

10/7/2019

DATA MINING (CS 634-META-19F)

MIDTERM PROJECT -
Under guidance of
Professor Jason Wang

Submitted By -
Name - Garima Tuli
Email Id -
gt98@njit.edu

Table of Contents

Platforms Used.....	2
Instructions for Running the Program	2
Input (Datasets Used)	2
SOURCE CODE FOR APRIORI ALGORITHM IMPLEMENTATION IN JAVA	8
Screenshot of Source Code from Eclipse IDE	15
Output (Screenshots with user specified values for minimum support and confidence on 5 different datasets)	19
Conclusion.....	31

Platforms Used

Programming Language – Java

Development Environment – Eclipse IDE

Operating System – Windows

Hardware – Laptop

Instructions for Running the Program

- This Apriori Implementation is written in Java. To run it, right click on the java file, Click “Run as Java Application”
- You need to change the Dataset name in the program code each time you need to run the program on a different dataset as shown below:

```
String fileName = "dataset\\database1.txt";
String fileName = "dataset\\database2.txt";
String fileName = "dataset\\database3.txt";
String fileName = "dataset\\database4.txt";
String fileName = "dataset\\database5.txt";
```
- Program asks user to enter the minimum support value and minimum confidence value. After user gives input, the association Rules satisfying the minimum Support & Confidence Values entered by user gets generated in the output.

Input (Datasets Used)

Dataset 1

- **10 Items used in Dataset_1 are as follows:**

Coffee,Jam,Tea,Eggs,Honey,Bread,Juice,Cheese,Milk,Butter

- **Transactions for above dataset are as follows:**

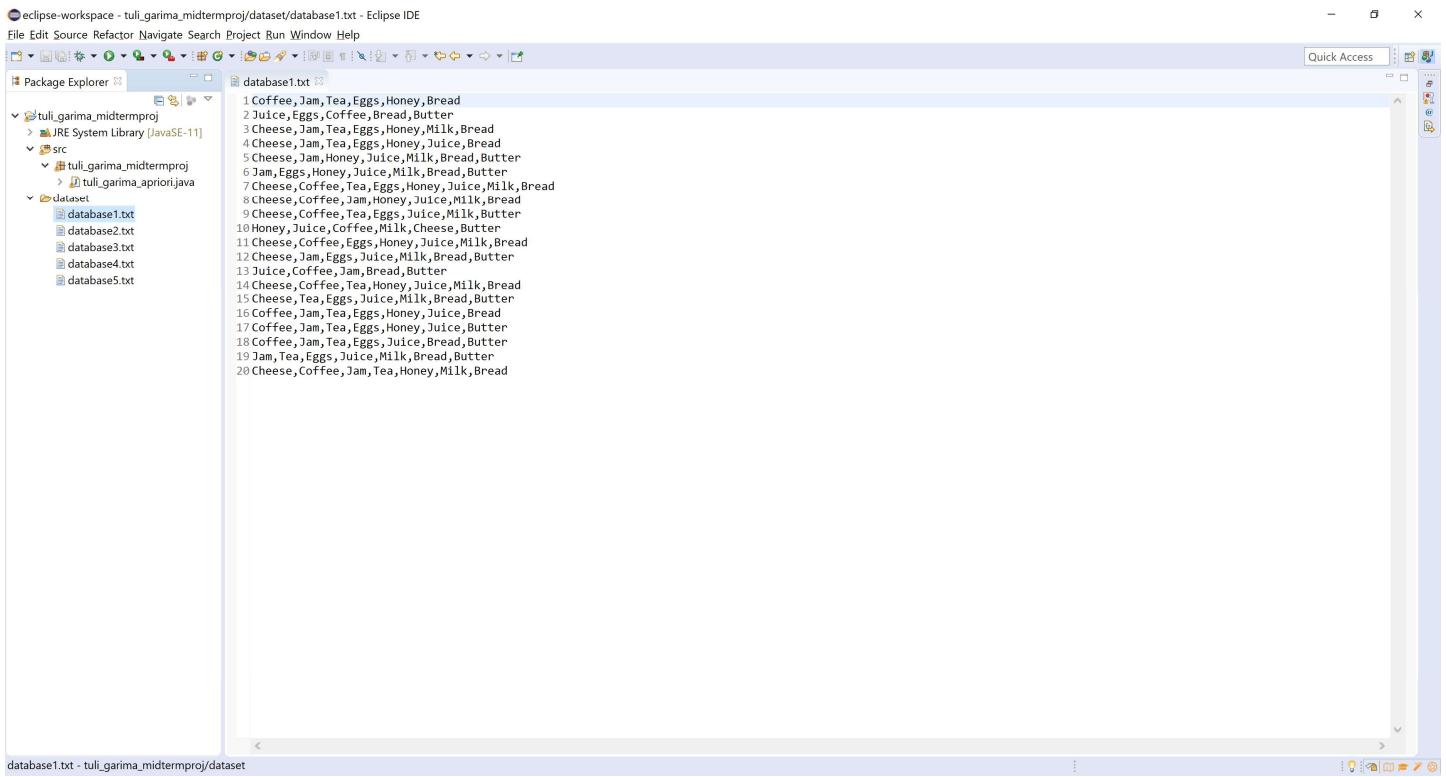
Coffee,Jam,Tea,Eggs,Honey,Bread

Juice,Eggs,Coffee,Bread,Butter

Cheese,Jam,Tea,Eggs,Honey,Milk,Bread

Cheese,Jam,Tea,Eggs,Honey,Juice,Bread
Cheese,Jam,Honey,Juice,Milk,Bread,Butter
Jam,Eggs,Honey,Juice,Milk,Bread,Butter
Cheese,Coffee,Tea,Eggs,Honey,Juice,Milk,Bread
Cheese,Coffee,Jam,Honey,Juice,Milk,Bread
Cheese,Coffee,Tea,Eggs,Juice,Milk,Butter
Honey,Juice,Coffee,Milk,Cheese,Butter
Cheese,Coffee,Eggs,Honey,Juice,Milk,Bread
Cheese,Jam,Eggs,Juice,Milk,Bread,Butter
Juice,Coffee,Jam,Bread,Butter
Cheese,Coffee,Tea,Honey,Juice,Milk,Bread
Cheese,Tea,Eggs,Juice,Milk,Bread,Butter
Coffee,Jam,Tea,Eggs,Honey,Juice,Bread
Coffee,Jam,Tea,Eggs,Honey,Juice,Butter
Coffee,Jam,Tea,Eggs,Juice,Bread,Butter
Jam,Tea,Eggs,Juice,Milk,Bread,Butter
Cheese,Coffee,Jam,Tea,Honey,Milk,Bread

- **Screenshot of Dataset_1 from Eclipse IDE**



Dataset 2

- **10 Items used in Dataset_2 are as follows:**

Orange,Apple,Pumpkin,Cherry,Grapes,Avocado,Tomato,Onion,Lemon,Banana

- **Transactions for above dataset are as follows:**

Orange,Apple,Onion,Grapes,Cherry,Pumpkin,Lemon

Orange,Apple,Tomato,Avocado,Cherry,Pumpkin

Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon

Orange,Tomato,Avocado,Grapes,Cherry,Banana

Apple,Lemon,Avocado,Onion,Grapes,Banana

Orange,Apple,Tomato,Grapes,Cherry,Pumpkin

Orange,Avocado,Onion,Grapes,Cherry,Pumpkin,Banana

Orange,Apple,Tomato,Avocado,Onion,Cherry,Pumpkin,Lemon

Avocado,Lemon,Grapes,Cherry,Banana

Orange,Apple,Tomato,Onion,Grapes,Cherry,Lemon

Orange,Apple,Avocado,Onion,Pumpkin,Banana

Orange,Tomato,Avocado,Grapes,Cherry,Lemon,Banana

Onion,Grapes,Cherry,Pumpkin,Lemon

Orange,Apple,Tomato,Avocado,Onion,Grapes,Cherry,Pumpkin

Orange,Apple,Avocado,Lemon,Banana

Orange,Tomato,Onion,Grapes,Cherry,Pumpkin

Grapes,Cherry,Lemon

Orange,Apple,Tomato,Avocado,Onion,Grapes,Pumpkin

Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon

Orange,Apple,Tomato,Avocado,Cherry,Grapes,Banana

- **Screenshot of Dataset_2 from Eclipse IDE**

The screenshot shows the Eclipse IDE interface. The Package Explorer view on the left displays a project named 'tuli_garima_midtermproj' containing a 'src' folder with a Java file 'tuli_garima_apriori.java', and a 'dataset' folder containing five text files: 'database1.txt', 'database2.txt', 'database3.txt', 'database4.txt', and 'database5.txt'. The 'database2.txt' file is currently selected and open in the central text editor window. The content of the file is a list of 20 transactions, each consisting of a sequence of items separated by commas. The items listed are Orange, Apple, Tomato, Avocado, Cherry, Grapes, Onion, Lemon, and Banana.

```
1 Orange,Apple,Onion,Grapes,Cherry,Pumpkin,Lemon
2 Orange,Apple,Tomato,Avocado,Cherry,Pumpkin
3 Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
4 Orange,Tomato,Avocado,Onion,Grapes,Cherry,Banana
5 Apple,Lemon,Avocado,Onion,Grapes,Cherry,Banana
6 Orange,Apple,Tomato,Grapes,Cherry,Pumpkin
7 Orange,Avocado,Onion,Grapes,Cherry,Pumpkin,Banana
8 Orange,Apple,Tomato,avocado,Onion,Cherry,Pumpkin,Lemon
9 Avocado,Lemon,Grapes,Cherry,Banana
10 Orange,Apple,Tomato,Onion,Grapes,Cherry,Lemon
11 Orange,Apple,Avocado,Onion,Pumpkin,Banana
12 Orange,Tomato,Avocado,Grapes,Cherry,Lemon,Banana
13 Onion,Apple,Cherry,Pumpkin,Lemon
14 Orange,Apple,Tomato,Avocado,Onion,Grapes,Cherry,Pumpkin
15 Orange,Apple,Avocado,Lemon,Banana
16 Orange,Tomato,Onion,Grapes,Cherry,Pumpkin
17 Grapes,Cherry,Lemon
18 Orange,Apple,Tomato,Avocado,Onion,Grapes,Pumpkin
19 Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
20 Orange,Apple,Tomato,Avocado,Cherry,Grapes,Banana
```

