****

Name: Krithivas Swaminathan

Email: KS583@njit.edu

Subject: CS634 Data Mining

**Source Code**

**aprioriMining.java**

**package** aprioriMining;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.HashSet;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.Set;

**public** **class** aprioriMining {

**static** HashMap<String, Integer> *supportCalc* = **new** HashMap<String,Integer>();

**static** List<List<String>> *lists* = **new** ArrayList<List<String>>();

**static** HashMap<String, Integer> *map* =**new** HashMap<String,Integer>();

**static** HashMap<String, Integer> *twoMap* = **new** HashMap<String,Integer>();

**public** **static** **void** main ( String args[]) **throws** IOException

{

**try**

{

File f = **null**;

BufferedReader systemInput = **new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("Association Rule Mining");

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println("Choose DataBase");

System.***out***.println("1.Shoprite");

System.***out***.println("2.Walmart");

System.***out***.println("3.Amazon");

System.***out***.println("4.Nike");

System.***out***.println("5.Kmart");

System.***out***.println("Enter your option : ");

**int** option = Integer.*parseInt*(systemInput.readLine());

**switch** (option) {

**case** 1:

f = **new** File("shoprite.txt");

**break**;

**case** 2:

f = **new** File("walmart.txt");

**break**;

**case** 3:

f = **new** File("amazon.txt");

**break**;

**case** 4:

f = **new** File("nike.txt");

**break**;

**case** 5:

f = **new** File("kmart.txt");

**break**;

**default**:

**break**;

}

**int** transactions = *readFile*(f);

System.***out***.println("Enter Minimum Support Value in percentage %: ");

**int** minSup= Integer.*parseInt*(systemInput.readLine());

System.***out***.println("Enter Minimum Confidence Value in percentage %: ");

**int** minConf = Integer.*parseInt*(systemInput.readLine());

**int** supVal = *removeLowSupportVal*(minSup,transactions);

*secondIteration*();

*thirdIteration*(supVal);

}

**catch** (NumberFormatException e)

{

System.***out***.println("Please enter a valid number");

}

**catch** (FileNotFoundException e) {

System.***out***.println("File not found in the path");

}

**catch** (NullPointerException e) {

// **TODO**: handle exception

System.***out***.println("Please enter a number between 1 to 5");

}

}

**private** **static** **void** thirdIteration(**int** supVal) {

HashMap<String, Integer> tempMap = **new** HashMap<String,Integer>();

ArrayList<String> tempList = **new** ArrayList<String>();

tempMap = *map*;

**for**(Entry<String, Integer> entry :tempMap.entrySet() )

{

**int** checker=0;

checker=entry.getValue();

**if**(checker<supVal)

tempList.add(entry.getKey());

}

**for**(String loop:tempList)

{

*map*.remove(loop);

}

System.***out***.println("Second Database Scan (2nd Iteration)");

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println();

System.***out***.println("Generating Frequent Itemsets...");

ArrayList<String> tempSet = **new** ArrayList<>();

**for**(String keys : *map*.keySet())

{

tempSet.add(keys);

}

System.***out***.println();

System.***out***.println("Freq.Set Support Value");

System.***out***.println("~~~~~~~~~~ ~~~~~~~~~~~~~");

**for** (**int** b=0;b<tempSet.size();b++)

{

System.***out***.println(tempSet.get(b)+" ------------> "+*map*.get(tempSet.get(b)));

}

System.***out***.println();

**for**(**int** i =0; i < tempSet.size(); i ++)

{

**for**(**int** j =i+1; j < tempSet.size();j++)

{

String name = tempSet.get(i)+","+tempSet.get(j);

name=name.trim();

ArrayList<String> t = **new** ArrayList<>();

String[] enum1 = tempSet.get(i).split(",");

String[] enum2= tempSet.get(j).split(",");

**for** (String f : enum1)

{

t.add(f);

}

**for**(String g : enum2)

{

t.add(g);

}

*buildThirdMap*(t,name);

