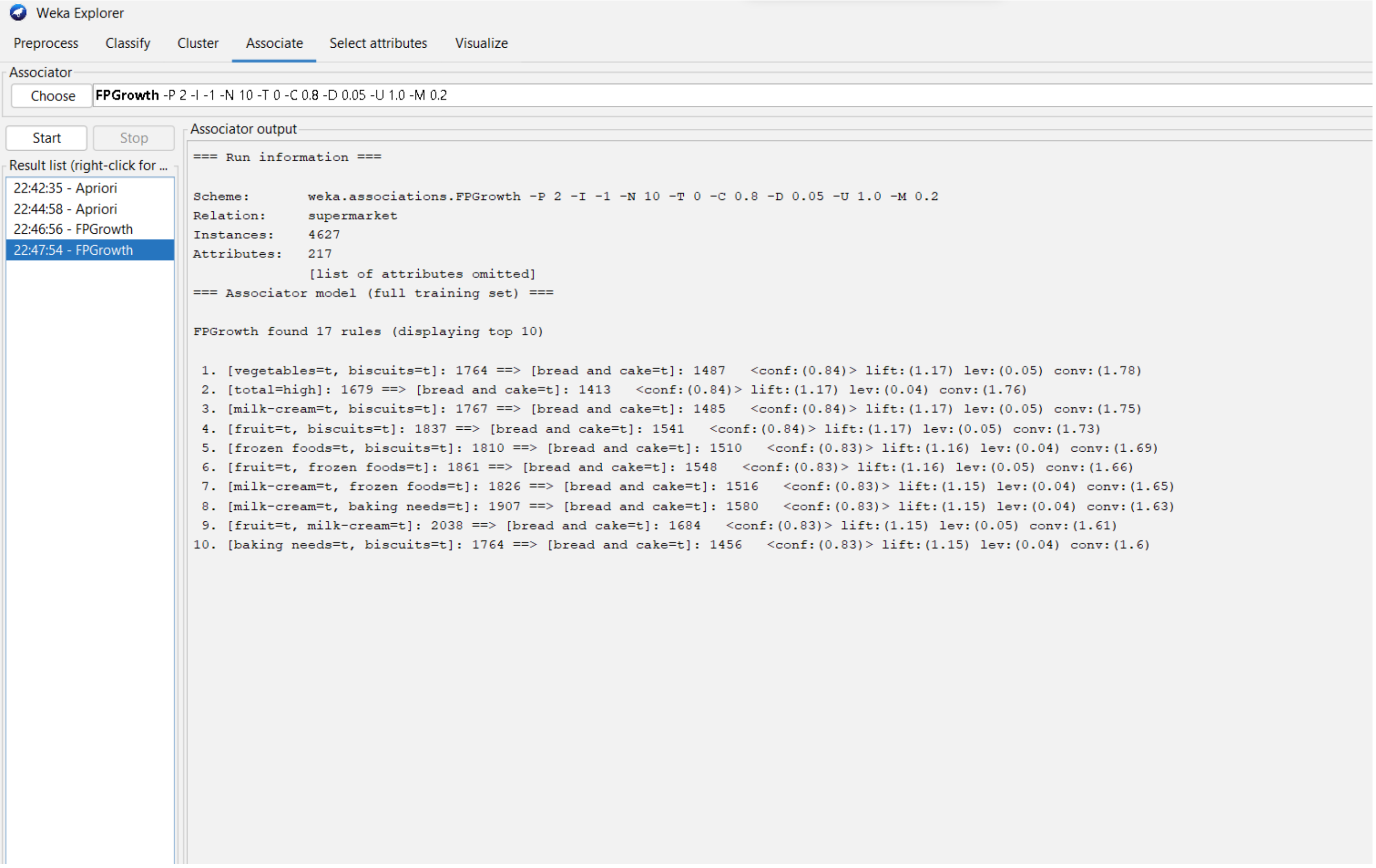
22:42:35 - Apriori
22:44:58 - Apriori
22:46:56 - FPGrowth
22:47:54 - FPGrowth

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 17 rules (displaying top 10)

 1. [vegetables=t, biscuits=t]: 1764 ==> [bread and
 2. [total=high]: 1679 ==> [bread and cake=t]: 1413
 3. [milk-cream=t, biscuits=t]: 1767 ==> [bread and
 4. [fruit=t, biscuits=t]: 1837 ==> [bread and cake=
 5. [frozen foods=t, biscuits=t]: 1810 ==> [bread ar
 6. [fruit=t, frozen foods=t]: 1861 ==> [bread and c
 7. [milk-cream=t, frozen foods=t]: 1826 ==> [bread
 8. [milk-cream=t, baking needs=t]: 1907 ==> [bread
 9. [fruit=t, milk-cream=t]: 2038 ==> [bread and cak
10. [baking needs=t, biscuits=t]: 1764 ==> [bread ar
```

weka.associations.FPGrowth

Class implementing the FP-growth algorithm for finding large item sets without candidate generation.

