| **Name:** | **James Lewis** |
| --- | --- |
| **Roll No:** | **32** |
| **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

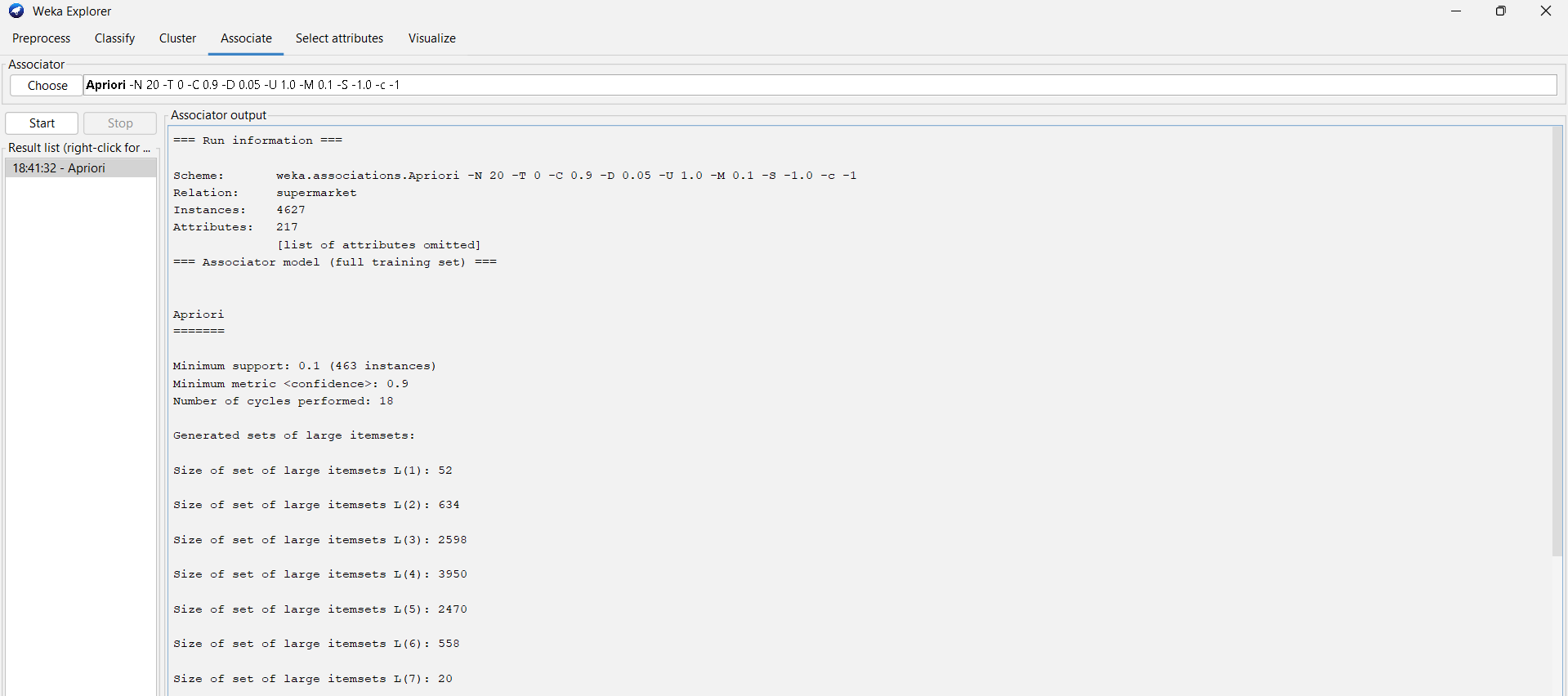
**Frequent Itemset –** An itemset whose support is greater than or equal to minsup threshold.