}

}

HashMap<String, Integer> randMap = **new** HashMap<String,Integer>();

ArrayList<String> randList = **new** ArrayList<String>();

randMap = *twoMap*;

**for**(Entry<String, Integer> entry :randMap.entrySet() )

{

**int** checker=0;

checker=entry.getValue();

**if**(checker<supVal)

randList.add(entry.getKey());

}

**for**(String loop:randList)

{

*twoMap*.remove(loop);

}

*removeDuplicates*(*twoMap*,randList);

}

**private** **static** **void** removeDuplicates(HashMap<String, Integer> twoMap2,ArrayList<String> randList2)

{

ArrayList<String> dup = **new** ArrayList<>();

ArrayList<String> finalList=**new** ArrayList<>();

**for**(String keys : twoMap2.keySet())

{

dup.add(keys);

}

**for** (**int** i=0;i<dup.size();i++)

{

**boolean** duplicate = **false**;

String[] splitter = dup.get(i).split(",");

**for** (**int** j = 0 ; j < splitter.length ; j++)

{

**for**(**int** k = j+1 ; k < splitter.length ; k ++)

{

**if** ((splitter[j].trim()).equals(splitter[k].trim()))

{

duplicate = **true** ;

}

}

}

**if** (duplicate==**false**)

{

finalList.add(dup.get(i));

}

}

*printoutFinal*(finalList);

System.***out***.println();

System.***out***.println("Exiting.. BYE");

}

**private** **static** **void** printoutFinal(ArrayList<String> finalList)

{

**if** (finalList.isEmpty()) {

System.***out***.println("Third DataBase Scan (3rd Iteration)");

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println();

System.***out***.println("Generating Frequent Itemsets...");

System.***out***.println();

System.***out***.println("Freq. Set Support Value");

System.***out***.println("~~~~~~~~~~ ~~~~~~~~~~~~~~");

ArrayList<String> g = **new** ArrayList<>();

Set<String> tempSet = **new** HashSet<>();

**for** (String key : *twoMap*.keySet())

{

g.add(key);

}

**for**(**int** h = 0 ; h < g.size(); h ++)

{

Set<String> st = **new** HashSet<>();

String[] sp = g.get(h).split(",");

**for** (**int** e =0; e < sp.length ; e ++)

{

st.add(sp[e].trim());

}

**if** (!(tempSet.containsAll(st)))

{

System.***out***.println(st+" ----------> "+*twoMap*.get(g.get(h)));

}

tempSet.addAll(st);

}

System.***out***.println();

System.***out***.println("No more association rules can be generated");

}

**else**

{

System.***out***.println("Final DataBase Scan");

System.***out***.println("~~~~~~~~~~~~~~~~~~~");

System.***out***.println();

System.***out***.println("Generating Frequent Itemsets...");

System.***out***.println();

System.***out***.println("Freq. Set Support Value");

System.***out***.println("~~~~~~~~~~ ~~~~~~~~~~~~~~");

**for** (**int** i = 0 ; i < finalList.size(); i++)

{

System.***out***.println(finalList.get(i)+" ------> "+*twoMap*.get(finalList.get(i)));

}

System.***out***.println();

System.***out***.println("No more Association rules can be Generated.");

}

}

**private** **static** **void** buildThirdMap(ArrayList<String> t, String name)

{

**for** (**int** x = 0 ; x < *lists*.size();x++ )

{

List<String> temp = **new** ArrayList<String>();

temp = *lists*.get(x);

String[] enum3 = temp.get(0).split(",");

ArrayList<String> rand = **new** ArrayList<String>();

**for**(String l : enum3)

{

rand.add(l);

}

**if**(rand.containsAll(t))

{

**if** (!*twoMap*.containsKey(name))

{

*twoMap*.put(name,1);

}

**else**

{

*twoMap*.put(name,*twoMap*.get(name)+1);

}

}

}

}

**private** **static** **void** secondIteration() {

ArrayList<String> freqSet = **new** ArrayList<String>();

