Frequent Itemset Generation & Association Rule Mining:

* This is an implementation of Apriori algorithm for frequent itemset generation.

Also, FP growth algorithm is used to generate frequent itemset thereby reducing execution time.

* It finds association rules based on the frequent itemset generated.

Every rule is composed by two different sets of items, also known as *itemset*, X and Y , where X is called *antecedent* or left-hand-side (LHS) and Y *consequent* or right-hand-side (RHS).

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I = {milk , bread , butter , beer , diapers }

An example rule for the supermarket could be {butter, bread} ⇒ {milk}

meaning that if butter and bread are bought, customers also buy milk.

* It finds the maximal and closed frequent itemset for a given value of minimum support and confidence.

Support is an indication of how frequently the itemset appears in the dataset. The support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

Confidence is an indication of how often the rule has been found to be true. The confidence value of a rule, X => Y with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y.

Input:

* Dataset should preferably be a .csv file without any missing values.
* Path of the .csv file should be the input along with the minimum support and confidence for which frequent itemset are to be found.
* If the transaction has more than one appearance of an item then only one appearance is considered.

Output:

* Frequent itemset along with their support values are stored in a .txt file.
* Association rules along with the support of antecedent, consequent and confidence of the rule is stored in another.txt file
* Maximal subsets and closed subsets are generated and stored in a .txt file.

**Note**: DataMining folder finds frequent itemset by FP Growth Algorithm whereas DatMining1 folder uses Apriori Method for frequent itemset generation.