**Data Warehousing and Data Mining.**

**Name : Antuley Aman Siraj.**

**Roll No. : 23CO25.**

**Class : TE-CO**

**Batch : 01**

**Experiment - 09**

**Aim:** Implementation of Association Rule Mining Apriori Algorithm.

**Theory:** The Apriori algorithm is a fundamental approach in association rule mining, aiming to discover frequent itemsets and generate meaningful associations within a dataset. It starts by defining a minimum support threshold (min\_sup), considering itemsets frequent if their occurrence exceeds this threshold. Initially, the algorithm identifies individual items and counts their appearances, establishing 1-itemsets. In the subsequent step, it employs a level-wise candidate generation strategy, combining frequent (k-1)-itemsets to generate candidate k-itemsets. During this process, it prunes candidates that contain infrequent subsets, leveraging the Apriori property, which states that any subset of a frequent itemset must also be frequent. Next, the algorithm scans the dataset to count occurrences and calculate support for each candidate itemset. Those falling below the minimum support threshold are pruned. Once the frequent itemsets are determined, association rules are generated, typically in the form A -> B, where A and B are disjoint itemsets. The confidence of an association rule (A -> B) is calculated as the support of {A, B} divided by the support of A. Further, the rules are pruned based on a minimum confidence threshold (min\_conf). In the end, the output comprises the set of frequent itemsets and the associated pruned association rules meeting the specified minimum support and confidence thresholds. The Apriori algorithm finds extensive applications, particularly in market basket analysis and scenarios where revealing patterns and associations in large datasets is pivotal.