**for**(String findSz: *supportCalc*.keySet())

{

freqSet.add(findSz);

}

**for**(**int** i = 0; i < freqSet.size() ; i ++)

{

**for** (**int** j= i+1 ; j < freqSet.size() ; j++)

{

String str = freqSet.get(i)+","+freqSet.get(j);

*buildMap*(freqSet,i,j,str);

}

}

}

**private** **static** **void** buildMap(ArrayList<String> set, **int** a , **int** b, String str)

{

List<String> tempList =**new** ArrayList<String>();

tempList.add(set.get(a).trim());

tempList.add(set.get(b).trim());

str=str.trim();

**for**(**int** i =0 ; i< *lists*.size() ; i++)

{

List<String> build = **new** ArrayList<String>();

build = *lists*.get(i);

**if**((build.get(0).contains(set.get(a)) && build.get(0).contains(set.get(b))))

{

**if**(!*map*.containsKey(str))

{

*map*.put(str,1);

}

**else**

{

*map*.put(str,*map*.get(str)+1);

}

}

}

}

**private** **static** **int** readFile(File file) **throws** IOException

{

**int** transCounter =0;

BufferedReader br = **new** BufferedReader(**new** FileReader(file));

String letsRead = **null**;

**while**((letsRead = br.readLine())!=**null**)

{

String[] transactionsArray = letsRead.split(":",-1);

String[] itemsArray = transactionsArray[1].split(",",-1);

ArrayList<String> tempList = **new** ArrayList<String>();

tempList.add(transactionsArray[1]);

*lists*.add(tempList);

**for**(String temp: itemsArray)

{

**if**(!*supportCalc*.containsKey(temp))

*supportCalc*.put(temp, 1);

**else**

*supportCalc*.put(temp,*supportCalc*.get(temp)+1);

}

transCounter ++;

}

br.close();

**return** transCounter;

}

**private** **static** **int** removeLowSupportVal(**int** minSup, **int** transCounter) **throws** NumberFormatException, IOException

{

Map<String, Integer> map = **new** HashMap<String, Integer>();

ArrayList<String> ar = **new** ArrayList<>();

**double** offset =0;

offset = (minSup \*.01)\*transCounter;

**int** roundOff= (**int**) (offset+0.5);

map=*supportCalc*;

**for**(Entry<String, Integer> entry :map.entrySet() )

{

**int** checker=0;

checker=entry.getValue();

**if**(checker<roundOff)

ar.add(entry.getKey());

}

System.***out***.println("Generating All Association Rules...");

System.***out***.println();

System.***out***.println("First Database Scan (1st Iteration)");

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println();

System.***out***.println("Generating Frequent Itemsets...");

**for**(String loop:ar)

{

*supportCalc*.remove(loop);

}

ArrayList<String> iterate = **new** ArrayList<>();

**for**(String sr : *supportCalc*.keySet())

{

iterate.add(sr);

}

System.***out***.println();

System.***out***.println("Freq. Set Support Value");

System.***out***.println("~~~~~~~~~~ ~~~~~~~~~~~~~");

**for** (**int** b=0;b<iterate.size();b++)

{

System.***out***.println(iterate.get(b)+" -----------> "+*supportCalc*.get(iterate.get(b)));

}

System.***out***.println("Association rules eliminated in this iteration are : "+ar );

System.***out***.println();

