Report and screenshots of Datamining midterm project

CS 634, spring 2019

Programming language: python 3.7

Type of data files: text (.txt)

SID: 31470206

Files

In this project we have 7 files including:

- Apriori.py: which is the program
- Items.txt: which is a list of 10 items
- transactions1.txt … transactions5.txt: which are the data files

For the list of items, there are 10 common bought items chosen from Walmart. The items.txt file can be seen in Figure 1.

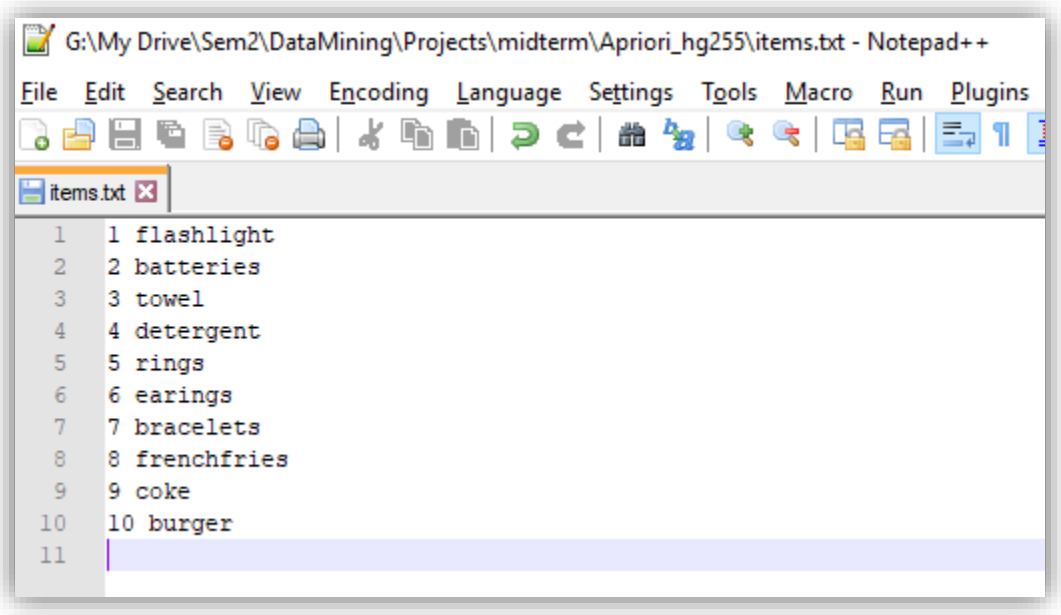

Figure 1) list of items

The transactions files were created manually. In listing the transactions, the purpose was to make meaningful shopping records and to consider how common the items are bought to get more various values for support and confidence, in order to get more interesting association rules. These files are created similar to each other and there not much difference among them; however, in **transactions3** the transaction IDs are purposely given in a different way. Each data file is simply a list of records with a number as ID assigned to each of them.

Program

I am a beginner in python, so here I just made a very simple program to perform the Apriori algorithm step by step as introduced in the class.

Initializations

The program takes three inputs:

1. name of the file containing transactions
2. the value for minimum support
3. the value for minimum confidence

```
4 file = input('Enter the name of transactions file: ') or 'transactions1.txt'
5 min_sup = float(input("Enter minimum support in percentage: ")) or 20
6 min_conf = float(input("Entern minimum confidence in percentage: ")) or 50
```

Then there is part to read data from the input file. The input file (f) is read line by line. Then each word (including the IDs) are stored in a variable called “shop” and added to the list of transactions. After having the list of records, this list is used to gather a set of different items that are in the input file. So the list of items is acquired from the transactions and not as an input to the program.

```
 8 ##################  reading data
 9 f = open(file, "r")
10 transactions = []
11 items = set()
12 for line in f:   #save shoppings in transactions
13     shop = line.split()
14     transactions.append(shop)
15     for word in re.findall(r'\w+', line):
16         if not word.isdigit(): #avoid reading transaction IDs as items
17             items.add(word) #set of items
```