**Program :**

**import java.util.\*;**

**public class Exp9 {**

**public static void main(String ar[]) throws Exception {**

**String ip[] = { "i1i2i5", "i2i4", "i2i3", "i1i2i4", "i1i3", "i2i3", "i1i3", "i1i2i3i5", "i1i2i3" };**

**int minSupp = 2;**

**int i1 = 0, i2 = 0, i3 = 0, i4 = 0, i5 = 0;**

**for (int i = 0; i < ip.length; i++) {**

**if (ip[i].contains("i1"))**

**i1++;**

**if (ip[i].contains("i2"))**

**i2++;**

**if (ip[i].contains("i3"))**

**i3++;**

**if (ip[i].contains("i4"))**

**i4++;**

**if (ip[i].contains("i5"))**

**i5++;**

**}**

**System.out.println("Items \t\t Count");**

**System.out.println("I1 \t\t\t\t " + i1);**

**System.out.println("I2 \t\t\t\t " + i2);**

**System.out.println("I3 \t\t\t\t " + i3);**

**System.out.println("I4 \t\t\t\t " + i4);**

**System.out.println("I5 \t\t\t\t " + i5);**

**int i1i2 = 0, i1i3 = 0, i1i4 = 0, i1i5 = 0, i2i3 = 0, i2i4 = 0, i2i5 = 0, i3i4 = 0, i3i5 = 0, i4i5 = 0;**

**for (int i = 0; i < ip.length; i++) {**

**if (ip[i].contains("i1") && ip[i].contains("i2"))**

**i1i2++;**

**if (ip[i].contains("i1") && ip[i].contains("i3"))**

**i1i3++;**

**if (ip[i].contains("i1") && ip[i].contains("i4"))**

**i1i4++;**

**if (ip[i].contains("i1") && ip[i].contains("i5"))**

**i1i5++;**

**if (ip[i].contains("i2") && ip[i].contains("i3"))**

**i2i3++;**

**if (ip[i].contains("i2") && ip[i].contains("i4"))**

**i2i4++;**

**if (ip[i].contains("i2") && ip[i].contains("i5"))**

**i2i5++;**

**if (ip[i].contains("i3") && ip[i].contains("i4"))**

**i3i4++;**

**if (ip[i].contains("i3") && ip[i].contains("i5"))**

**i3i5++;**

**if (ip[i].contains("i4") && ip[i].contains("i5"))**

**i4i5++;**

**}**

**System.out.println("\nItems \t\t Count");**

**if (i1i2 >= 2)**

**System.out.println("I1I2 \t\t\t " + i1i2);**

**if (i1i3 >= 2)**

**System.out.println("I1I3 \t\t\t " + i1i3);**

**if (i1i4 >= 2)**

**System.out.println("11I4 \t\t\t " + i1i4);**

**if (i1i5 >= 2)**

**System.out.println("I1I5 \t\t\t " + i1i5);**

**if (i2i3 >= 2)**

**System.out.println("I2I3 \t\t\t " + i2i3);**

**if (i2i4 >= 2)**

**System.out.println("I2I4 \t\t\t " + i2i4);**

**if (i2i5 >= 2)**

**System.out.println("I2I5 \t\t\t " + i2i5);**

**if (i3i4 >= 2)**

**System.out.println("I3I4 \t\t\t " + i3i4);**

**if (i3i5 >= 2)**

**System.out.println("I3I5 \t\t\t " + i3i5);**

**if (i4i5 >= 2)**

**System.out.println("I4I5 \t\t\t " + i4i5);**

**int i1i2i3 = 0, i1i2i5 = 0, i1i2i4 = 0, i1i3i5 = 0, i2i3i4 = 0, i2i3i5 = 0, i2i4i5 = 0;**

**for (int i = 0; i < ip.length; i++) {**

**if (ip[i].contains("i1") && ip[i].contains("i2") && ip[i].contains("i3"))**

**i1i2i3++;**

**if (ip[i].contains("i1") && ip[i].contains("i2") && ip[i].contains("i5"))**

**i1i2i5++;**

**if (ip[i].contains("i1") && ip[i].contains("i2") && ip[i].contains("i4"))**

**i1i2i4++;**

**if (ip[i].contains("i1") && ip[i].contains("i3") && ip[i].contains("i5"))**

**i1i3i5++;**

**if (ip[i].contains("i3") && ip[i].contains("i2") && ip[i].contains("i4"))**

**i2i3i4++;**

**if (ip[i].contains("i3") && ip[i].contains("i2") && ip[i].contains("i5"))**

**i2i3i5++;**

**if (ip[i].contains("i4") && ip[i].contains("i2") && ip[i].contains("i5"))**

**i2i4i5++;**

**}**

**System.out.println("\nItems \t\t Count");**

**if (i1i2i3 >= 2)**

**System.out.println("I1I2I3 \t\t\t " + i1i2i3);**

**if (i1i2i5 >= 2)**

**System.out.println("I1I2I5 \t\t\t " + i1i2i5);**

**if (i1i2i4 >= 2)**

**System.out.println("11I2I4 \t\t\t " + i1i2i4);**

**if (i1i3i5 >= 2)**

**System.out.println("11I3I5 \t\t\t " + i1i3i5);**

**if (i2i3i4 >= 2)**

**System.out.println("I2I3I4 \t\t\t " + i2i3i4);**

**if (i2i3i5 >= 2)**

**System.out.println("I2I3I4 \t\t\t " + i2i3i5);**

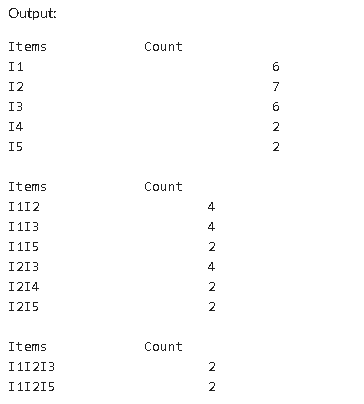
**if (i2i4i5 >= 2)**

**System.out.println("I2I4I5 \t\t\t " + i2i4i5);**

**}**

**}**

**Output :**

****

**Conclusion:**

In conclusion, the Apriori algorithm is a foundational tool in association rule mining. It efficiently identifies frequent item sets within a dataset, aiding in the discovery of meaningful associations. By setting a minimum support threshold, it sifts through the dataset to isolate item sets that meet the specified criteria. The algorithm's process involves candidate itemset generation, pruning, and association rule derivation. It has wide-ranging applications, particularly in market basket analysis and pattern recognition, providing valuable insights and aiding decision-making in diverse domains. The Apriori algorithm's effectiveness lies in its ability to optimize efficiency, making it an essential tool for data mining and analysis.