**return** roundOff;

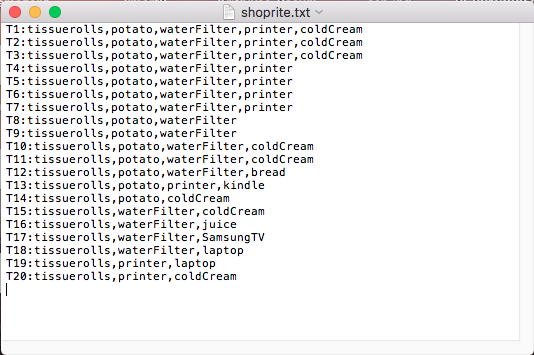
}

}

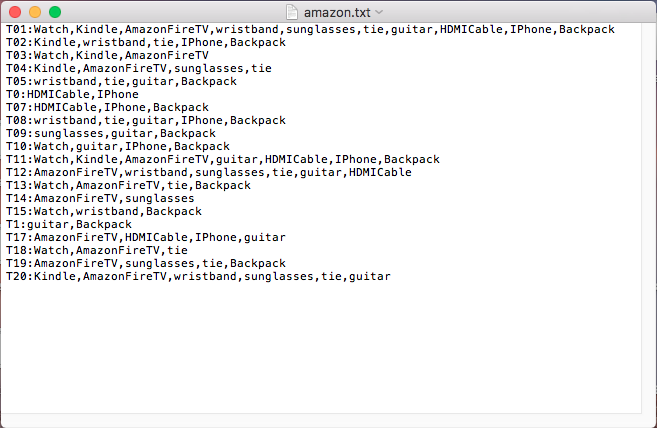
SCREENSHOTS

DATABASES

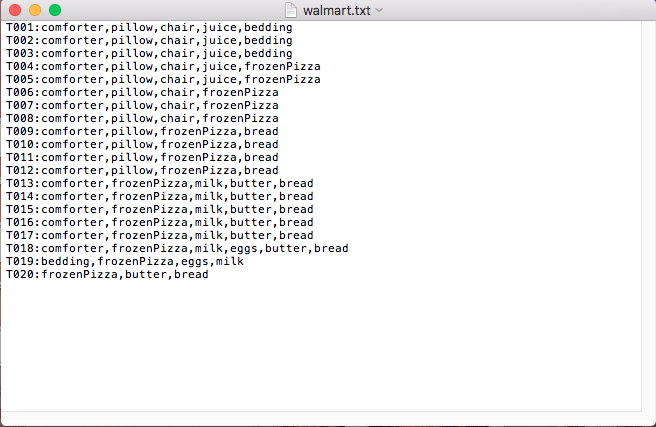
1. **shoprite.txt**



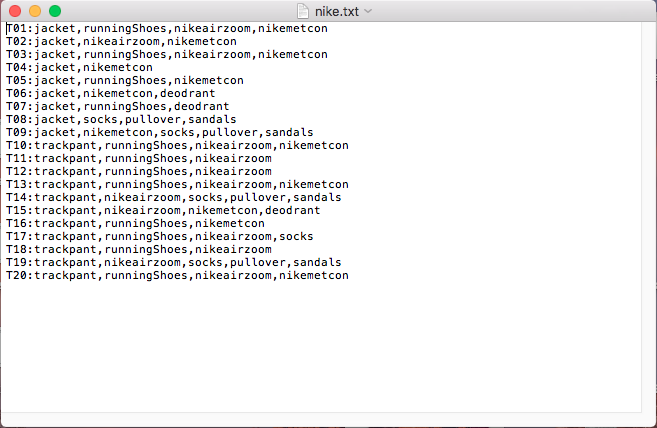
1. **amazon.txt**