Dataset 3

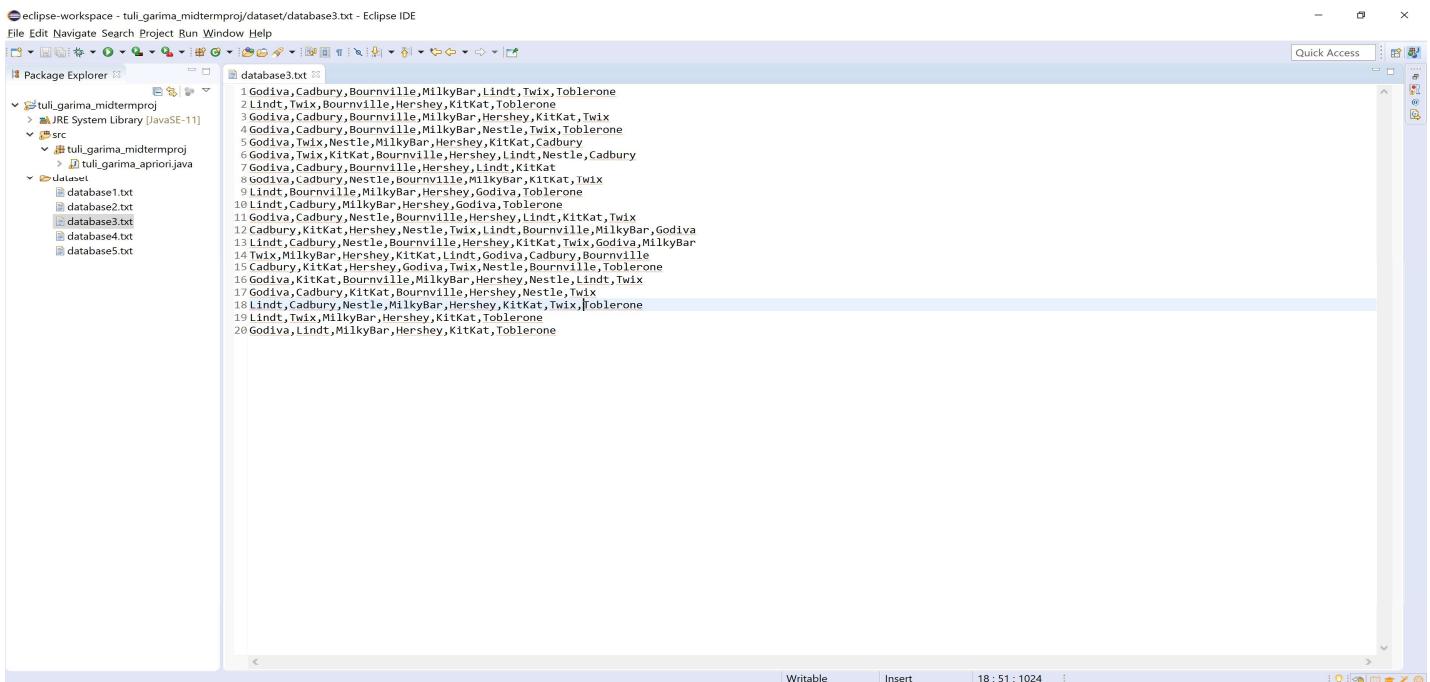
- **10 Items used in Dataset_3 are as follows:**

Lindt,Bournville,Godiva,Twix,Cadbury,KitKat,Hershey,MilkyBar,Toblerone,Nestle

- **Transactions for above dataset are as follows:**

Godiva,Cadbury,Bournville,MilkyBar,Lindt,Twix,Toblerone
Lindt,Twix,Bournville,Hershey,KitKat,Toblerone
Godiva,Cadbury,Bournville,MilkyBar,Hershey,KitKat,Twix
Godiva,Cadbury,Bournville,MilkyBar,Nestle,Twix,Toblerone
Godiva,Twix,Nestle,MilkyBar,Hershey,KitKat,Cadbury
Godiva,Twix,KitKat,Bournville,Hershey,Lindt,Nestle,Cadbury
Godiva,Cadbury,Bournville,Hershey,Lindt,KitKat
Godiva,Cadbury,Nestle,Bournville,MilkyBar,KitKat,Twix
Lindt,Bournville,MilkyBar,Hershey,Godiva,Toblerone
Lindt,Cadbury,MilkyBar,Hershey,Godiva,Toblerone
Godiva,Cadbury,Nestle,Bournville,Hershey,Lindt,KitKat,Twix
Cadbury,KitKat,Hershey,Nestle,Twix,Lindt,Bournville,MilkyBar,Godiva
Lindt,Cadbury,Nestle,Bournville,Hershey,KitKat,Twix,Godiva,MilkyBar
Twix,MilkyBar,Hershey,KitKat,Lindt,Godiva,Cadbury,Bournville
Cadbury,KitKat,Hershey,Godiva,Twix,Nestle,Bournville,Toblerone
Godiva,KitKat,Bournville,MilkyBar,Hershey,Nestle,Lindt,Twix
Godiva,Cadbury,KitKat,Bournville,Hershey,Nestle,Twix
Lindt,Cadbury,Nestle,MilkyBar,Hershey,KitKat,Twix,Toblerone
Lindt,Twix,MilkyBar,Hershey,KitKat,Toblerone
Godiva,Lindt,MilkyBar,Hershey,KitKat,Toblerone

- **Screenshot of Dataset_3 from Eclipse IDE**



Dataset 4

- **10 Items used in Dataset_4 are as follows:**

Printer,Bluetooth,USB,Laptop,Monitor,Keyboard,Mouse,HPIink,MemoryCard,EyeGlass

- **Transactions for above dataset are as follows:**

Printer,USB,Laptop,Mouse,HPIink,EyeGlass

Printer,Laptop,Mouse,HPIink,EyeGlass

Printer,USB,Monitor,Keyboard,Mouse,HPIink,Laptop

Printer,Monitor,Keyboard,Mouse,HPIink,Laptop

Bluetooth,USB,Laptop,MemoryCard,EyeGlass

Printer,Monitor,Keyboard,Mouse,HPIink,Laptop

Printer,USB,Laptop,HPIink,EyeGlass

Printer,Laptop,Mouse,HPIink,EyeGlass

Bluetooth,USB,Laptop,MemoryCard,EyeGlass

Bluetooth,Monitor,Keyboard,Mouse,Laptop

Bluetooth,USB,Laptop,MemoryCard,EyeGlass

Printer,Monitor,Keyboard,Mouse,HPIink,EyeGlass

Printer,Laptop,HPIink,EyeGlass

Printer,USB,HPIink,Laptop

Bluetooth,Laptop,Mouse,MemoryCard,EyeGlass

Printer,Monitor,Keyboard,Mouse,HPIink

Printer,USB,Monitor,Keyboard,Mouse,HPIink

Printer,Laptop,Mouse,HPIink,EyeGlass

Bluetooth,USB,MemoryCard,Printer,HPIink,Laptop

Bluetooth,USB,MemoryCard,Printer,Laptop,HPIink,EyeGlass

- **Screenshot of Dataset_4 from Eclipse IDE**

```
eclipse-workspace - tuli_garima_midtermproj/dataset/database4.txt - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
src
  tuli_garima_apriori.java
dataset
  database1.txt
  database2.txt
  database3.txt
  database4.txt
  database5.txt
database4.txt
1 Printer,USB,Laptop,Mouse,HPIink,EyeGlass
2 Printer,Laptop,Mouse,HPIink,EyeGlass
3 Printer,Monitor,Keyboard,Mouse,HPIink,Laptop
4 Printer,Monitor,Keyboard,Mouse,HPIink,EyeGlass
5 Bluetooth,USB,Laptop,MemoryCard,EyeGlass
6 Printer,Monitor,Keyboard,Mouse,HPIink,EyeGlass
7 Printer,USB,Laptop,HPIink,EyeGlass
8 Printer,Laptop,Mouse,HPIink,EyeGlass
9 Bluetooth,USB,Laptop,MemoryCard,EyeGlass
10 Bluetooth,Monitor,Keyboard,Mouse,Laptop
11 Bluetooth,USB,Laptop,MemoryCard,EyeGlass
12 Printer,Monitor,Keyboard,Mouse,HPIink,EyeGlass
13 Printer,Laptop,HPIink,EyeGlass
14 Printer,USB,HPIink,Laptop
15 Bluetooth,Laptop,Mouse,MemoryCard,EyeGlass
16 Printer,Monitor,Keyboard,Mouse,EyeGlass
17 Printer,USB,Monitor,Keyboard,Mouse,HPIink
18 Printer,Laptop,Mouse,HPIink,EyeGlass
19 Bluetooth,USB,MemoryCard,Printer,HPIink,Laptop
20 Bluetooth,USB,MemoryCard,Printer,Laptop,HPIink,EyeGlass
```