Then there is a short part for initializations. Since the input values for support and confidence are given by the user in percentage, they are divided by 100 to be used in the program. Number of items and number of transactions are also stored.

```
19 ################## initializations
20 min_sup = min_sup/100
21 min_conf = min_conf/100
22 numberOfTransactions = len(transactions)
23 numberOfItems = len(items)
24 rules = []
```

functions

There are a small number of functions used alongside main() function to execute the steps of the algorithm.

```
28 def supportOf(itemset):
29     #calculate support
30     count=0
31     for i in range(numberOfTransactions):
32         if(itemset.issubset(set(transactions[i]))):
33             count += 1
34     return count/numberOfTransactions
```

This function calculates the support value of any given itemset by first counting the number of transaction records containing the item and then dividing this value by the total number of transactions.

```
37 def dictCreate(itemset):
38     # define dict to store itemset and its support
39     itemDict = {"itemset":{},"support": 0.0}
40     itemDict["itemset"] = itemset
41     itemDict["support"] = supportOf(itemset)
42     return itemDict
```

This one is used to create an instance of dictionary to make it possible to store the support of an itemset with it. Since the types of two records are different (set, float) a dictionary is used.

```
45 def joinItemSets(itemsets, length):
46     itemsetsTemp=[]
47 
48     for t in range(len(itemsets)):
49         for q in range(t+1,len(itemsets)):
50             if not itemsetsTemp.__contains__(itemsets[t].union(itemsets[q])):
51                 if (len(itemsets[t].union(itemsets[q]))==length):
52                     itemsetsTemp.append(itemsets[t].union(itemsets[q]))
53     return itemsetsTemp
```

This function is used to join itemsets of arbitrary length k and create itemsets of length k+1. All possible orders of the members of the itemset are generated and only those of the right length (k+1) that are not generated already, are added to the list of temporary itemsets. This is not the final itemset as there are still some members that need to be removed due to low support value.

```
55 def detectExtraItemsets(itemsetsTemp, nonfreqList):
56     #remove itemsets with low support that are generated by joining other itemsets
57     extraItemsets = []
58     for u in itemsetsTemp:
59         for w in nonfreqList:
60             if w.issubset(u):
61                 extraItemsets.append(u)
62     return extraItemsets
```

This function is used to detect those itemsets that are generated by joining two frequent itemsets, but being a superset of a non-frequent itemset. For example:

$$if \begin{cases} \{a,c\}\ is\ non-frequent \\ \{a,b\}\ is\ frequent \\ \{b,c\}\ is\ frequent \end{cases} \Rightarrow \{a,b\} \cup \{b,c\} = \{a,b,c\}$$

The resulting set comes from two frequent sets, but it contains a subset that is not frequent. Therefore, this kind of itemsets should not be considered as frequent.

```
65 def generateAssociationRules(frequentList, minimumConfidence):
66     #  association rules
67     for i in frequentList:
68         if len(i["itemset"]) >= 2:
69             #generate all possible tuples of each itemset
70             subs = itertools.chain(*[itertools.combinations(i["itemset"], q + 1) for q, a in enumerate(i["itemset"])])
71 
72             #rule leftSide ==> rightSide
73             for leftSide in subs:
74                 leftSide = set(leftSide)
75                 rightSide = i["itemset"].difference(leftSide)
76                 if rightSide:
77                     leftSidSet = next(item for item in frequentList if item["itemset"] == leftSide)
78                     confidence = float( i["support"] / leftSidSet["support"] )
79                     if confidence >= minimumConfidence:
80                         rules.append((leftSide, rightSide, i["support"], confidence))
81     return(rules)
```

This part is used to generate association rules based on the given frequent itemsets and their confidence value. For each set of frequent items, first all subsets of it are generated using itertools and stored as tuples in “subs” value. Then for each subset (except {} and the whole set) the complementary subset (rightSide) of it (leftSide) is generated by “difference” subtraction. The “leftSide” is replaced by the corresponding set in the frequent list to access the support of it and avoid calculating the support again. Lastly, the confidence value is calculated based on the formula and the ones with big enough confidence are added as an association rule to the list “rules”.

```
85 def printResults(associationRules):
86 
87     print("\nminimum support: %.2d %%" %(100*min_sup))
88     print("minimum confidence: %.2f %%" %(100*min_conf))
89     print("input data file: " , os.path.abspath(file))
90 
91     j = 1
92     print("\n     list of items: \n ====================== \n")
93     for i in items:
94         print(j , " " , i , "\n")
95         j += 1
96 
97     print("\n     list of transactions: \n ============================== \n")
98     for q in transactions:
99         print("ID: %d items:" %int(q[0]), end=" ")
100        print(*q[1:] , sep = ", ")
101 
102     #print all association rules
103    print("\n     Association Rules: \n ========================")
104    for i in associationRules:
105        sup = i[2]*100
106        cnf = i[3]*100
107        print(i[0] , " ===> " , i[1], " with support: %.2f %%, and confidence: %.2f %%" % (sup, cnf))
```