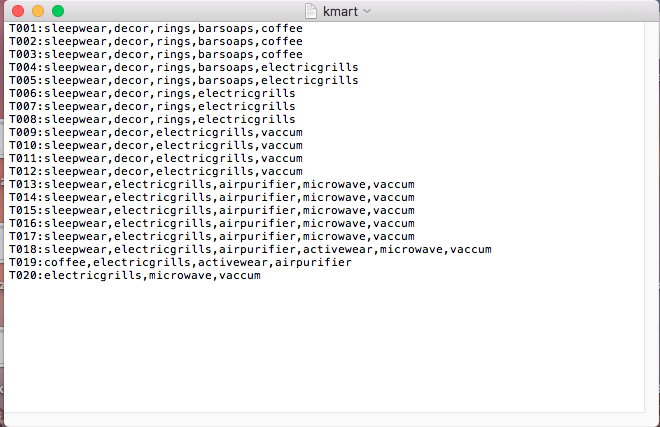
1. **walmart.txt**

****

1. **nike.txt**

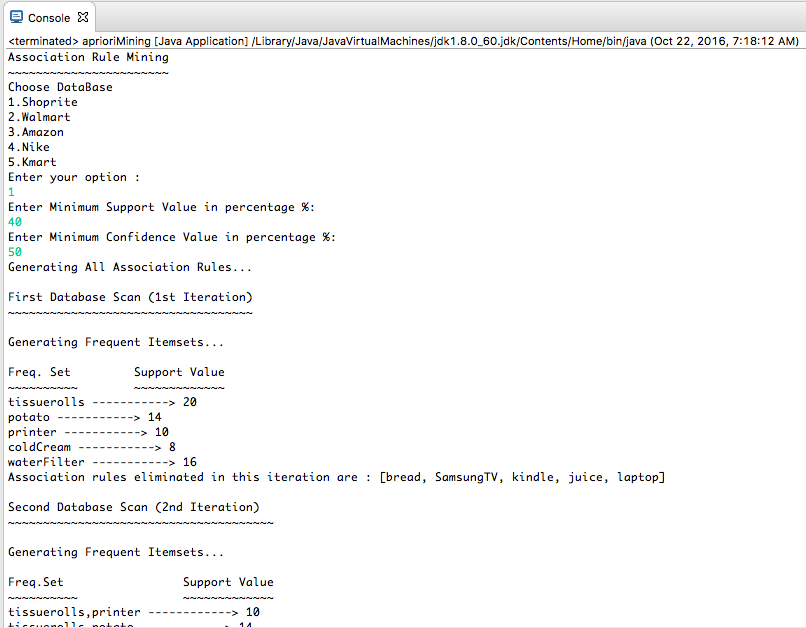


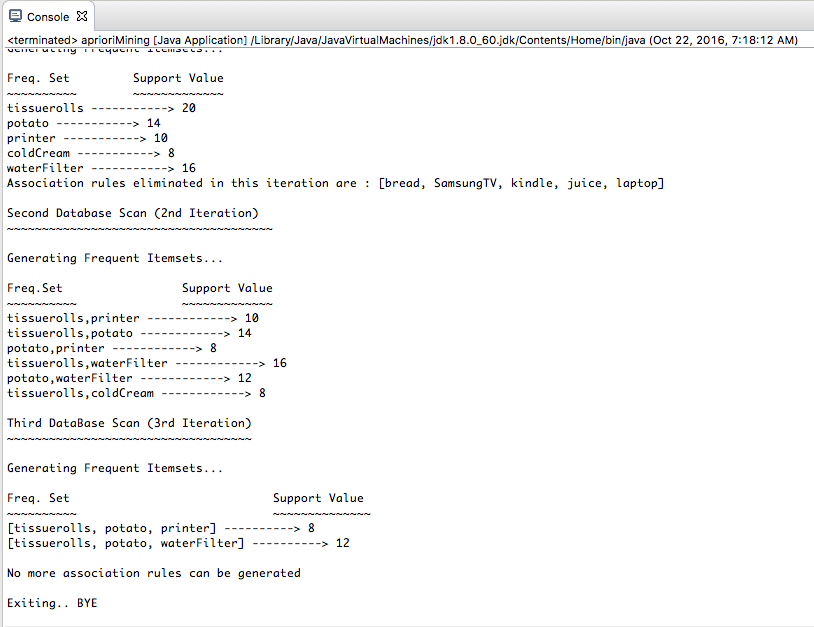
1. **kmart.txt**

****

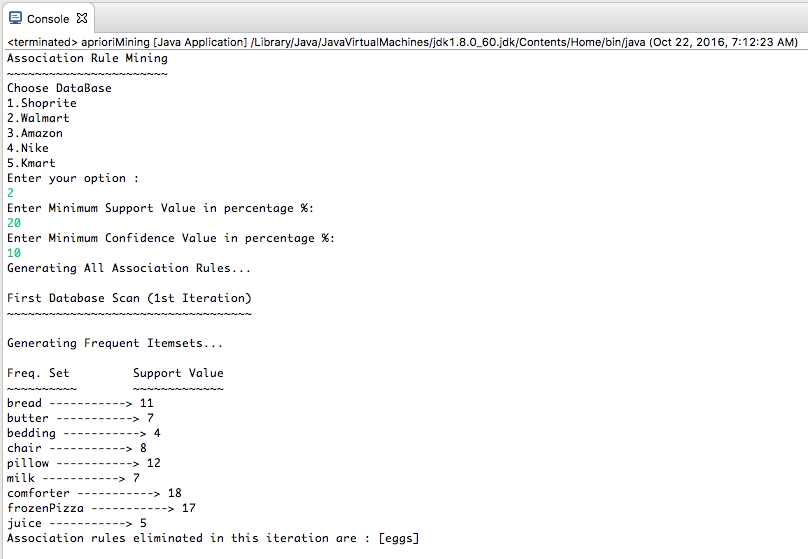