Dataset 5

- **10 Items used in Dataset_5 are as follows:**

ToothPaste,ToothBrush,Handwash,Moisturizer,NailPolish,LaundryDetergent,PaperTowels,Shampoo,Conditioner,HairOil

- **Transactions for above dataset are as follows:**

ToothPaste,ToothBrush,LaundryDetergent,Shampoo,HairOil

Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil

ToothPaste,ToothBrush,LaundryDetergent

ToothPaste,ToothBrush,LaundryDetergent

Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil

Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil

ToothPaste,ToothBrush,LaundryDetergent,HairOil

Handwash,Moisturizer,LaundryDetergent,HairOil

ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil

ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil

Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil

ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil

Handwash,Moisturizer,LaundryDetergent,HairOil

ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil

Handwash,Moisturizer,LaundryDetergent,HairOil

ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil

ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,

ToothPaste,Handwash,Moisturizer,NailPolish,LaundryDetergent

Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil

Moisturizer,NailPolish,PaperTowels,HairOil

ToothPaste,Moisturizer,NailPolish,LaundryDetergent,Shampoo,Conditioner,HairOil

ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,HairOil

- **Screenshot of Dataset_5 from Eclipse IDE**

The screenshot shows the Eclipse IDE interface. The Package Explorer view on the left displays a project structure with a 'src' folder containing 'tuli_garima_midtermproj' and 'tuli_garima_apriori.java'. The 'Database' folder contains five files: 'database1.txt', 'database2.txt', 'database3.txt', 'database4.txt', and 'database5.txt'. The 'database5.txt' file is currently selected and open in the central text editor window. The editor shows a list of 20 transactions, each consisting of a sequence of items separated by commas. The items listed are: ToothPaste, ToothBrush, LaundryDetergent, Shampoo, HairOil, Handwash, Moisturizer, NailPolish, PaperTowels, Shampoo, Conditioner, HairOil.

```
1 ToothPaste,ToothBrush,LaundryDetergent,Shampoo,HairOil
2 Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
3 ToothPaste,ToothBrush,LaundryDetergent
4 ToothPaste,ToothBrush,LaundryDetergent
5 Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
6 Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
7 ToothPaste,ToothBrush,LaundryDetergent,HairOil
8 Handwash,Moisturizer,LaundryDetergent,HairOil
9 ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
10 Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,HairOil
11 Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
12 ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
13 Handwash,Moisturizer,LaundryDetergent,HairOil
14 ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
15 ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,
16 ToothPaste,Handwash,Moisturizer,NailPolish,LaundryDetergent
17 Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
18 Moisturizer,NailPolish,PaperTowels,HairOil
19 ToothPaste,Moisturizer,NailPolish,LaundryDetergent,Shampoo,Conditioner,HairOil
20 ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,HairOil
```

SOURCE CODE FOR APRIORI ALGORITHM IMPLEMENTATION IN JAVA

```
package tuli_garima_midtermproj;

import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class tuli_garima_apriori {

    // ArrayList to store unique items from all transactions
    private static ArrayList<String> uniqueItemsList = new ArrayList<String>();

    // Data Structures to store all transactions
    private static ArrayList<HashSet<String>> transactionAllItemsList
        = new ArrayList<HashSet<String>>();

    // HashMap to store count per item in each transaction
    private static HashMap<String, Integer> currentCandidacyList = new HashMap<String,
    Integer>();

    // ArrayList to store frequent items
    private static ArrayList<String> frequentItemsList = new ArrayList<String>();

    // HashMap to store items whose count/no of transactions >= support
    private static HashMap<String, Integer> frequentItemsForAssociations
        = new HashMap<String, Integer>();

    // ArrayList to store frequent List after first iteration
    private static ArrayList<String> frequentItemsListAfterFirstIteration
        = new ArrayList<String>();

    // ArrayList to store each possible combination from frequent list
    private static ArrayList<String> combinationItemsList = new ArrayList<String>();

    // Main Function

    public static void main(String args[]) {
        System.out.println("\nWelcome to Garima Tuli's Apriori Algorithm\n");
        System.out.println("*****");
        System.out.println("\nFollowing is the Input Dataset:");
        System.out.println();

        // Call Function to read dataset from file
        readDataFromFile();
        System.out.println("\n*****");
        System.out.println();
    }
}
```

```

// Input Minimum Support Value from User
    System.out.println("Enter Minimum Support Value: ");
    Scanner sc = new Scanner(System.in);
    double support = sc.nextDouble();
    System.out.println();

// Input Minimum Confidence Value from User
    System.out.println("Enter Minimum Confidence Value: ");
    double confidence = sc.nextDouble();
    System.out.println("\n*****");
    System.out.println("\nAssociation Rules satisfying the minimum Support
                      & Confidence Values entered by user above are as follows:");

// Logic to generate candidacy List, frequent item set and associations

    int itemCount = 1;

    if (itemCount == 1) {
        candidacyListforOneorAllCombinations(uniqueItemList, itemCount);
        frequentList(support);
        frequentItemsListAfterFirstIteration
            = new ArrayList<String>(frequentItemsList);
        itemCount = itemCount + 1;
    }

    if (itemCount == 2) {
        String[] itemCountArray = new String[itemCount];
        combinationItemsList.clear();

        ListWithTwoItems(frequentItemsListAfterFirstIteration,
                        frequentItemsListAfterFirstIteration.size(),
                        itemCount, 0, itemCountArray, 0);

        candidacyListforOneorAllCombinations(combinationItemsList, itemCount);
        frequentList(support);
        itemCount = itemCount + 1;
    }

    if (itemCount == 3) {
        while (frequentItemsList.size() != 0) {

            ListWithThreeOrMoreItems(frequentItemsList, itemCount);
            candidacyListforOneorAllCombinations(combinationItemsList, itemCount);
            frequentList(support);
            itemCount = itemCount + 1;
        }
    }

    findAllAssociations(frequentItemsForAssociations, confidence);
}

```

```

// Read Data from File

public static void readDataFromFile() {

    // Replace different file name here for different data set
    String fileName = "dataset\\database1.txt";
    // Local variable for each line
    String eachLine = null;
    // Unique Items
    HashSet<String> uniqueItems = new HashSet<String>();

    try {

        FileReader fileReader = new FileReader(fileName);
        BufferedReader bufferedReader = new BufferedReader(fileReader);

        while ((eachLine = bufferedReader.readLine()) != null) {

            // Local Items to form transactionItemsList
            HashSet<String> localItems = new HashSet<String>();
            String formedList[] = eachLine.split(",");
            for (int i = 0; i < formedList.length; i++) {
                localItems.add(formedList[i]);
                uniqueItems.add(formedList[i]);
            }

            System.out.println(eachLine);

            // transactionItemsList will contain set of all itemset
            transactionAllItemsList.add(localItems);
        }

    } catch (Exception e) {
        System.out.println("In to Exception" + e);
    }

    // finalList is arraylist of unique items from itemset
    uniqueItemsList = new ArrayList<String>(uniqueItems);
}

// Function to form candidacy list (for one or all possible combinations in all transactions)

public static void candidacyListforOneorAllCombinations(ArrayList<String> receivedUniqueItemsList, int receivedCount) {

    if (receivedCount == 1) {
        currentCandidacyList.clear();

        // Loop through final List which is a unique list (finalList)
        // and all items (transactionItemsList)
}

```

```

// to generate count per item in each transaction

for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
    String oneItem = receivedUniqueItemsList.get(i);

    for (int j = 0; j < transactionAllItemsList.size(); j++) {
        if (transactionAllItemsList.get(j).contains(oneItem)) {
            if (currentCandidacyList.containsKey(oneItem)) {
                int counter = currentCandidacyList.get(oneItem);
                currentCandidacyList.put(oneItem, ++counter);
            } else {
                currentCandidacyList.put(oneItem, 1);
            }
        }
    }
} else {
    currentCandidacyList.clear();
    for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
        String st = receivedUniqueItemsList.get(i);
        String[] splitString = st.split(" ");
        for (int j = 0; j < transactionAllItemsList.size(); j++) {
            boolean fl = true;
            for (int k = 0; k < splitString.length; k++) {

                if (!transactionAllItemsList.get(j).contains(splitString[k]))
                {
                    fl = false;
                    break;
                }
            }
            if (fl == true) {
                if (currentCandidacyList.containsKey(st)) {
                    int value = currentCandidacyList.get(st);
                    currentCandidacyList.put(st, ++value);
                } else {
                    currentCandidacyList.put(st, 1);
                }
            }
        }
    }
}
}

// Function to form ArrayList for two items per transaction

public static void ListWithTwoItems(ArrayList<String> receivedFrequentItemsList,
int receivedFrequentItemsListSize, int itemCountSize, int i,
String[] itemCountArray, int j) {

if (i == itemCountSize) {
    String st = "";
    for (int k = 0; k < itemCountSize; k++) {