This function is for printing the inputs, transactions, and the association rules alongside with their support and confidence scores.

```
92 def identifyFrequentItemsets(boughtItems, transactionRecords):
93     #generate frequent itemsets
94     freqList = []
95     itemList = list(boughtItems)
96     itemsets = [set([itemList[k]]) for k in range(len(itemList))] #list of 1-itemsets
97 
98     i = 1 #length of itemsets at each run
99 
100    #itemsets longer than the longest transaction (not considering transaction ID) are not interesting
101    while i <= len(max(transactionRecords,key=len)) - 1:
102 
103        nonfreqList = []
104        itemsetsTemp=[]
105        unionsToRemove = []
106 
107        #creating frequent i-itemsets
108        for j in itemsets: # each j is an itemset
109            itemWithSup = dictCreate(j)
110            # check with minimum support
111            if supportOf(j) >= min_sup:
112                freqList.append(itemWithSup)
113            else:
114                nonfreqList.append(j)
115 
116        for k in nonfreqList: #remove itemsets with less than enough support
117            itemsets.remove(k)
118 
119        i += 1
120 
121        itemsetsTemp = joinItemSets(itemsets, i) # join i-itemsets to get (i+1)-itemsets
122 
123        unionsToRemove = detectExtraItemsets(itemsetsTemp, nonfreqList)
124 
125        for p in unionsToRemove:
126            itemsetsTemp.remove(p)
127 
128        itemsets = itemsetsTemp
129        if len(itemsets) <=1: #stop if there is only 0 or 1 itemset is left
130            break
131    return freqList
```

This function is to get the frequent itemsets. Given as input the list of all items and transaction records, the frequent i-itemsets are generated in a while loop. Starting from the 1-itemsets, the frequent ones according to the support values are stored in “freqList” and by adding 1 to i, the next itemsets are created. In any case, if one or less itemsets are remained, it breaks and returns the frequent itemsets.

```
134 def main():
135     freqList = identifyFrequentItemsets(items, transactions)
136     rules = generateAssociationRules(freqList, min_conf)
137     printResults(rules)
138 
139 if __name__ == '__main__':
140     main()
```

In the main function, first the frequent itemsets are gathered, then the association rules are calculated and printed.

Examples of outputs

1. **transactions1**

transactions1.txt

```
1 flashlight batteries
2 rings earings
3 burger coke frenchfries
4 rings bracelets
5 batteries flashlight
6 rings earings bracelets
7 flashlight batteries towel
8 burger coke
9 detergent flashlight batteries
10 earings rings bracelets
11 burger frenchfries coke
12 flashlight batteries frenchfries coke
13 towel detergent rings earings bracelets
14 burger frenchfries
15 batteries flashlight
16 rings bracelets
17 coke burger towel
18 batteries flashlight coke burger
19 earings bracelets rings
20 batteries frenchfries rings earings bracelets
```

```
    list of items:
 ======================

1    batteries

2    frenchfries

3    earings

4    detergent

5    coke

6    burger

7    bracelets

8    rings

9    towel

10    flashlight
```

```
    list of transactions:
 ==============================

ID: 1 items: flashlight, batteries
ID: 2 items: rings, earings
ID: 3 items: burger, coke, frenchfries
ID: 4 items: rings, bracelets
ID: 5 items: batteries, flashlight
ID: 6 items: rings, earings, bracelets
ID: 7 items: flashlight, batteries, towel
ID: 8 items: burger, coke
ID: 9 items: detergent, flashlight, batteries
ID: 10 items: earings, rings, bracelets
ID: 11 items: burger, frenchfries, coke
ID: 12 items: flashlight, batteries, frenchfries, coke
ID: 13 items: towel, detergent, rings, earings, bracelets
ID: 14 items: burger, frenchfries
ID: 15 items: batteries, flashlight
ID: 16 items: rings, bracelets
ID: 17 items: coke, burger, towel
ID: 18 items: batteries, flashlight, coke, burger
ID: 19 items: earings, bracelets, rings
ID: 20 items: batteries, frenchfries, rings, earings, bracelets
```

