

• Viva Questions :-

Q1) State the Apriori Law or property. How does apriori algo be significantly used for data mining application?

- Apriori property :- "All non-empty subsets of a frequent itemset must also be frequent".
- Significance :- used to discover frequent itemsets & generate association rules in large datasets, aiding in market basket analysis, recommendation system etc.

Q2) Enlist the limitation of the Apriori Algo ~~must~~ worst implementation on huge datasets for data mining.

- High Computational Cost for large datasets.
- Generates numerous Candidate itemsets, the memory usage
- Inefficient with low support threshold due to excessive Iterative
- Slow with dense or high-dimensional datasets.

Experiment 3-03

• Objective:- Demonstrate of association rule using any dataset using Apriori algo in weka explorer.

• Theory:- Association rules mining method for determining frequent itemsets and Association rule.

• Steps:- ① Create itemsets starting from 1 element in each, till all the possible itemsets are considered.

② Remove those itemsets having support count and confidence less than the threshold.

③ In the last step, itemsets with more possible elements will be the frequent itemset.

④ From the frequent itemsets, determine the association rule by computing confidence for each possible Association rule & consider only those with confidence value more than the threshold.

• Result:- In this experiment have seen the total rules generated theoretically and by the weka are matching, hence the rules generated are 5.

EXPERIMENT 3

Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Associate

Choose

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

Start

Stop

Result list (right-click for ...)

2059x45 - Apriori

Associate output

----- ASSOCIATOR: MODE= FULL TRAINING SET -----

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. biscuit<= frozen food<= fruit<= total-high 768 ==> bread and cake<= 723 <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)

2. baking need<= biscuit<= fruit<= total-high 760 ==> bread and cake<= 696 <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)

3. baking need<= frozen food<= fruit<= total-high 770 ==> bread and cake<= 705 <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)

4. biscuit<= fruit<= vegetable<= total-high 815 ==> bread and cake<= 746 <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)

5. party snack food<= fruit<= total-high 854 ==> bread and cake<= 779 <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)

6. biscuit<= frozen food<= vegetable<= total-high 797 ==> bread and cake<= 725 <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)

7. baking need<= biscuit<= vegetable<= total-high 772 ==> bread and cake<= 701 <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)

8. biscuit<= fruit<= total-high 994 ==> bread and cake<= 866 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)

9. frozen food<= fruit<= vegetable<= total-high 834 ==> bread and cake<= 757 <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)

10. frozen food<= fruit<= total-high 969 ==> bread and cake<= 877 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)

Status

OK

Log

x0