**Execution Outputs**

**Execution 1: Shoprite database with support= 40% and confidence =50%**

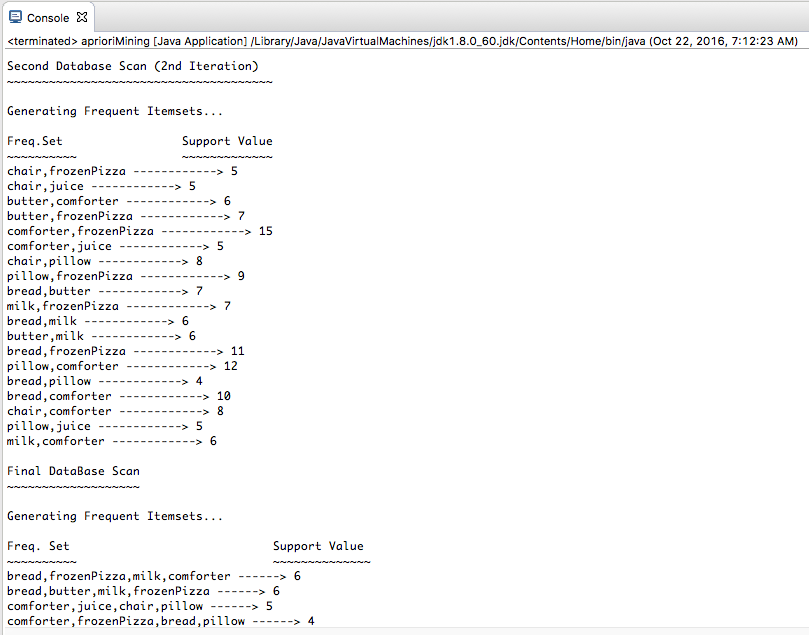


Output continued .. 

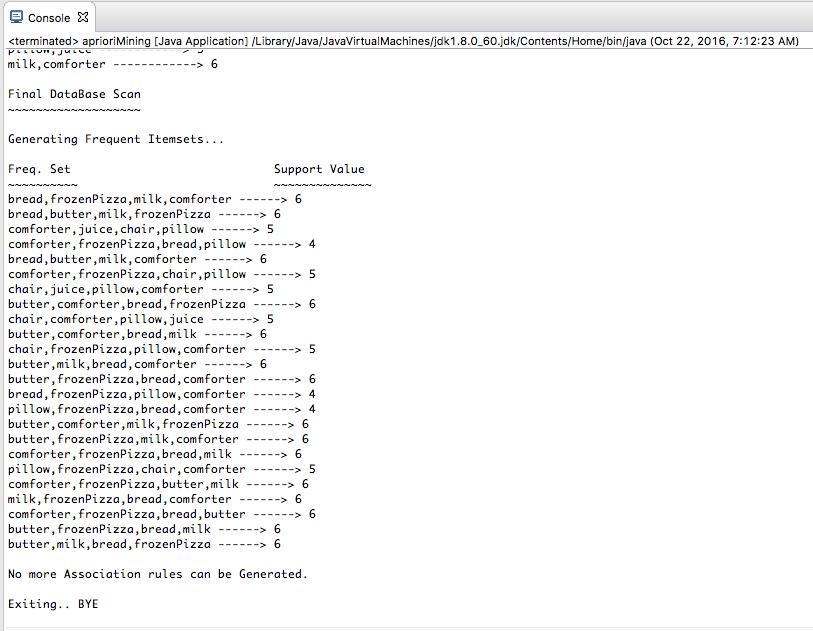
**Execution 2: Walmart database with support= 20% and confidence =10%**