```

```

        st = st + itemCountArray[k] + " ";
    }
    combinationItemsList.add(st);
    return;
}
if (j >= receivedFrequentItemsListSize) {
    return;
}

itemCountArray[i] = receivedFrequentItemsList.get(j);

ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize,
itemCountSize, i + 1, itemCountArray,j + 1);

ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize,
itemCountSize, i, itemCountArray,j + 1);

}

// Function to form ArrayList for three or more items per transaction

private static void ListWithThreeOrMoreItems(ArrayList<String> receivedFrequentItemsList,
int receivedCount) {
    combinationItemsList.clear();

    for (int i = 0; i < receivedFrequentItemsList.size(); i++) {
        String eachString = receivedFrequentItemsList.get(i).trim();
        String st = eachString.substring(0, eachString.lastIndexOf(" "));

        for (int j = i + 1; j < receivedFrequentItemsList.size(); j++) {
            String eachString1 = receivedFrequentItemsList.get(j).trim();
            String st1 = eachString1.substring(0, eachString1.lastIndexOf(" "));
            if (st.equals(st1)) {
                combinationItemsList.add(eachString + " " +
eachString1.substring(eachString1.lastIndexOf(" ") + 1));
            }
        }
    }
}

// Function to form frequent List and frequent List for association
public static void frequentList(double receivedSupport) {
    frequentItemsList.clear();
    receivedSupport = receivedSupport / 100;
    Set<String> items = currentCandidacyList.keySet();

    for (String eachItem : items) {
        if ((double) currentCandidacyList.get(eachItem)
        / transactionAllItemsList.size() >= receivedSupport) {
            frequentItemsList.add(eachItem);
        }
    }
}

```

```

        frequentItemsForAssociations.put(eachItem,
        currentCandidacyList.get(eachItem));
    }
}

// Function to form all the associations
public static void findAllAssociations(HashMap<String, Integer>
receivedFrequentItemsForAssociations, double confidenceFromUser) {

    for (String st : receivedFrequentItemsForAssociations.keySet()) {
        String[] splitArray = st.split(" ");

        if (splitArray.length == 2) {
            printAssociationBasedOnConfidence(st,
            receivedFrequentItemsForAssociations.get(st),
            splitArray[0], splitArray[1], confidenceFromUser);

            printAssociationBasedOnConfidence(st,
            receivedFrequentItemsForAssociations.get(st),
            splitArray[1], splitArray[0], confidenceFromUser);
        } else if (splitArray.length == 3) {

            for (int i = 0; i < splitArray.length; i++) {
                if (i == 0) {
                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i], splitArray[i + 1] + " " +
                    splitArray[i + 2], confidenceFromUser);

                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i + 1] + " " +
                    splitArray[i + 2], splitArray[i], confidenceFromUser);
                } else if (i == 1) {

                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i], splitArray[i - 1] + " " +
                    splitArray[i + 1], confidenceFromUser);

                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i - 1] + " " + splitArray[i + 1],
                    splitArray[i], confidenceFromUser);
                } else {
                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i], splitArray[i - 2] + " " +
                    splitArray[i - 1], confidenceFromUser);

                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                    splitArray[i - 2] + " " + splitArray[i - 1],
                    splitArray[i], confidenceFromUser);
                }
            }
        }
    }
}

```

```

        }
    }
}

// Function to print associations for confidence greater than user provided minimum confidence
value

public static void printAssociationBasedOnConfidence(String eachFrequentListItem,
int countOfEachFrequentListItem, String leftHandSide,
String rightHandSide, double confidenceFromUser) {

    int count = 0;
    String[] splitArray = leftHandSide.split(" ");
    int length = splitArray.length;

    for (int i = 0; i < transactionAllItemsList.size(); i++) {
        boolean fl = true;
        for (int j = 0; j < length; j++) {
            if(!transactionAllItemsList.get(i).contains(splitArray[j]))
            {
                fl = false;
                break;
            }
        }
        if (fl == true) {
            count = count + 1;
        }
    }

    double confidenceOfEachAssociation
    = (double) countOfEachFrequentListItem / (count * 1.0);

    double percentageUserConfidence = (double) confidenceFromUser / 100;

    if (confidenceOfEachAssociation >= percentageUserConfidence) {
        System.out.println(leftHandSide + "->" + rightHandSide);
    }
}
}

```

Screenshot of Source Code from Eclipse IDE

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```
1 package tuli_garima_midtermproj;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.util.ArrayList;
6 import java.util.HashMap;
7 import java.util.HashSet;
8 import java.util.Scanner;
9 import java.util.Set;
10
11 public class tuli_garima_apriori {
12
13     // ArrayList to store unique items from all transactions
14     private static ArrayList<String> uniqueItemList = new ArrayList<String>();
15
16     // Data Structures to store all transactions
17     private static ArrayList<HashSet<String>> transactionAllItemList = new ArrayList<HashSet<String>>();
18
19     // HashMap to store count per item in each transaction
20     private static HashMap<String, Integer> currentCandidacyList = new HashMap<String, Integer>();
21
22     // ArrayList to store frequent items
23     private static ArrayList<String> frequentItemList = new ArrayList<String>();
24
25     // HashMap to store items whose count/no of transactions >= support
26     private static HashMap<String, Integer> frequentItemsForAssociations = new HashMap<String, Integer>();
27
28     // ArrayList to store frequent List after first iteration
29     private static ArrayList<String> frequentItemListAfterFirstIteration = new ArrayList<String>();
30
31     // ArrayList to store each possible combination from frequent list
32     private static ArrayList<String> combinationItemList = new ArrayList<String>();
33
34     // Main Function
35
36     public static void main(String args[]) {
37         System.out.println("\nWelcome to Garima Tuli's Apriori Algorithm\n");
38         System.out.println("*****");
39         System.out.println("\nFollowing is the Input Dataset:");
40         System.out.println();
41
42         // Call Function to read dataset from file
43         readDataFromFile();
44         System.out.println("\n*****");
45         System.out.println();
46     }
47
48     // Input Minimum Support Value from User
49     Scanner sc = new Scanner(System.in);
50     double support = sc.nextDouble();
51     System.out.println();
52
53     // Input Minimum Confidence Value from User
54     System.out.println("Enter Minimum Confidence Value: ");
55     double confidence = sc.nextDouble();
56     System.out.println("*****");
57     System.out.println(
58         "\nAssociation Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:");
59
60     // Logic to generate candidacy List, frequent item set and associations
61
62     int itemCount = 1;
63
64     if (itemCount == 1) {
65         candidacyListForOneOrAllCombinations(uniqueItemList, itemCount);
66         frequentList(support);
67         frequentItemListAfterFirstIteration = new ArrayList<String>(frequentItemList);
68         itemCount = itemCount + 1;
69     }
70
71     if (itemCount == 2) {
72         String[] itemCountArray = new String[itemCount];
73         combinationItemList.clear();
74
75         listWithTwoItems(frequentItemListAfterFirstIteration, frequentItemListAfterFirstIteration.size(),
76                         itemCount, 0, itemCountArray, 0);
77
78         candidacyListForOneOrAllCombinations(combinationItemList, itemCount);
79         frequentList(support);
80         itemCount = itemCount + 1;
81     }
82
83     if (itemCount == 3) {
84         while (frequentItemList.size() != 0) {
85
86             listWithThreeOrMoreItems(frequentItemList, itemCount);
87             candidacyListForOneOrAllCombinations(combinationItemList, itemCount);
88             frequentList(support);
89             itemCount = itemCount + 1;
90         }
91     }
92 }
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```
47     // Input Minimum Support Value from User
48     Scanner sc = new Scanner(System.in);
49     double support = sc.nextDouble();
50     System.out.println();
51
52     // Input Minimum Confidence Value from User
53     double confidence = sc.nextDouble();
54     System.out.println("*****");
55     System.out.println(
56         "\nAssociation Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:");
57
58     // Logic to generate candidacy List, frequent item set and associations
59
60     int itemCount = 1;
61
62     if (itemCount == 1) {
63         candidacyListForOneOrAllCombinations(uniqueItemList, itemCount);
64         frequentList(support);
65         frequentItemListAfterFirstIteration = new ArrayList<String>(frequentItemList);
66         itemCount = itemCount + 1;
67     }
68
69     if (itemCount == 2) {
70         String[] itemCountArray = new String[itemCount];
71         combinationItemList.clear();
72
73         listWithTwoItems(frequentItemListAfterFirstIteration, frequentItemListAfterFirstIteration.size(),
74                         itemCount, 0, itemCountArray, 0);
75
76         candidacyListForOneOrAllCombinations(combinationItemList, itemCount);
77         frequentList(support);
78         itemCount = itemCount + 1;
79     }
80
81     if (itemCount == 3) {
82         while (frequentItemList.size() != 0) {
83
84             listWithThreeOrMoreItems(frequentItemList, itemCount);
85             candidacyListForOneOrAllCombinations(combinationItemList, itemCount);
86             frequentList(support);
87             itemCount = itemCount + 1;
88         }
89     }
90 }
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```
1 package tuli_garima_midtermproj;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import java.util.HashSet;
7 import java.util.StringTokenizer;
8
9 public class tuli_garima_apriori {
10     public static void main(String[] args) {
11         findAllAssociations(frequentItemsForAssociations, confidence);
12     }
13
14     // Read Data from File
15
16     public static void readDataFromFile() {
17
18         // Replace different file name here for different data set
19         String fileName = "dataset\\database1.txt";
20         // Local variable for each line
21         String eachLine = null;
22         // Unique Items
23         HashSet<String> uniqueItems = new HashSet<String>();
24
25         try {
26
27             FileReader fileReader = new FileReader(fileName);
28             BufferedReader bufferedReader = new BufferedReader(fileReader);
29
30             while ((eachLine = bufferedReader.readLine()) != null) {
31
32                 // Local Items to form transactionItemsList
33                 HashSet<String> localItems = new HashSet<String>();
34                 String formedList[] = eachLine.split(",");
35
36                 for (int i = 0; i < formedList.length; i++) {
37                     localItems.add(formedList[i]);
38                     uniqueItems.add(formedList[i]);
39                 }
40
41                 System.out.println(eachLine);
42
43                 // transactionItemsList will contain set of all itemset
44                 transactionAllItemsList.add(localItems);
45             }
46
47         } catch (Exception e) {
48             System.out.println("In to Exception" + e);
49         }
50
51         // finalList is arraylist of unique items from itemset
52         uniqueItemsList = new ArrayList<String>(uniqueItems);
53     }
54 }
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```
1 package tuli_garima_midtermproj;
2
3 import java.util.ArrayList;
4 import java.util.HashMap;
5 import java.util.HashSet;
6 import java.util.List;
7 import java.util.Map;
8
9 public class tuli_garima_apriori {
10     public static void candidacyListForOneOrAllCombinations(ArrayList<String> receivedUniqueItemsList,
11         int receivedCount) {
12
13         // Function to form candidacy list (for one or all possible combinations in all
14         // transactions)
15
16         if (receivedCount == 1) {
17             currentCandidacyList.clear();
18
19             // Loop through final List which is a unique list (finalList)
20             // and all items (transactionItemsList)
21             // to generate count per item in each transaction
22
23             for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
24                 String oneItem = receivedUniqueItemsList.get(i);
25
26                 for (int j = 0; j < transactionAllItemsList.size(); j++) {
27                     if (transactionAllItemsList.get(j).contains(oneItem)) {
28                         if (currentCandidacyList.containsKey(oneItem)) {
29                             int counter = currentCandidacyList.get(oneItem);
30                             currentCandidacyList.put(oneItem, ++counter);
31                         } else {
32                             currentCandidacyList.put(oneItem, 1);
33                         }
34                     }
35                 }
36             }
37         } else {
38             currentCandidacyList.clear();
39             for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
40                 String st = receivedUniqueItemsList.get(i);
41                 String[] splitString = st.split(" ");
42
43                 for (int j = 0; j < transactionAllItemsList.size(); j++) {
44                     boolean fl = true;
45                     for (int k = 0; k < splitString.length; k++) {
46                         if (!transactionAllItemsList.get(j).contains(splitString[k])) {
47                             fl = false;
48                             break;
49                         }
50                     }
51
52                     if (fl == true) {
53                         if (currentCandidacyList.containsKey(st)) {
54                             int value = currentCandidacyList.get(st);
55                             currentCandidacyList.put(st, ++value);
56                         } else {
57                         currentCandidacyList.put(st, 1);
58                         }
59                     }
60                 }
61             }
62         }
63     }
64 }
```

This screenshot shows the Eclipse IDE interface with the package explorer open, displaying the project structure and the contents of the tuli_garima_apriori.java file.

