AY: 2025-26

Class:	**TE**	**Semester:**	**V**
Course Code:	**CSC504**	**Course Name:**	**Data Warehousing and Mining**

Name of Student:	**Shravani Sandeep Raut**
Roll No. :	**51**
Experiment No.:	**05**
Title of the Experiment:	**Using open-source tools Implement Association Mining Algorithms.**
Date of Performance:	
Date of Submission:	

Evaluation

Performance Indicator	**Max. Marks**	**Marks Obtained**
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	**Exceed Expectations (EE)**	**Meet Expectations (ME)**	**Meet Expect Below Expectations (BE)**
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by
Name of Faculty : Ms. Neha Raut
Signature :
Date:

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.

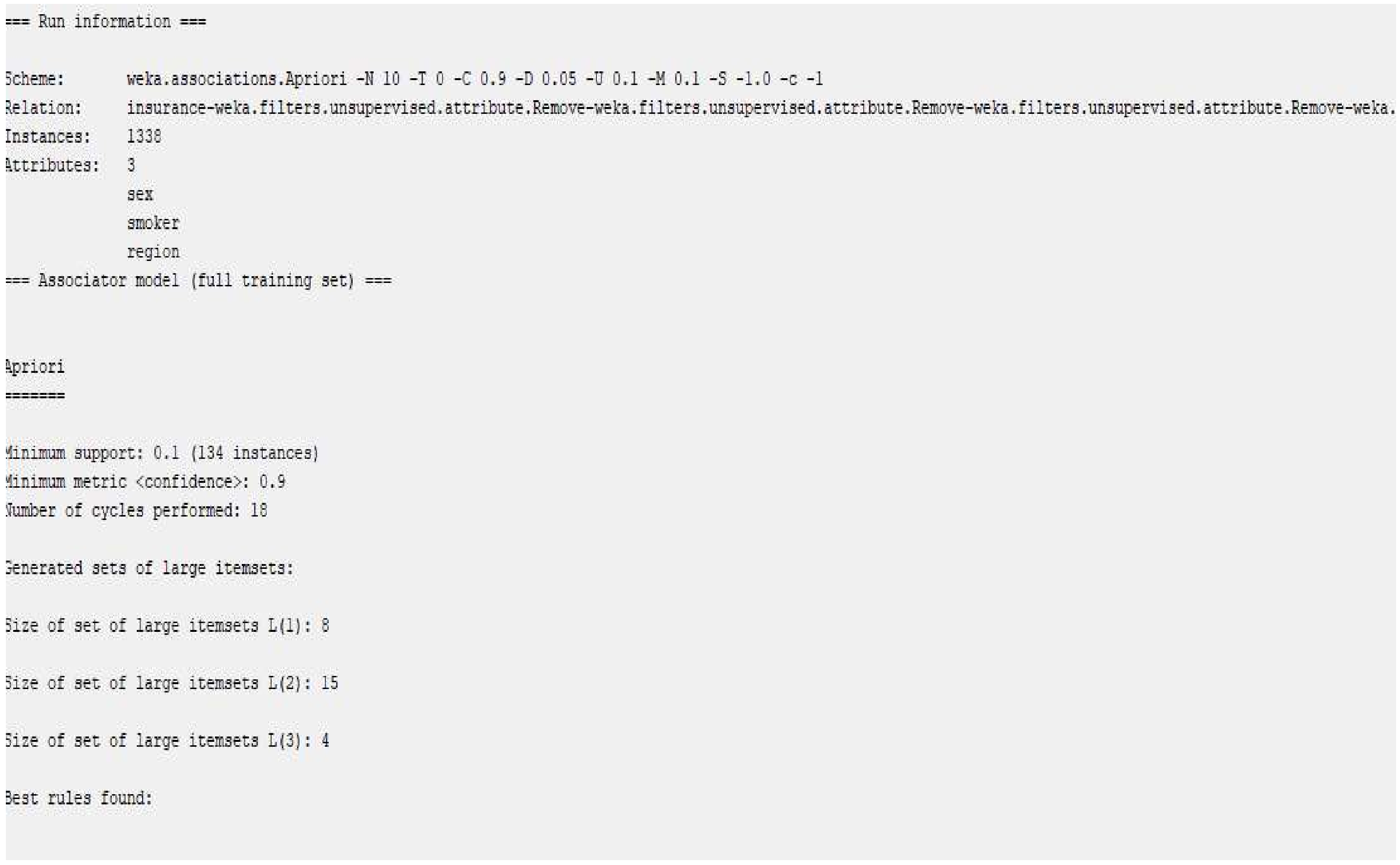
Association Rule – An implication expression of the form X Y, where X and Y are any 2 itemsets.

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:



```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 0.1 -M 0.1 -S -1.0 -c -1
Relation:     insurance-weka.filters.unsupervised.attribute.Remove-weka.filters.unsupervised.attribute.Remove-weka.filters.unsupervised.attribute.Remove-weka.
Instances:    1338
Attributes:   3
              sex
              smoker
              region
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.1 (134 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 15

Size of set of large itemsets L(3): 4

Best rules found:
```

Conclusion:

1) Explain the main steps involved in the Apriori algorithm.
2) What are the key parameters in the Apriori algorithm and how do they affect its performance

Main Steps of the Apriori Algorithm

The **Apriori algorithm** is used for **association rule mining**, i.e., finding frequent itemsets and rules in transactional datasets. Its main steps are:

1. **Generate Candidate Itemsets (C1, C2, …)**
   - Start with all **1-itemsets** (single items).
   - Count how many transactions each item appears in (support count).
2. **Prune Non-Frequent Itemsets**
   - Eliminate candidate itemsets whose **support < minimum support threshold**.
   - The remaining are **frequent itemsets (L1, L2, …)**.
3. **Generate Next-Level Candidates**
   - Use frequent k-itemsets to generate candidate (k+1)-itemsets.
   - Only consider combinations where all subsets are frequent (Apriori property).
4. **Repeat Pruning**
   - Count support for the new candidate itemsets.
   - Prune those below **min support**.
   - Continue until no more candidates remain.
5. **Generate Association Rules**
   - For each frequent itemset, generate **rules of the form X ⇒ Y**.
   - Only keep rules with confidence ≥ **minimum confidence threshold**.
6. **Evaluate Rules**
   - Optionally, compute metrics like **lift, leverage, conviction** to measure strength and usefulness.

Key Parameters in Apriori and Their Effect

Parameter	Meaning	Effect on Performance / Results
Minimum Support (minSup)	Threshold for how often an itemset must appear to be considered frequent	Higher minSup → fewer itemsets, faster execution, fewer rules. Lower minSup → more itemsets, slower execution, more rules, possibly noise.
Minimum Confidence (minConf)	Threshold for rule reliability (confidence of X ⇒ Y)	Higher minConf → rules are stronger but fewer. Lower minConf → more rules, including weaker ones.
Delta (support decrement step)	Amount by which support is reduced in incremental searches	Smaller delta → finer-grained search, more candidate itemsets, slower execution. Larger delta → faster but may miss some itemsets.
Number of Rules to Output (numRules)	Maximum number of rules to display	Limits output for readability. A smaller number shows only top rules; a larger number may include weaker rules.
Upper Bound Support	Maximum support to start from (usually 1.0)	Starting from 100% support ensures algorithm works top-down.
Attributes / Columns	Number of items/attributes in the dataset	More attributes → more potential itemsets → increased computation time. Fewer attributes → fewer rules.