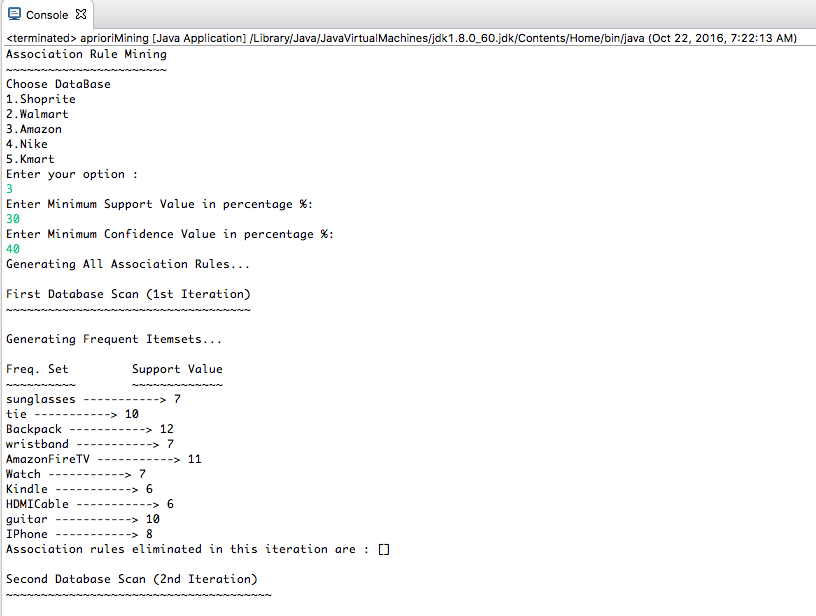
Output continued ..



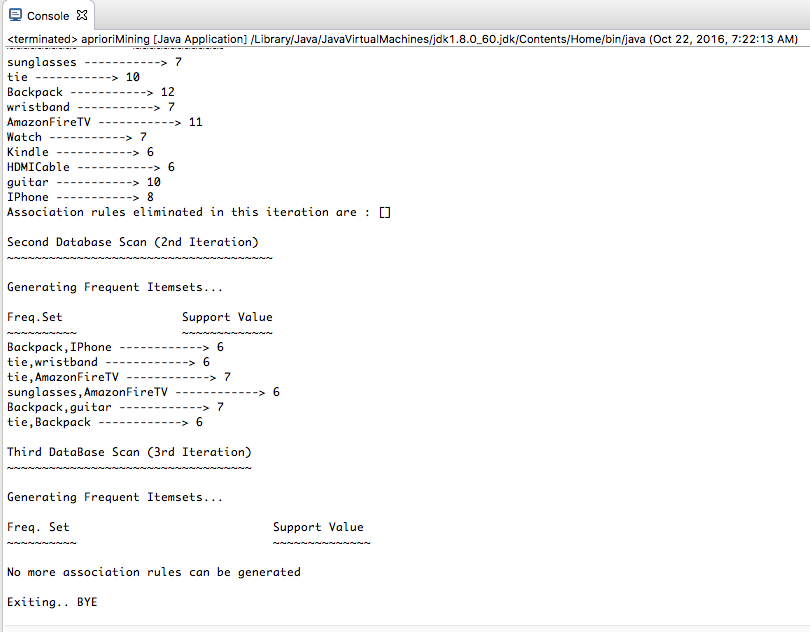
Output continued ..