```
182         currentCandidacyList.put(st, 1);
183     }
184   }
185 }
186 }
187 }
188 }
189 }
190 // Function to form ArrayList for two items per transaction
191 public static void ListWithTwoItems(ArrayList<String> receivedFrequentItemsList, int receivedFrequentItemsListSize,
192   int itemCountSize, int i, String[] itemCountArray, int j) {
193
194   if (i == itemCountSize) {
195     String st = "";
196     for (int k = 0; k < itemCountSize; k++) {
197       st = st + itemCountArray[k] + " ";
198     }
199     combinationItemsList.add(st);
200     return;
201   }
202   if (j >= receivedFrequentItemsListSize) {
203     return;
204   }
205
206   itemCountArray[i] = receivedFrequentItemsList.get(j);
207
208   ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize, itemCountSize, i + 1, itemCountArray,
209     j + 1);
210
211   ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize, itemCountSize, i, itemCountArray,
212     j + 1);
213
214 }
215
216 // Function to form ArrayList for three or more items per transaction
217
218 private static void ListWithThreeOrMoreItems(ArrayList<String> receivedFrequentItemsList, int receivedCount) {
219   combinationItemsList.clear();
220
221   for (int i = 0; i < receivedFrequentItemsList.size(); i++) {
222     String eachString = receivedFrequentItemsList.get(i).trim();
223     String st = eachString.substring(0, eachString.lastIndexOf(" "));
224
225     for (int j = i + 1; j < receivedFrequentItemsList.size(); j++) {
226
227       String eachString1 = receivedFrequentItemsList.get(j).trim();
228       String st1 = eachString1.substring(0, eachString1.lastIndexOf(" "));
229       if (st.equals(st1)) {
230         combinationItemsList
231           .add(eachString + " " + eachString1.substring(eachString1.lastIndexOf(" ") + 1));
232
233     }
234   }
235
236
237 // Function to form frequent List and frequent List for association
238 public static void frequentList(double receivedSupport) {
239   frequentItemList.clear();
240   receivedSupport = receivedSupport / 100;
241   Set<String> items = currentCandidacyList.keySet();
242
243   for (String eachItem : items) {
244     if ((double) currentCandidacyList.get(eachItem) / transactionAllItemsList.size() >= receivedSupport) {
245       frequentItemList.add(eachItem);
246       frequentItemsForAssociations.put(eachItem, currentCandidacyList.get(eachItem));
247     }
248   }
249
250
251 // Function to form all the associations
252 public static void findAllAssociations(HashMap<String, Integer> receivedFrequentItemsForAssociations,
253   double confidenceFromUser) {
254
255   for (String st : receivedFrequentItemsForAssociations.keySet()) {
256     String[] splitArray = st.split(" ");
257
258     if (splitArray.length == 2) {
259       printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[0],
260         splitArray[1], confidenceFromUser);
261
262       printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[0],
263         splitArray[0], confidenceFromUser);
264     } else if (splitArray.length == 3) {
265
266       for (int i = 0; i < splitArray.length; i++) {
267         if (i == 0) {
268           printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
269             splitArray[i + 1] + " " + splitArray[i + 2], confidenceFromUser);
270
271         printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
272           splitArray[i + 1] + " " + splitArray[i + 2], confidenceFromUser);
273       }
274     }
275   }
276 }
```

This screenshot shows the Eclipse IDE interface with the package explorer open, displaying the project structure and the contents of the tuli_garima_apriori.java file.

```
227   String eachString1 = receivedFrequentItemsList.get(j).trim();
228   String st1 = eachString1.substring(0, eachString1.lastIndexOf(" "));
229   if (st.equals(st1)) {
230     combinationItemsList
231       .add(eachString + " " + eachString1.substring(eachString1.lastIndexOf(" ") + 1));
232
233   }
234 }
235
236
237 // Function to form frequent List and frequent List for association
238 public static void frequentList(double receivedSupport) {
239   frequentItemList.clear();
240   receivedSupport = receivedSupport / 100;
241   Set<String> items = currentCandidacyList.keySet();
242
243   for (String eachItem : items) {
244     if ((double) currentCandidacyList.get(eachItem) / transactionAllItemsList.size() >= receivedSupport) {
245       frequentItemList.add(eachItem);
246       frequentItemsForAssociations.put(eachItem, currentCandidacyList.get(eachItem));
247     }
248   }
249
250
251 // Function to form all the associations
252 public static void findAllAssociations(HashMap<String, Integer> receivedFrequentItemsForAssociations,
253   double confidenceFromUser) {
254
255   for (String st : receivedFrequentItemsForAssociations.keySet()) {
256     String[] splitArray = st.split(" ");
257
258     if (splitArray.length == 2) {
259       printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[0],
260         splitArray[1], confidenceFromUser);
261
262       printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[0],
263         splitArray[0], confidenceFromUser);
264     } else if (splitArray.length == 3) {
265
266       for (int i = 0; i < splitArray.length; i++) {
267         if (i == 0) {
268           printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
269             splitArray[i + 1] + " " + splitArray[i + 2], confidenceFromUser);
270
271         printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
272           splitArray[i + 1] + " " + splitArray[i + 2], confidenceFromUser);
273       }
274     }
275   }
276 }
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_apriori.java
271     printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
272                                         splitArray[i + 1] + " " + splitArray[i + 2], splitArray[i], confidenceFromUser);
273 } else if (i == 1) {
274     printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
275                                         splitArray[1], splitArray[1 - 1] + " " + splitArray[1 + 1], confidenceFromUser);
276 } else {
277     printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
278                                         splitArray[i - 1] + " " + splitArray[i + 1], splitArray[i], confidenceFromUser);
279 } else {
280     printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
281                                         splitArray[i], splitArray[i - 2] + " " + splitArray[i - 1], confidenceFromUser);
282 } else {
283     printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
284                                         splitArray[i - 2] + " " + splitArray[i - 1], splitArray[i], confidenceFromUser);
285 } else {
286 }
287 }
288 }
289 }
290 }
291 // Function to print associations for confidence greater than user provided minimum confidence value
292 public static void printAssociationBasedOnConfidence(String eachFrequentListItem, int countOfEachFrequentListItem,
293                                                       String leftHandSide, String rightHandSide, double confidenceFromUser) {
294
295     int count = 0;
296     String[] splitArray = leftHandSide.split(" ");
297     int length = splitArray.length;
298
299     for (int i = 0; i < transactionAllItemsList.size(); i++) {
300         boolean fl = true;
301         for (int j = 0; j < length; j++) {
302             if (!transactionAllItemsList.get(i).contains(splitArray[j])) {
303                 fl = false;
304                 break;
305             }
306         }
307         if (fl == true) {
308             count = count + 1;
309         }
310     }
311
312     double confidenceOfEachAssociation = (double) countOfEachFrequentListItem / (count * 1.0);
313
314     double confidenceOfEachAssociation = (double) countOfEachFrequentListItem / (count * 1.0);
315

```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_apriori.java
316     double percentageUserConfidence = (double) confidenceFromUser / 100;
317
318     if (confidenceOfEachAssociation >= percentageUserConfidence) {
319         System.out.println(leftHandSide + " -> " + rightHandSide);
320     }
321 }
322 }
323 }
324