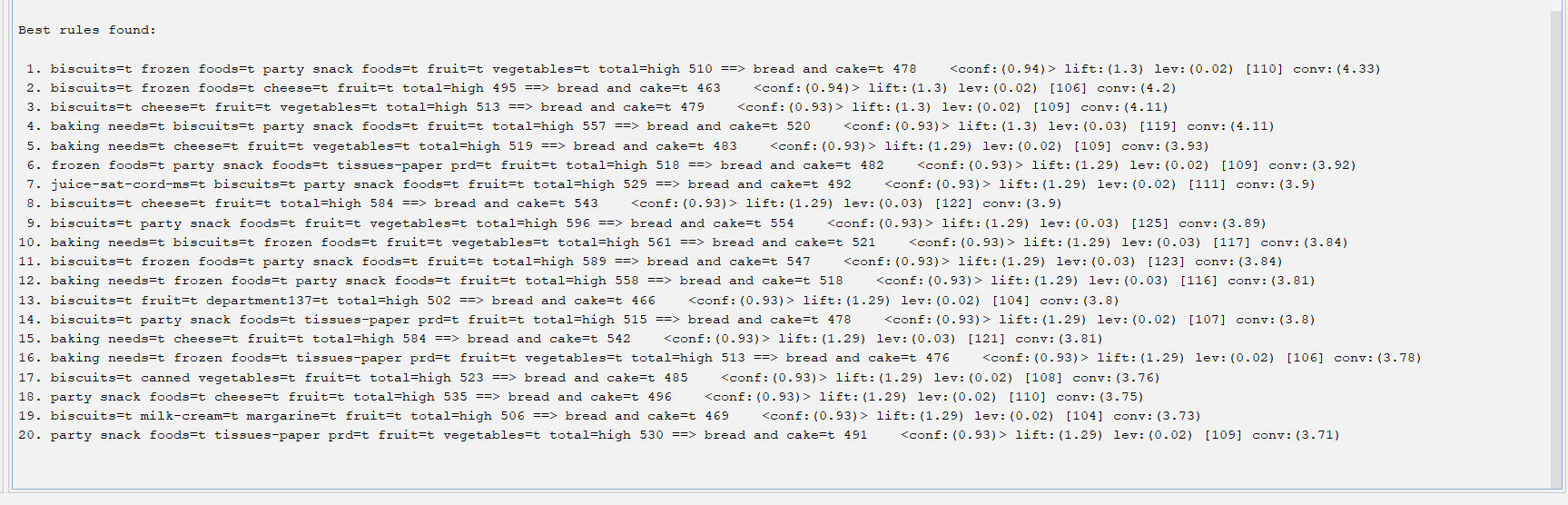
**Association Rule –** An implication expression of the form X Y, where X and Y are any 2 itemsets.

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.**

Main Steps Involved in the Apriori Algorithm:

1. Set Minimum Support Threshold:  
    Define a minimum support value to filter out itemsets that appear less frequently in the dataset.
2. Find Frequent 1-Itemsets:  
    Scan the dataset to identify all single items that satisfy the minimum support threshold. These are called frequent 1-itemsets.
3. Generate Candidate k-Itemsets:  
    Combine the frequent (k−1)-itemsets to form new candidate itemsets of size k.
4. Prune Infrequent Itemsets:  
    Calculate the support of each candidate k-itemset and remove those that do not meet the minimum support requirement.
5. Repeat Until No New Frequent Itemsets Are Found:  
    Continue generating and pruning larger itemsets (2-itemsets, 3-itemsets, etc.) until no more frequent itemsets can be formed.
6. Generate Association Rules:  
    From the frequent itemsets, create rules of the form X ⇒ Y and compute their confidence, which measures how often Y appears in transactions containing X.
7. Filter Strong Rules:  
    Keep only those rules that meet the minimum confidence threshold. These rules represent the strongest and most useful associations in the dataset.

**What are the key parameters in the Apriori algorithm and how do they affect its performance?**  
Key Parameters in the Apriori Algorithm and Their Effects on Performance:

1. Support:  
    Defines how frequently an itemset appears in the dataset.  
   * Higher support values make the algorithm faster but may ignore useful, less frequent patterns.
   * Lower support values include more itemsets but increase computation time.
2. Confidence:  
    Measures the reliability of an association rule (how often Y appears in transactions containing X).  
   * Higher confidence generates fewer but stronger rules.
   * Lower confidence produces more rules, including weaker ones.
3. Lift:  
    Evaluates how much more likely X and Y occur together than if they were independent.  
   * A higher lift indicates a stronger and more meaningful relationship between items.
4. Delta:  
    Controls how much the minimum support decreases in each iteration.  
   * Smaller delta gives finer results but increases runtime.
   * Larger delta speeds up execution but might miss some rules.
5. Number of Rules (NumRules):  
    Specifies how many top rules to generate.  
   * A smaller number gives a concise output.
   * A larger number provides more insights but takes longer to process.

Together, these parameters balance the accuracy, relevance, and efficiency of the Apriori algorithm’s results.