Name:	Saurabhsing Dipaksing Pardeshi
Roll No:	35
Class/Sem:	TE/V
Experiment No.:	5
Title:	Using open-source tools Implement Association Mining Algorithms.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To implement Apriori Algorithm on a large dataset using Open-source tool WEKA.

Objective: To make students well versed with open-source tools like WEKA to implement Apriori algorithm.

Theory:

- Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an itemset occurs in a transaction.
- A typical example is a Market Based Analysis. Market Based Analysis is one of the key techniques used by large relations to show associations between items.
- It allows retailers to identify relationships between the items that people buy together frequently.
- Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Support Count () – Frequency of occurrence of a itemset.
Here ({Milk, Bread, Diaper})= 2

Frequent Itemset – An itemset whose support is greater than or equal to minsup threshold.

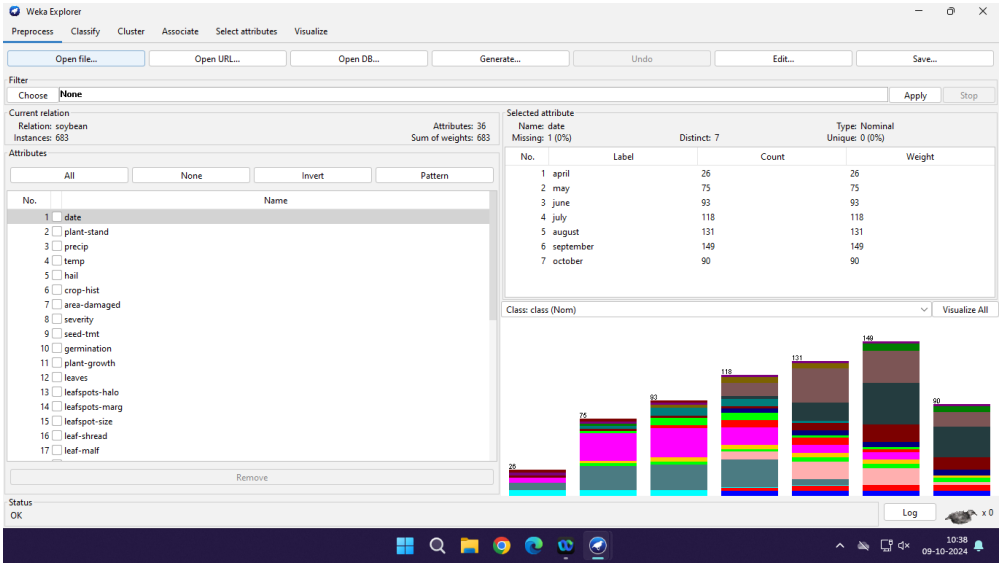
Association Rule – An implication expression of the form X Y, where X and Y are any 2 itemsets.

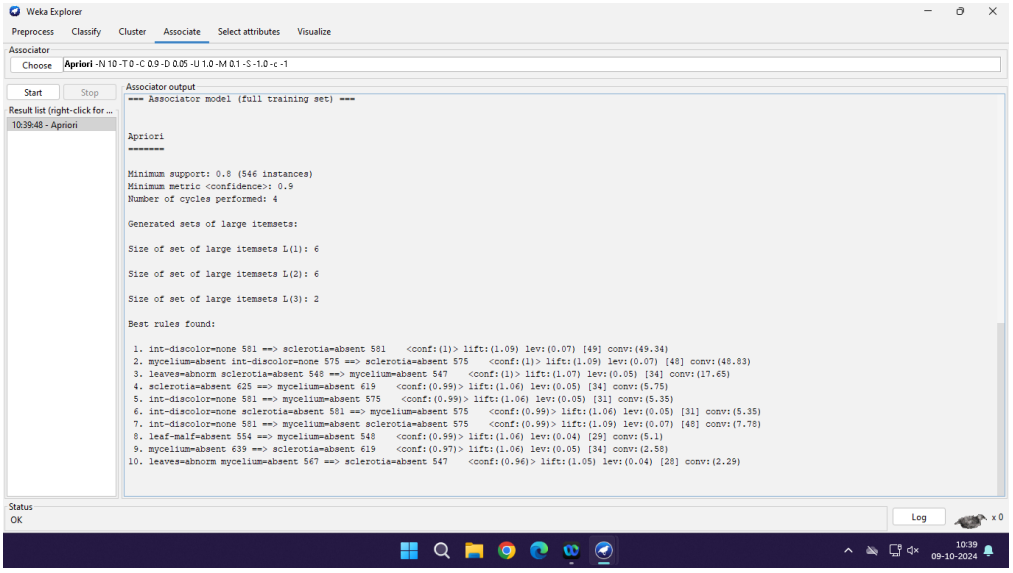
Example: {Milk, Diaper}{Beer}

- WEKA contains an implementation of the Apriori algorithm. The algorithm works only with discrete data.