So these items are stored in “items” list and the transactions are stored in “transactions”. Then they are input to:

```
freqList = identifyFrequentItemsets(items, transactions)
```

in function `identifyFrequentItemsets(items, transactions)`, a list of 1-itemsets is created and called “itemsets”. Then if its support is big enough, it is added to “freqList” as a dict with its support value. Otherwise it is stored in “nonfreqList”. Then the non-frequent items are removed from itemsets and the remainings are joined together. After joining, the sets that have a subset in non-frequent list are also removed (which doesn’t occur in this example).

```
initial 2-itemsets:
{'batteries', 'frenchfries'}
{'batteries', 'earings'}
{'batteries', 'coke'}
{'batteries', 'burger'}
{'batteries', 'bracelets'}
{'rings', 'batteries'}
{'batteries', 'flashlight'}
{'earings', 'frenchfries'}
{'coke', 'frenchfries'}
{'burger', 'frenchfries'}
{'bracelets', 'frenchfries'}
{'rings', 'frenchfries'}
{'flashlight', 'frenchfries'}
{'coke', 'earings'}
{'burger', 'earings'}
{'earings', 'bracelets'}
{'rings', 'earings'}
{'flashlight', 'earings'}
{'coke', 'burger'}
{'coke', 'bracelets'}
{'rings', 'coke'}
{'coke', 'flashlight'}
{'burger', 'bracelets'}
{'rings', 'burger'}
{'burger', 'flashlight'}
{'rings', 'bracelets'}
{'flashlight', 'bracelets'}
{'rings', 'flashlight'}
```

```
remained 2-itemsets:
{'batteries', 'flashlight'}
{'earings', 'bracelets'}
{'rings', 'earings'}
{'coke', 'burger'}
{'rings', 'bracelets'}
```

Finally, the association rules are generated using the frequent list. With **support = 20%** and with **0% confidence** these rules all rules would be accepted:

```
     Association Rules:
 ========================
{'batteries'}  ===>  {'flashlight'}  with support: 35.00 %, and confidence: 87.50 %
{'flashlight'}  ===>  {'batteries'}  with support: 35.00 %, and confidence: 100.00 %
{'earings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 83.33 %
{'bracelets'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 71.43 %
{'rings'}  ===>  {'earings'}  with support: 30.00 %, and confidence: 75.00 %
{'earings'}  ===>  {'rings'}  with support: 30.00 %, and confidence: 100.00 %
{'coke'}  ===>  {'burger'}  with support: 25.00 %, and confidence: 83.33 %
{'burger'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 83.33 %
{'rings'}  ===>  {'bracelets'}  with support: 35.00 %, and confidence: 87.50 %
{'bracelets'}  ===>  {'rings'}  with support: 35.00 %, and confidence: 100.00 %
```

But for example a minimum **confidence of 85%** and **support = 20%** will filter some of the results:

```
     Association Rules:
 ========================
{'batteries'}  ===>  {'flashlight'}  with support: 35.00 %, and confidence: 87.50 %
{'flashlight'}  ===>  {'batteries'}  with support: 35.00 %, and confidence: 100.00 %
{'earings'}  ===>  {'rings'}  with support: 30.00 %, and confidence: 100.00 %
{'rings'}  ===>  {'bracelets'}  with support: 35.00 %, and confidence: 87.50 %
{'bracelets'}  ===>  {'rings'}  with support: 35.00 %, and confidence: 100.00 %
```

Increasing the amount of support to for example **35%** can also reduce the number of rules:

```
     Association Rules:
 ========================
{'batteries'}  ===>  {'flashlight'}  with support: 35.00 %, and confidence: 87.50 %
{'flashlight'}  ===>  {'batteries'}  with support: 35.00 %, and confidence: 100.00 %
{'rings'}  ===>  {'bracelets'}  with support: 35.00 %, and confidence: 87.50 %
{'bracelets'}  ===>  {'rings'}  with support: 35.00 %, and confidence: 100.00 %
```

