Title: Association Rules

Objective/Aim: Perform the following tasks:

1. Implement apriori algorithm for generating association rules.
2. Experiment with different values of support, confidence and maximum rule length.
3. Tabulate the result with frequent item sets, total number of rules generated for different support and confidence.
4. Find interesting rules from above combined rules using following measures/metrics.
    a. Lift
    b. Chi-square test x^2
    c. All_confidence measure
    d. Max_confidence measure
    e. Kulczynski measure
    f. Cosine measure

Introduction:

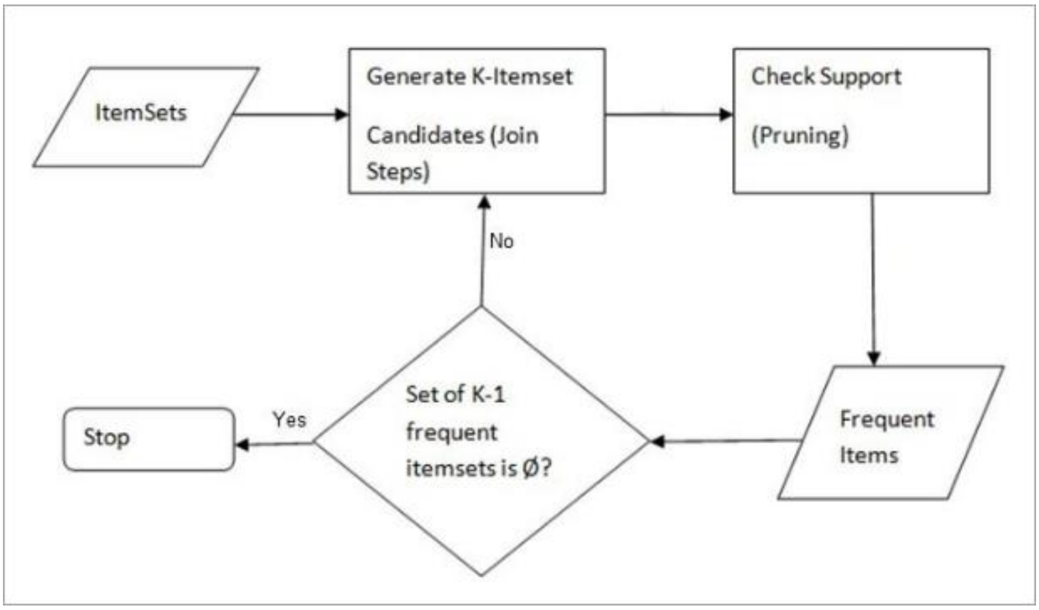
Association rules: Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is Market Based Analysis

Theory/Algorithm:

1. 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.
    a. Apriori Property – All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes that - All subsets of a frequent itemset must be frequent(Apriori propertry). If an itemset is infrequent, all its supersets will be infrequent.
    b. Support: Support refers to the default popularity of any product. You find the support as a quotient of the division of the number of transactions comprising that product by the total number of transactions.
    c. Confidence: This says how likely item Y is purchased when item X is purchased, expressed as {X -> Y}. This is measured by the proportion of transactions with item X, in which item Y also appears.
    d. Lift: Lift basically tells us that the likelihood of buying a Burger and Ketchup together is 3.33 times more than the likelihood of just

buying the ketchup. A Lift of 1 means there is no association between products A and B. Lift of greater than 1 means products A and B are more likely to be bought together.

Documentation/Block Diagrams:



Procedure:

1. Apriori algorithm:
   a. Step 1: Make a frequency table of all the products that appear in all the transactions. Now, short the frequency table to add only those products with a threshold support level of over 50 percent.
   b. Step 2: Create pairs of products i.e. k=2
   c. Step 3: Implementing the same threshold support of 50 percent and consider the products that are more than 50 percent.
   d. Repeat step 2,3 for k=3,4,5...

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence } (A \longrightarrow B) = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

Pseudocode: Pseudo-code :

C_k: Candidate itemset of size k

```
$L_k$: frequent itemset of size k

L1 = {frequent items};

for (k = 1; Lk !=  ; k++) do begin

    Ck+1 = candidates generated from Lk;

    for each transaction t in database do

        increment the count of all candidates in Ck+1 that are contained in t
    $L_{k+1}$ = candidates in Ck+1 with min_support

    end

return $U_k$ $L_k$;
```

Conclusion:

1. Advantages
    a. Easy to understand algorithm
    b. Join and Prune steps are easy to implement on large itemsets in large databases
2. Disadvantages
    a. It requires high computation if the itemsets are very large and the minimum support is kept very low.
    b. The entire database needs to be scanned.

References:

1. https://www.softwaretestinghelp.com/apriori-algorithm/
2. https://www.slideshare.net/INSOFE/apriori-algorithm-36054672