delta	0.05
doNotCheckCapabilities	False
findAllRulesForSupportLevel	False
lowerBoundMinSupport	0.2
maxNumberOfItems	-1
metricType	Confidence
minMetric	0.8
numRulesToFind	10
positiveIndex	2
rulesMustContain	
transactionsMustContain	
upperBoundMinSupport	1.0
useORForMustContainList	False



Associator: **FPGrowth** -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 17 rules (displaying top 10)

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

CONCLUSION:

DISCUSSION AND VIVA VOCE:

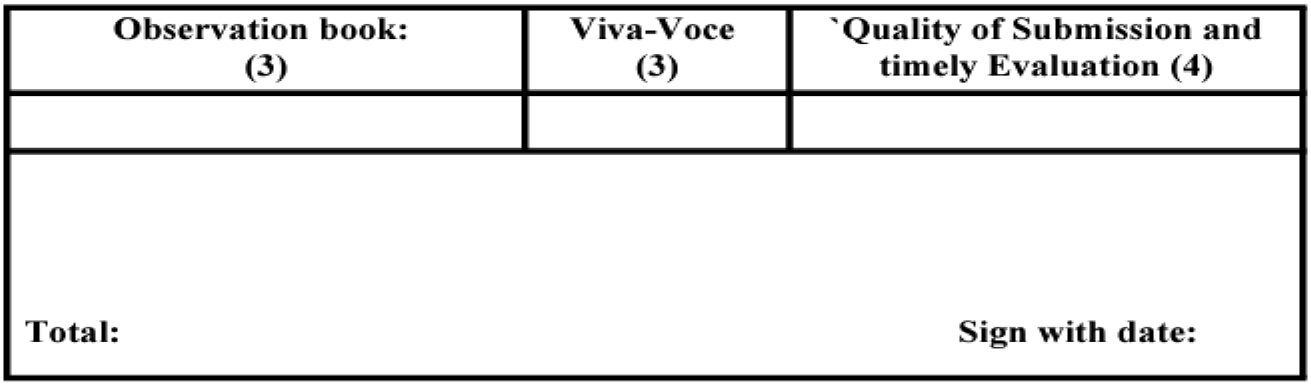
- What is association rule mining also called as?
- What are the applications of association rule?
- What are the disadvantages of association rule mining?
- Explain the term Support and Confidence
- How Apriori algorithm is differ with F-P Growth

REFERENCE:

- https://www.google.com/search?q=objective/in+apriori+algo&spell=1&sa=X&ved=2ahUKEwiOyqKGm5_-AhU1RmwGHfu6CUsQBSgAegQIBhAB&biw=1366&bih=600&dpr=1
- https://www.google.com/search?q=association+rule+mining+theory&oq=association+rule+mini

ng+theory&aqs=chrome..69i57j33i160l2.13169j0j7&sourceid=chrome&ie=UTF-8 https

Data Mining – Concepts and Techniques, Jiawei Han & Micheline Kamber, Morgan Kauf-mann Publishers, Elsevier, 2nd Edition, 2006.



Observation book: **(3)**	**Viva-Voce** **(3)**	**`Quality of Submission and timely Evaluation (4)**
Total:		**Sign with date:**