2. transactions2

transactions2.txt

```
1 towel coke burger
2 frenchfries burger coke
3 flashlight batteries
4 earings bracelets
5 frenchfries coke burger flashlight batteries
6 burger frenchfries coke towel detergent
7 flashlight rings earings batteries
8 earings bracelets rings batteries flashlight
9 flashlight batteries coke frenchfries burger
10 frenchfries coke
11 coke batteries burger flashlight frenchfries
12 earings rings towel detergent bracelets
13 detergent coke frenchfries
14 bracelets earings towel bracelets
15 batteries flashlight
16 coke frenchfries
17 bracelets earings burger frenchfries coke rings
18 coke frenchfries burger
19 batteries flashlight
20 bracelets rings detergent burger coke frenchfries
```

```
     list of transactions:
 ==============================

ID: 1 items: towel, coke, burger
ID: 2 items: frenchfries, burger, coke
ID: 3 items: flashlight, batteries
ID: 4 items: earings, bracelets
ID: 5 items: frenchfries, coke, burger, flashlight, batteries
ID: 6 items: burger, frenchfries, coke, towel, detergent
ID: 7 items: flashlight, rings, earings, batteries
ID: 8 items: earings, bracelets, rings, batteries, flashlight
ID: 9 items: flashlight, batteries, coke, frenchfries, burger
ID: 10 items: frenchfries, coke
ID: 11 items: coke, batteries, burger, flashlight, frenchfries
ID: 12 items: earings, rings, towel, detergent, bracelets
ID: 13 items: detergent, coke, frenchfries
ID: 14 items: bracelets, earings, towel, bracelets
ID: 15 items: batteries, flashlight
ID: 16 items: coke, frenchfries
ID: 17 items: bracelets, earings, burger, frenchfries, coke, rings
ID: 18 items: coke, frenchfries, burger
ID: 19 items: batteries, flashlight
ID: 20 items: bracelets, rings, detergent, burger, coke, frenchfries
```

Support = 20%, confidence = 50%

```
     Association Rules:
 ========================
{'batteries'}  ===>  {'flashlight'}  with support: 40.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 40.00 %, and confidence: 100.00 %
{'coke'}  ===>  {'frenchfries'}  with support: 55.00 %, and confidence: 91.67 %
{'frenchfries'}  ===>  {'coke'}  with support: 55.00 %, and confidence: 100.00 %
{'burger'}  ===>  {'frenchfries'}  with support: 40.00 %, and confidence: 88.89 %
{'frenchfries'}  ===>  {'burger'}  with support: 40.00 %, and confidence: 72.73 %
{'earings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 83.33 %
{'bracelets'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 83.33 %
{'rings'}  ===>  {'earings'}  with support: 20.00 %, and confidence: 80.00 %
{'earings'}  ===>  {'rings'}  with support: 20.00 %, and confidence: 66.67 %
{'coke'}  ===>  {'burger'}  with support: 45.00 %, and confidence: 75.00 %
{'burger'}  ===>  {'coke'}  with support: 45.00 %, and confidence: 100.00 %
{'rings'}  ===>  {'bracelets'}  with support: 20.00 %, and confidence: 80.00 %
{'bracelets'}  ===>  {'rings'}  with support: 20.00 %, and confidence: 66.67 %
{'coke'}  ===>  {'burger', 'frenchfries'}  with support: 40.00 %, and confidence: 66.67 %
{'burger'}  ===>  {'coke', 'frenchfries'}  with support: 40.00 %, and confidence: 88.89 %
{'frenchfries'}  ===>  {'coke', 'burger'}  with support: 40.00 %, and confidence: 72.73 %
{'coke', 'burger'}  ===>  {'frenchfries'}  with support: 40.00 %, and confidence: 88.89 %
{'coke', 'frenchfries'}  ===>  {'burger'}  with support: 40.00 %, and confidence: 72.73 %
{'burger', 'frenchfries'}  ===>  {'coke'}  with support: 40.00 %, and confidence: 100.00 %
```

