Title :- Apriori Algorithm.

Problem Statement :- Apply apriori algorithm to find frequently occuring items from given data and generate strong association using support & confidence thresholds.

Objectives :-
- Understanding the concepts of association rules.
- Creating association rules to derive reccommendations depending on the confidence of the rules.

Outcomes :
Students will be able to
- understand the concept of association rules.
- Create association rules to derive recommendations depending on the confidences of the rules.

Software and Hardware Requirements :
- Fedora 20 / windows 10
- Jupyter Notebook / Google colab.

Theory :
Apriori Algorithm :
It is used for finding frequent item sets in a dataset from boolean association rules. It uses prior knowledge of frequent itemset properties we apply an iterative approch or

level wise search where k frequent itemsets are used to find fill Itemsets.

Apriori Property :-

All non-empty dataset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure Apriori con assumes that.

All Subset of a frequent itemset must be frequent If an item is in frequent, all its super sets will be freq infrequent.

Important Definitions:

1) Support :- It is one of the measure of interestingness. This tells about the usefullness & certainity of rules 5% Support means 5% of transactions in database follow the rule.

Support $(A \rightarrow B)$ = Support-count $(A \cup B)$

2) Confidence : A confidence of 60% means that 60% of customers who purchased milk and butter also brought bread.

Confidence $(A \rightarrow B)$ = Support-count$(A \cup B)$ / support-count(A)

2) If a rule satisfies both minimum support & ~~maximum~~ minimum confidence it is strange rule.

3) Support_count (x) : No of transactions in which x appears If x is $(A \cup B)$ then it is the no of transactions in which A & B.

4) **Maximum Itemset :** An itemset is maximal frequent if none of its supersets are frequent.

5) **closed itemset :** An itemset is closed if none of its immediate supersets have same support. count same as Itemset.

6) **K-itemset :** Itemset which contains k items is a k itemset so it can be ~~solid~~ said that an itemset is frequent if the corresponding support counts greater than minimum support count.

Limitations :-

1) Computationally Expensive : Even through the apriori algorithm reduces the number of candidate itemset to consider this number could still be huge when store inventories are large or when the support threshold is low. How ever using hash tables we can sort candidate itemset more effeciently.

2) Spurious associations : Analysis of large inventories would involve more itemset configuration and support threshold might have to be lowered to detect certain associations However lowing the threshold might also increase the number of spurious association detected.

Conclusion :- Thus, we have successfully applied and implemented apriori algorithm to find frequently occuring items from given data & generated strong association rules using support, confidence thresholds.