LP2 - Assignment - A3

Title:- Apriori algorithm.

Problem statement:- Apply apriori algorithm to find frequent occuring items from a given data & find strong association rules using support & confidence threshold.

Objective:- Implementing apriori algorithm.

Outcome:- We will learn the mine frequent itemsets to generate stron association rules.

S/W & H/W:-

1) 64 bit OS
2) Python 3

Theory:-

- Apriori employs a iterative approach known as level wise search where k-itemsets are used to explore (k+1) itemsets.
- set of frequent 1-itemsets found by scanning the DB to accumulate the count for each item & collecting these items. The resulting set is L_1.
- Next L_1 is used to find L_2, the set of 2-itemsets which is used to find L_2 & so on until no more k-itemsets can be found.

- The finding of each L_k requires one full scan of the DB.

Generating associating rule:-

1) Once the frequent itemsets from a transaction in a database D have been found, it is straight forward to generate strong association rules.

$$\text{confidence } (A \Rightarrow B) = P(B|A) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

- For each subsets l, generate all non-empty subsets of l.
- For every non-empty subsets s of l, output the rule

"$s \Rightarrow (l-s)$ if

$$\frac{\text{support\_count}(l)}{\text{support\_count}(s)}$$

Algorithm:-

- L_1 = find_frequent_1-itemset (D)
- for ($k=2$, $L_{k-1} \neq \phi$, $k++$) {

C_k = apriori_gen (L_{k-1});
for each transaction $t \in D$ {

C_t = subset (C_k, t)
for each candidate $c \in C_t$
c.count ++
}
$L_k = \{ c \in C_k \mid c.count \geq min\_sup \}$
}
return $L = \cup_k L_k$

procedure apriori-gen (L_{k-1} : frequent (k-1)-itemsets)
1) for each itemset $L_1 \in L_{k-1}$
2) for each itemset $L_2 \in L_{k-1}$
3) if ($L_1[1] = L_2[1] \cap L_1[2] = L_2[2] \cap$
4) ($L_1[k-2] = L_2[k-2]$) $\cap$
5) $L_1[k-1] < L_2[k-1]$ then {
6) $c = L_1 \times L_2$;
7) if has-infrequent-subset (c: candidate,
k-itemset);
8) then delete c;
9) else add c to C_k
10)
11) }
12) return C_k;

procedure has_infrequent_subset
1) for each (K-1)-subset S of c
2) if S ∈ L_{K-1} then
3) return TRUE
4) return FALSE

Test cases:-

Description	Expected	Actual
1) Preprocess data find total txns.	Success	Success
2) Generate frequent itemset.	Freq. itemset generated	Freq. itemset generated.
3) Strong association rule creation	Association rules created	Association rules created
4) Apriori property holds up.	Apriori was satisfied	Apriori was satisfied.

Conclusion:- Thus, we have implemented apriori algo & performed market basket analysis.