```

Console

```
tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 7:47:38 PM)

Welcome to Garima Tuli's Apriori Algorithm
*****
Following is the Input Dataset:
Coffee, Tam, Tea, Eggs, Honey, Bread
Juice, Eggs, Coffee, Bread, Butter
Cheese, Jam, Tea, Eggs, Honey, Milk, Bread
Cheese, Jam, Tea, Eggs, Honey, Juice, Bread
Cheese, Jam, Honey, Juice, Milk, Bread, Butter
Jam, Eggs, Honey, Juice, Milk, Bread, Butter
Cheese, Coffee, Tea, Eggs, Honey, Juice, Milk, Bread
Cheese, Coffee, Jam, Honey, Juice, Milk, Bread
Cheese, Coffee, Tea, Eggs, Juice, Milk, Butter
Honey, Juice, Coffee, Milk, Cheese, Butter
Cheese, Coffee, Eggs, Honey, Juice, Milk, Bread
Cheese, Jam, Eggs, Juice, Milk, Bread, Butter
Juice, Coffee, Jam, Honey, Juice, Milk, Bread
Cheese, Coffee, Tea, Honey, Juice, Milk, Bread
Cheese, Tea, Eggs, Juice, Milk, Bread, Butter
Coffee, Jam, Tea, Eggs, Honey, Juice, Bread
Coffee, Jam, Tea, Eggs, Honey, Juice, Butter
Coffee, Jam, Tea, Eggs, Juice, Bread, Butter
Jam, Tea, Eggs, Juice, Milk, Bread, Butter
Cheese, Coffee, Jam, Tea, Honey, Milk, Bread
*****
```

Enter Minimum Support Value:

Output (Screenshots with user specified values for minimum support and confidence on 5 different datasets)

Screenshot of Output for Dataset 1:

User enters minimum support value as 40 and minimum confidence value as 75 (as displayed in green color in the screenshot below).

This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure: `tuli_garima_midtermproj` containing `src` and `dataset`. The `src` folder contains `tuli_garima_apriori.java`. The `dataset` folder contains five files: `database1.txt`, `database2.txt`, `database3.txt`, `database4.txt`, and `database5.txt`.
- Console:** Displays the output of the Java application. It starts with "Welcome to Garima Tuli's Aprori Algorithm" and "*****". Then it prints the input dataset, which consists of 20 lines of itemsets separated by newlines. Examples include "Coffee, Jam, Tea, Eggs, Honey, Bread" and "Cheese, Jam, Tea, Eggs, Honey, Milk, Bread".
- Output:** The console shows user input at the bottom:
 - "Enter Minimum Support Value: **40**" (in green)
 - "Enter Minimum Confidence Value: **75**" (in green)

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:05:59 PM)

Enter Minimum Support Value:
40

Enter Minimum Confidence Value:
75

Assosiation Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:

```
Juice->Bread
Bread->Juice
Milk->Bread
Tea->Juice
Cheese->Bread
Jam Juice->Bread
Jam Bread->Juice
Cheese->Honey
Bread Eggs->Juice
Juice Eggs->Bread
Honey->Bread
Cheese->Bread Milk
Bread Milk->Cheese
Cheese Milk->Bread
Cheese Bread->Milk
Coffee->Bread
Tea->Eggs
Jam Eggs->Bread
Cheese Bread->Honey
Honey Cheese->Bread
Bread Milk->Juice
Juice Milk->Bread
Honey Cheese->Milk
Honey Milk->Cheese
Tea->Bread
Butter->Juice
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

tuli_garima_apriori.java

```
319     if (confidenceOfEachAssociation >= percentageUserConfidence) {
320         System.out.println(leftHandSide + "-" + rightHandSide);
321     }
322 }
```

Console

<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:05:59 PM)