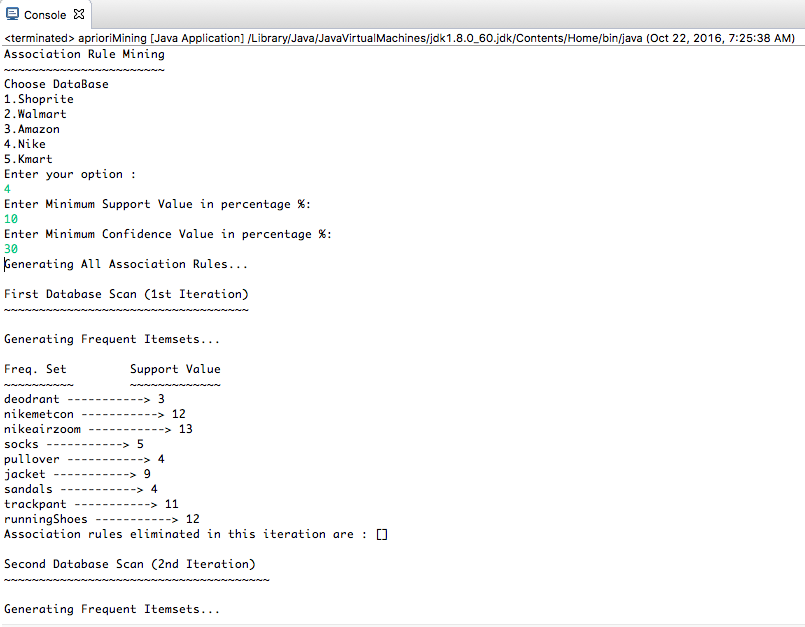
**Execution 3: Amazon database with support= 30% and confidence =40%**



Output continued ..



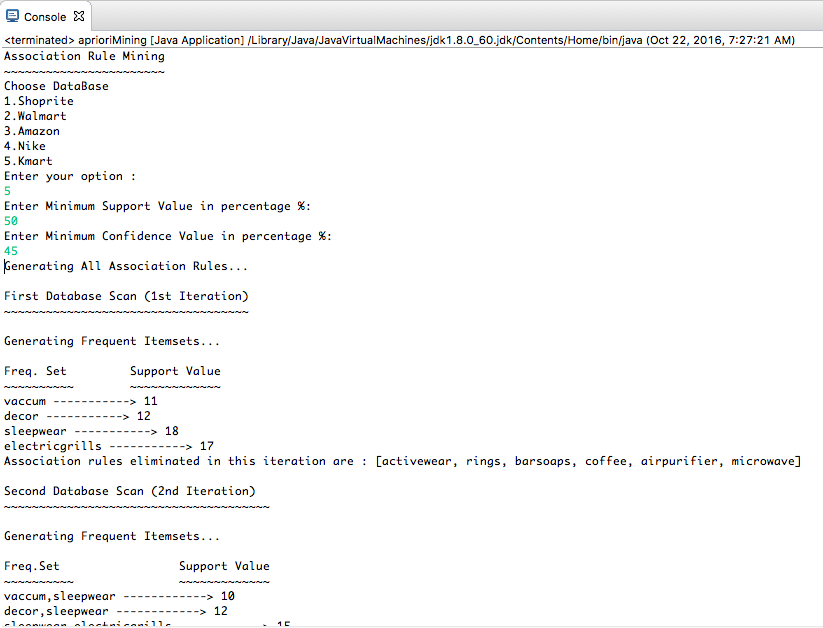
**Execution 4: Nike database with support= 10% and confidence =30%**



Output continued ..



**Execution 5: Kmart database with support= 50% and confidence =45%**



Output continued ..