- It can identify statistical dependencies between groups of attributes.
- Apriori algorithm can compute all rules that have a given minimum support and exceed a given confidence.
- Clicking on the "Associate" tab will bring up the interface for the association rule algorithms.
- The Apriori algorithm which we will use is the default algorithm selected. However, in order to change the parameters for this run (e.g., support, confidence, etc.) we click on the text box immediately to the right of the "Choose" button. Note that this box, at any given time, shows the specific command line arguments that are to be used for the algorithm.
- WEKA allows the resulting rules to be sorted according to different metrics such as confidence, leverage, and lift. We can also change the default value of rules (10) to be 20; this indicates that the program will report no more than the top 20 rules. The upper bound for minimum support is set to 1.0 (100%) and the lower bound to 0.1 (10%).
- Apriori in WEKA starts with the upper bound support and incrementally decreases support (by delta increments which by default is set to 0.05 or 5%). The algorithm halts when either the specified number of rules are generated, or the lower bound for min. support is reached. Once the parameters have been set, the command line text box will show the new command line. We now click on start to run the program. This results in a set of rules. The panel on the left ("Result list") now shows an item indicating the algorithm that was run and the time of the run. You can perform multiple runs in the same session each time with different parameters. Each run will appear as an item in the Result list panel. Clicking on one of the results in this list will bring up the details of the run, including the discovered rules in the right panel. In addition, right-clicking on the result set allows us to save the result buffer into a separate file. Note that the rules were discovered based on the specified threshold values for support and lift. For each rule, the frequency counts for the LHS and RHS of each rule is given, as well as the values for confidence, lift, leverage, and conviction. In most cases, it is sufficient to focus on a combination of support, confidence, and either lift or leverage to quantitatively measure the "quality" of the rule. However, the real value of a rule, in terms of usefulness and action ability, is subjective and depends heavily on the particular domain and business objectives.

OUTPUT:





Conclusion:

Explain the main steps involved in the Apriori algorithm.

The Apriori algorithm is a systematic approach for discovering frequent itemsets and association rules in large datasets. The process can be broken down into the following steps:

1. Set a minimum support threshold to filter out infrequent itemsets. This threshold determines the minimum number of times an itemset must appear to be considered frequent.

2. Scan the dataset to identify all 1-itemsets (individual items) that meet the minimum support threshold.
3. Use the frequent 1-itemsets to generate candidate 2-itemsets by combining frequent items from the previous step.
4. Scan the dataset again to calculate the support (frequency) of the candidate 2-itemsets. Prune those that do not meet the minimum support threshold.
5. Iterate the process by generating larger candidate itemsets (e.g., 3-itemsets, 4-itemsets, etc.) using frequent itemsets from the previous iteration. After each iteration, scan the dataset to calculate support and prune infrequent itemsets.
6. Once all frequent itemsets are identified, the algorithm generates association rules by splitting the frequent itemsets into left-hand side (LHS) and right-hand side (RHS).
7. Calculate the confidence for each rule, which measures how often the RHS appears in transactions that contain the LHS.
8. Prune rules that do not meet the minimum confidence threshold, retaining only those rules that pass the threshold.
9. Sort the association rules based on metrics such as confidence, lift, or leverage to highlight the most interesting patterns.
10. The process continues until no more frequent itemsets or association rules can be generated, providing valuable insights into the relationships between items in the dataset.

What are the key parameters in the Apriori algorithm and how do they affect its performance?

The key parameters in the Apriori algorithm and their effects on performance are:

1. Support: Controls how frequently an itemset must appear to be considered. Higher support speeds up the algorithm but may miss patterns, while lower support increases complexity.
2. Confidence: Measures the strength of association rules. Higher confidence generates fewer, stronger rules, while lower confidence increases the number of rules.
3. Lift: Indicates the strength of a rule's association. A higher lift filters out weaker correlations.
4. Delta: Decreases support incrementally in each iteration. Smaller delta slows down the algorithm but allows more itemsets.
5. Max Number of Rules: Limits the number of rules generated, affecting both performance and the comprehensiveness of the results.