```
Butter->Juice
Coffee->Juice
Cheese->Milk
Milk->Cheese
Tea Bread->Eggs
Tea Eggs->Bread
Tea Eggs->Juice
Tea Juice->Eggs
Eggs->Bread
Cheese Milk->Juice
Cheese->Juice Milk
Juice Milk->Cheese
Juice Cheese->Milk
Honey->Juice
Cheese->Juice
Butter Eggs->Juice
Honey Milk->Bread
Bread Coffee->Juice
Milk->Juice
Jam->Bread
Butter Bread->Juice
Eggs->Juice
Jam->Juice
Honey Jam->Bread
Honey Juice->Bread
Bread Cheese->Juice
Juice Cheese->Bread
```

Screenshot of Output for Dataset 2:

User enters minimum support value as 30 and minimum confidence value as 50 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99 // Replace different file name here for different data set
100 String fileName = "dataset\\database2.txt";
101 // Local variable for each line
102 String orchLine = null;
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Welcome to Garima Tuli's Apriori Algorithm

Following is the Input Dataset:

Orange,Apple,Onion,Grapes,Cherry,Pumpkin,Lemon
Orange,Apple,Tomato,Avocado,Cherry,Pumpkin
Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
Orange,Tomato,Avocado,Grapes,Cherry,Banana
Apple,Lemon,Avocado,Onion,Grapes,Banana
Orange,Apple,Tomato,Grapes,Cherry,Pumpkin
Orange,Avocado,Onion,Grapes,Cherry,Pumpkin,Banana
Orange,Apple,Tomato,Avocado,Onion,Cherry,Pumpkin,Lemon
Avocado,Lemon,Grapes,Cherry,Banana
Orange,Apple,Tomato,Onion,Grapes,Cherry,Lemon
Orange,Apple,Avocado,Onion,Pumpkin,Banana
Orange,Tomato,Avocado,Grapes,Cherry,Lemon,Banana
Onion,Grapes,Cherry,Pumpkin,Lemon
Orange,Apple,Tomato,Avocado,Onion,Grapes,Cherry,Pumpkin
Orange,Apple,Avocado,Lemon,Banana
Orange,Tomato,Onion,Grapes,Cherry,Pumpkin
Grapes,Cherry,Lemon
Orange,Apple,Tomato,Avocado,Onion,Grapes,Pumpkin
Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
Orange,Apple,Tomato,Avocado,Cherry,Grapes,Banana

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99 // Replace different file name here for different data set
100 String fileName = "dataset\\database2.txt";
101 // Local variable for each line
102 String orchLine = null;
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Enter Minimum Support Value:
30

Enter Minimum Confidence Value:
50

Association Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:

Apple->Onion Avocado
Onion Avocado->Apple
Onion->Apple Avocado
Apple Avocado->Onion
Avocado->Apple Onion
Apple Onion->Avocado
Apple->Cherry
Cherry->Apple
Onion->Orange
Orange->Onion
Cherry Onion->Apple
Apple Onion->Cherry
Onion->Apple Cherry
Apple Cherry->Onion
Grapes Avocado->Apple
Apple Avocado->Grapes
Apple Grapes->Avocado

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
<
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Apple Grapes->Avocado
Apple->Tomato
Tomato->Apple
Cherry->Avocado
Avocado->Cherry
Grapes->Avocado Orange
Avocado Orange->Grapes
Avocado->Grapes Orange
Grapes Orange->Avocado
Orange->Grapes Avocado
Grapes Avocado->Orange
Lemon Onion->Cherry
Lemon->Cherry Onion
Cherry Onion->Lemon
Onion->Cherry Lemon
Cherry Lemon->Onion
Avocado->Orange
Orange->Avocado
Pumpkin->Avocado
Avocado->Pumpkin
Grapes->Orange
Orange->Grapes
Avocado Onion->Grapes
Grapes Onion->Avocado
Onion->Grapes Avocado
Grapes Avocado->Onion
Apple->Avocado
Avocado->Apple
Grapes->Onion
Onion->Grapes
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
<
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Pumpkin->Orange Avocado
Orange Avocado->Pumpkin
Pumpkin Avocado->Orange
Avocado->Pumpkin Orange
Pumpkin Orange->Avocado
Cherry->Pumpkin
Pumpkin->Cherry
Banana->Orange
Cherry->Tomato
Tomato->Cherry
Cherry->Onion Pumpkin
Onion Pumpkin->Cherry
Onion->Cherry Pumpkin
Cherry Pumpkin->Onion
Pumpkin->Cherry Onion
Cherry Onion->Pumpkin
Orange Lemon->Apple
Apple Lemon->Orange
Lemon->Apple Orange
Apple Orange->Lemon
Onion->Orange Pumpkin
Orange Pumpkin->Onion
Orange->Onion Pumpkin
Onion Pumpkin->Orange
Pumpkin->Onion Orange
Onion Orange->Pumpkin
Onion->Avocado Pumpkin
Avocado Pumpkin->Onion
Avocado->Onion Pumpkin
Onion Pumpkin->Avocado
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt
database2.txt
database3.txt
database4.txt
database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)

Onion Pumpkin->Avocado
Pumpkin->Onion Avocado
Onion Avocado->Pumpkin
Cherry->Avocado Orange
Avocado Orange->Cherry
Avocado->Cherry Orange
Cherry Orange->Avocado
Orange->Cherry Avocado
Cherry Avocado->Orange
Grapes->Cherry Pumpkin
Cherry Pumpkin->Grapes
Cherry->Grapes Pumpkin
Grapes Pumpkin->Cherry
Pumpkin->Grapes Cherry
Grapes Cherry->Pumpkin
Tomato Avocado->Cherry
Tomato->Cherry Avocado
Cherry Avocado->Tomato
Cherry Tomato->Avocado
Cherry->Orange Onion
Orange Onion->Cherry
Orange->Cherry Onion
Cherry Onion->Orange
Onion->Cherry Orange
Cherry Orange->Onion
Cherry->Orange Pumpkin
Orange Pumpkin->Cherry
Orange->Cherry Pumpkin
Cherry Pumpkin->Orange
Pumpkin->Cherry Orange
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt
database2.txt
database3.txt
database4.txt
database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)

Pumpkin->Cherry Orange
Cherry Orange->Pumpkin
Onion->Avocado Orange
Avocado Orange->Onion
Avocado->Onion Orange
Onion Orange->Avocado
Onion Avocado->Orange
Apple->Orange Onion
Orange Onion->Apple
Orange->Apple Onion
Apple Onion->Orange
Onion->Apple Orange
Apple Orange->Onion
Apple->Lemon
Lemon->Apple
Pumpkin->Orange
Orange->Pumpkin
Apple->Grapes Cherry
Grapes Cherry->Apple
Apple Cherry->Grapes
Apple Grapes->Cherry
Cherry->Lemon
Lemon->Cherry
Avocado->Orange Tomato
Orange Tomato->Avocado
Avocado Tomato->Orange
Tomato->Avocado Orange
Avocado Orange->Tomato
Grapes->Tomato
Tomato->Grapes
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Tomato->Grapes
Avocado->Banana
Banana->Avocado
Orange Lemon->Cherry
Cherry Lemon->Orange
Lemon->Cherry Orange
Tomato->Orange
Orange->Tomato
Grapes->Pumpkin Onion
Pumpkin Onion->Grapes
Pumpkin->Grapes Onion
Grapes Onion->Pumpkin
Onion->Grapes Pumpkin
Grapes Pumpkin->Onion
Apple->Grapes
Grapes->Apple
Onion Lemon->Grapes
Onion->Grapes Lemon
Grapes Lemon->Onion
Lemon->Grapes Onion
Grapes Onion->Lemon
Grapes Pumpkin->Apple
Apple Pumpkin->Grapes
Pumpkin->Apple Grapes
Apple Grapes->Pumpkin
Onion Lemon->Apple
Onion->Apple Lemon
Apple Lemon->Onion
Lemon->Apple Onion
Apple Onion->Lemon

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Apple Onion->Lemon
Grapes->Tomato Orange
Tomato Orange->Grapes
Tomato->Grapes Orange
Grapes Orange->Tomato
Orange->Grapes Tomato
Grapes Tomato->Orange
Apple->Onion Pumpkin
Onion Pumpkin->Apple
Onion->Apple Pumpkin
Apple Pumpkin->Onion
Pumpkin->Apple Onion
Apple Onion->Pumpkin
Grapes->Cherry
Cherry->Grapes
Grapes->Cherry Orange
Cherry Orange->Grapes
Cherry->Grapes Orange
Grapes Orange->Cherry
Orange->Grapes Cherry
Grapes Cherry->Orange
Apple->Pumpkin
Pumpkin->Apple
Cherry->Orange
Orange->Cherry
Pumpkin Avocado->Apple
Pumpkin->Apple Avocado
Apple Avocado->Pumpkin
Apple Pumpkin->Avocado
Avocado Banana->Grapes

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt
database2.txt
database3.txt
database4.txt
database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Grapes Banana->Avocado
Banana->Grapes Avocado
Grapes Avocado->Banana
Apple->Orange Avocado
Orange Avocado->Apple
Orange->Apple Avocado
Apple Avocado->Orange
Avocado->Apple Orange
Apple Orange->Avocado
Grapes->Lemon
Lemon->Grapes
Onion->Pumpkin
Pumpkin->Onion
Grapes->Orange Onion
Orange Onion->Grapes
Orange->Grapes Onion
Grapes Onion->Orange
Onion->Grapes Orange
Grapes Orange->Onion
Apple->Tomato Orange
Tomato Orange->Apple
Tomato->Apple Orange
Apple Orange->Tomato
Apple Tomato->Orange
Apple->Grapes Onion
Grapes Onion->Apple
Apple Onion->Grapes
Onion->Apple Grapes
Apple Grapes->Onion
Onion->Cherry Orange

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt
database2.txt
database3.txt
database4.txt
database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
```

Grapes->Cherry Lemon
Cherry Lemon->Grapes
Cherry->Grapes Lemon
Grapes Lemon->Cherry
Lemon->Grapes Cherry
Grapes Cherry->Lemon
Onion->Avocado
Avocado->Onion
Apple->Grapes Orange
Grapes Orange->Apple
Grapes->Apple Orange
Apple Orange->Grapes
Orange->Apple Grapes
Apple Grapes->Orange
Cherry Pumpkin->Apple
Apple Pumpkin->Cherry
Pumpkin->Apple Cherry
Apple Cherry->Pumpkin
Cherry->Tomato Orange
Tomato Orange->Cherry
Tomato->Cherry Orange
Cherry Orange->Tomato
Orange->Cherry Tomato
Cherry Tomato->Orange
Grapes->Avocado
Avocado->Grapes
Avocado->Lemon
Lemon->Avocado
Grapes->Pumpkin Orange
Pumpkin Orange->Grapes

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99
100 // Replace different file name here for different data set
101     String fileName = "dataset\\database2.txt";
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Pumpkin->Grapes Orange
Grapes Orange->Pumpkin
Orange->Grapes Pumpkin
Grapes Pumpkin->Orange
Onion->Lemon
Lemon->Onion
Tomato Cherry->Grapes
Tomato->Grapes Cherry
Grapes Cherry->Tomato
Grapes Tomato->Cherry
Lemon->Orange
Grapes->Pumpkin
Pumpkin->Grapes
Apple->Orange
Orange->Apple
Apple->Onion
Onion->Apple
Grapes->Cherry Onion
Cherry Onion->Grapes
Cherry->Grapes Onion
Grapes Onion->Cherry
Onion->Grapes Cherry
Grapes Cherry->Onion
Apple->Orange Pumpkin
Orange Pumpkin->Apple
Orange->Apple Pumpkin
Apple Pumpkin->Orange
Pumpkin->Apple Orange
Apple Orange->Pumpkin
Banana Orange->Avocado
```

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
98    public static void readDataFromFile() {
99
100    // Replace different file name here for different data set
101    String fileName = "dataset\\database2.txt";
102    // Local variable for each line
103    String eachLine = null;
104    // Unique Items
105    HashSet<String> uniqueItems = new HashSet<String>();
106
107    try {
108        FileReader fileReader = new FileReader(fileName);
109        BufferedReader bufferedReader = new BufferedReader(fileReader);
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Banana Orange->AVOCADO
Banana->Avocado Orange
Avocado Orange->Banana
Avocado Banana->Orange
Banana->Grapes
Grapes->Avocado Cherry
Avocado Cherry->Grapes
Avocado->Grapes Cherry
Grapes Cherry->Avocado
Cherry->Grapes Avocado
Grapes Avocado->Cherry
Cherry Avocado->Apple
Apple Avocado->Cherry
Apple Cherry->Avocado
Avocado->Tomato
Tomato->Avocado
Cherry->Onion
Onion->Cherry
Cherry Tomato->Apple
Apple Tomato->Cherry
Tomato->Apple Cherry
Apple Cherry->Tomato
```

Screenshot of Output for Dataset 3:

User enters minimum support value as 60 and minimum confidence value as 80 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

The screenshot shows the Eclipse IDE interface with the project 'tuli_garima_midtermproj' open. The 'src' folder contains the file 'tuli_garima_apriori.java'. The code reads data from 'dataset\\database3.txt'. The 'Console' tab displays the input dataset, which consists of a list of candy bar names separated by commas. The list includes: Godiva, Cadbury, Bournville, MilkyBar, Lindt, Twix, Toblerone, Lindt, Twix, Bournville, Hershey, KitKat, Toblerone, Godiva, Cadbury, Bournville, MilkyBar, Hershey, KitKat, Twix, Godiva, Cadbury, Bournville, MilkyBar, Nestle, Twix, Toblerone, Godiva, Twix, KitKat, Bournville, Hershey, KitKat, Cadbury, Godiva, Twix, Nestle, MilkyBar, Hershey, KitKat, Lindt, Nestle, Cadbury, Godiva, Cadbury, Bournville, Hershey, Lindt, KitKat, Godiva, Cadbury, Nestle, Bournville, MilkyBar, KitKat, Twix, Lindt, Bournville, MilkyBar, Hershey, Godiva, Toblerone, Lindt, Cadbury, MilkyBar, Hershey, Godiva, Toblerone, Godiva, Cadbury, Nestle, Bournville, Hershey, Lindt, KitKat, Twix, Cadbury, KitKat, Hershey, Nestle, Twix, Lindt, Bournville, MilkyBar, Godiva, Lindt, Cadbury, Nestle, Bournville, Hershey, KitKat, Twix, Godiva, MilkyBar, Twix, MilkyBar, Hershey, KitKat, Lindt, Godiva, Cadbury, Bournville, Cadbury, KitKat, Hershey, Godiva, Twix, Nestle, Bournville, Toblerone, Godiva, KitKat, Bournville, MilkyBar, Hershey, Nestle, Lindt, Twix, Godiva, Cadbury, KitKat, Bournville, Hershey, Nestle, Twix, Lindt, Cadbury, Nestle, MilkyBar, Hershey, KitKat, Twix, Toblerone, Lindt, Twix, MilkyBar, Hershey, KitKat, Toblerone, Godiva, Lindt, MilkyBar, Hershey, KitKat, Toblerone.

The screenshot shows the Eclipse IDE interface with the project 'tuli_garima_midtermproj' open. The 'src' folder contains the file 'tuli_garima_apriori.java'. The code reads data from 'dataset\\database3.txt'. The 'Console' tab shows the user entering '60' as the minimum support value and '80' as the minimum confidence value. The output then lists the association rules generated by the algorithm, such as Cadbury->Bournville Godiva, Bournville Godiva->Cadbury, Bournville->Cadbury Godiva, Cadbury Godiva->Bournville, Cadbury Bournville->Godiva, Godiva->Bournville, Bournville->Godiva, Cadbury->Bournville, Bournville->Cadbury, MilkyBar->Godiva, Bournville->Hershey, KitKat->Hershey Twix, Hershey Twix->KitKat, KitKat Twix->Hershey, Twix->KitKat Hershey, KitKat Hershey->Twix, Hershey Godiva->KitKat, KitKat Godiva->Hershey.

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

tuli_garima_midtermproj

JRE System Library [JavaSE-11]

src

tuli_garima_midtermproj

tuli_garima_apriori.java

dataset

database1.txt

database2.txt

database3.txt

database4.txt

database5.txt

tuli_garima_apriori.java

```
99 // Replace different file name here for different data set
100 String fileName = "dataset\\database3.txt";
102 // Local variable for each line
```

Console

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:42:57 PM)
KitKat Hershey->Godiva
Cadbury->Hershey
Cadbury->Godiva
Godiva->Cadbury
Bournville->KitKat
Cadbury Godiva->Twix
Cadbury->Twix Godiva
Twix Godiva->Cadbury
Twix Cadbury->Godiva
Twix->Hershey
KitKat->Hershey
Hershey->KitKat
Cadbury->KitKat
Twix->Cadbury
Cadbury->Twix
Lindt->Hershey
Godiva Bournville->Twix
Twix Bournville->Godiva
Bournville->Twix Godiva
Twix Godiva->Bournville
Twix->Godiva
KitKat->Twix
Twix->KitKat
Twix->Bournville
Bournville->Twix
Hershey->Godiva
Godiva->Hershey
KitKat->Godiva
```

Screenshot of Output for Dataset 4:

User enters minimum support value as 35 and minimum confidence value as 85 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

The screenshot shows two instances of the Eclipse IDE interface, each displaying a Java application's console output. Both instances have a similar layout with a 'Package Explorer' view on the left and a 'Console' view on the right.

Top Console Output (Dataset 4 Input):

```
Welcome to Garima Tuli's Apriori Algorithm
*****
Following is the Input Dataset:
Printer,USB,Laptop,Mouse,HPInk,EyeGlass
Printer,Laptop,Mouse,HPInk,EyeGlass
Printer,USB,Monitor,Keyboard,Mouse,HPInk,Laptop
Printer,Monitor,Keyboard,Mouse,HPInk,Laptop
Bluetooth,USB,Laptop,MemoryCard,EyeGlass
Printer,Monitor,Keyboard,Mouse,HPInk,Laptop
Printer,USB,Laptop,HPInk,EyeGlass
Printer,Laptop,HPInk,EyeGlass
Printer,USB,HPInk,Laptop
Bluetooth,Laptop,Mouse,MemoryCard,EyeGlass
Bluetooth,Monitor,Keyboard,Mouse,Laptop
Bluetooth,USB,Laptop,MemoryCard,EyeGlass
Printer,Monitor,Keyboard,Mouse,HPInk,EyeGlass
Printer,Laptop,HPInk,EyeGlass
Printer,USB,HPInk,Laptop
Printer,Monitor,Keyboard,Mouse,HPInk
Printer,USB,Monitor,Keyboard,Mouse,HPInk
Printer,Laptop,Mouse,HPInk,EyeGlass
Bluetooth,USB,MemoryCard,Printer,HPInk,Laptop
Bluetooth,USB,MemoryCard,Printer,Laptop,HPInk,EyeGlass
*****
Enter Minimum Support Value:
35
Enter Minimum Confidence Value:
85
```

Bottom Console Output (Dataset 4 Association Rules):

```
Enter Minimum Support Value:
35
Enter Minimum Confidence Value:
85
*****
Association Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:
Monitor->Mouse
Laptop Printer->HPInk
Laptop HPInk->Printer
HPInk EyeGlass->Laptop
HPInk->Printer
Printer->HPInk
Monitor->Mouse Keyboard
Mouse Keyboard->Monitor
Monitor Keyboard->Mouse
Keyboard->Monitor Mouse
Monitor Mouse->Keyboard
Printer Mouse->HPInk
HPInk Mouse->Printer
Printer EyeGlass->Laptop
Keyboard->Mouse
Bluetooth->Laptop
Monitor->Keyboard
Keyboard->Monitor
USB->Laptop
EyeGlass Printer->HPInk
HPInk EyeGlass->Printer
Printer USB->HPInk
HPInk USB->Printer
EyeGlass->Laptop
```

Screenshot of Output for Dataset 5:

User enters minimum support value as 30 and minimum confidence value as 70 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

The screenshot shows two instances of the Eclipse IDE. The top instance displays the output of the Java application, and the bottom instance shows the source code of the Java file.

Output Window (Top):

```
Welcome to Garima Tuli's Apriori Algorithm
*****
Following is the Input Dataset:
ToothPaste,ToothBrush,LaundryDetergent,Shampoo,HairOil
Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
ToothPaste,ToothBrush,LaundryDetergent
ToothPaste,ToothBrush,LaundryDetergent
Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
ToothPaste,ToothBrush,LaundryDetergent,HairOil
Handwash,Moisturizer,LaundryDetergent,HairOil
ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
Moisturizer,NailPolish,PaperTowels,HairOil
ToothPaste,Moisturizer,NailPolish,LaundryDetergent,Shampoo,Conditioner,HairOil
ToothPaste,Hardwash,Moisturizer,NailPolish,PaperTowels,HairOil
*****
Enter Minimum Support Value:
30
Enter Minimum Confidence Value:
70
```

Source Code Window (Bottom):

```
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database5.txt";
102 // Local variable for each line
103 String eachLine = null;
```

Console Output (Bottom):

```
Enter Minimum Support Value:
30
Enter Minimum Confidence Value:
70
*****
Assosciation Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:
Moisturizer NailPolish->PaperTowels
PaperTowels NailPolish->Moisturizer
NailPolish->PaperTowels Moisturizer
PaperTowels Moisturizer->NailPolish
NailPolish->Moisturizer
Shampoo Conditioner->HairOil
HairOil Conditioner->Shampoo
Conditioner->HairOil Shampoo
ToothBrush->HairOil
Shampoo->HairOil
Shampoo Moisturizer->HairOil
ToothBrush ToothPaste->HairOil
ToothBrush->HairOil ToothPaste
HairOil ToothPaste->ToothBrush
HairOil ToothBrush->ToothPaste
Conditioner->Shampoo
Handwash->HairOil
*****
```

```
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 9:20:05 PM)
Shampoo -> Handwash
Shampoo ToothPaste -> HairOil
HairOil ToothPaste -> Shampoo
Handwash -> HairOil
Moisturizer -> Handwash
Handwash HairOil -> Moisturizer
Handwash -> Conditioner
Conditioner -> Moisturizer
Conditioner -> Shampoo
Shampoo -> Moisturizer
Conditioner -> Conditioner
Conditioner -> Shampoo
Conditioner -> HairOil
LaundryDetergent -> HairOil
NailPolish -> HairOil
ToothBrush -> ToothPaste
Shampoo PaperTowels -> HairOil
HairOil PaperTowels -> Shampoo
PaperTowels -> HairOil
Shampoo -> Moisturizer
Moisturizer -> HairOil
Moisturizer Conditioner -> HairOil
HairOil Conditioner -> Moisturizer
Conditioner -> HairOil
Moisturizer -> Handwash
NailPolish -> Handwash
NailPolish -> Moisturizer
Handwash NailPolish -> Moisturizer
Conditioner -> HairOil
NailPolish Moisturizer -> HairOil
NailPolish -> HairOil
Moisturizer -> HairOil
NailOil NailPolish -> Moisturizer
PaperTowels -> Shampoo
NailPolish -> PaperTowels
Handwash -> Moisturizer
Moisturizer -> Handwash
```

Conclusion

Thus, it's observed that Apriori Algorithm generates various Association rules for 5 different datasets when user inputs different minimum support & confidence values for each dataset.