Support = 40%, confidence = 70%

```
     Association Rules:
 ==========================
{'batteries'}  ===>  {'flashlight'}  with support: 40.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 40.00 %, and confidence: 100.00 %
{'coke'}  ===>  {'frenchfries'}  with support: 55.00 %, and confidence: 91.67 %
{'frenchfries'}  ===>  {'coke'}  with support: 55.00 %, and confidence: 100.00 %
{'burger'}  ===>  {'frenchfries'}  with support: 40.00 %, and confidence: 88.89 %
{'frenchfries'}  ===>  {'burger'}  with support: 40.00 %, and confidence: 72.73 %
{'coke'}  ===>  {'burger'}  with support: 45.00 %, and confidence: 75.00 %
{'burger'}  ===>  {'coke'}  with support: 45.00 %, and confidence: 100.00 %
```

4. transactions3

transactions3.txt

```
100 burger frenchfries
200 towel detergent
300 rings earings batteries
400 flashlight batteries coke
500 towel coke frenchfries burger detergent
600 earings bracelets rings
700 towel detergent frenchfries coke
800 towel batteries flashlight detergent
900 rings bracelets
1000 detergent coke frenchfries towel
1100 burger bracelets frenchfries coke
1200 towel rings earings detergent
1300 detergent bracelets earings burger
1400 towel flashlight batteries
1500 coke frenchfries burger
1600 flashlight towel detergent batteries
1700 burger frenchfries coke towel detergent rings bracelets earings
1800 towel detergent
1900 coke bracelets earings burger
2000 batteries flashlight coke burger frenchfries
```

```
    list of transactions:
 ==============================

ID: 100 items: burger, frenchfries
ID: 200 items: towel, detergent
ID: 300 items: rings, earings, batteries
ID: 400 items: flashlight, batteries, coke
ID: 500 items: towel, coke, frenchfries, burger, detergent
ID: 600 items: earings, bracelets, rings
ID: 700 items: towel, detergent, frenchfries, coke
ID: 800 items: towel, batteries, flashlight, detergent
ID: 900 items: rings, bracelets
ID: 1000 items: detergent, coke, frenchfries, towel
ID: 1100 items: burger, bracelets, frenchfries, coke
ID: 1200 items: towel, rings, earings, detergent
ID: 1300 items: detergent, bracelets, earings, burger
ID: 1400 items: towel, flashlight, batteries
ID: 1500 items: coke, frenchfries, burger
ID: 1600 items: flashlight, towel, detergent, batteries
ID: 1700 items: burger, frenchfries, coke, towel, detergent, rings, bracelets, earings
ID: 1800 items: towel, detergent
ID: 1900 items: coke, bracelets, earings, burger
ID: 2000 items: batteries, flashlight, coke, burger, frenchfries
```

In this input file the transactions IDs are different the other files.

Support = 20%, confidence = 50%

In this case, number of items in each transaction is more than previous files and that is why a 20% support is not going to filter much of the records. Because each item has probably appeared in more than one-fifth of the transactions. By increasing the value of support and confidence we can get more meaningful association rules.

```
    Association Rules:
 ========================
{'rings'}  ===>  {'earings'}  with support: 20.00 %, and confidence: 80.00 %
{'earings'}  ===>  {'rings'}  with support: 20.00 %, and confidence: 66.67 %
{'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 100.00 %
{'batteries'}  ===>  {'flashlight'}  with support: 25.00 %, and confidence: 83.33 %
{'earings'}  ===>  {'bracelets'}  with support: 20.00 %, and confidence: 66.67 %
{'bracelets'}  ===>  {'earings'}  with support: 20.00 %, and confidence: 66.67 %
{'burger'}  ===>  {'bracelets'}  with support: 20.00 %, and confidence: 50.00 %
{'bracelets'}  ===>  {'burger'}  with support: 20.00 %, and confidence: 66.67 %
{'detergent'}  ===>  {'towel'}  with support: 45.00 %, and confidence: 90.00 %
{'towel'}  ===>  {'detergent'}  with support: 45.00 %, and confidence: 90.00 %
{'frenchfries'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 50.00 %
{'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 50.00 %
{'coke'}  ===>  {'frenchfries'}  with support: 35.00 %, and confidence: 77.78 %
{'frenchfries'}  ===>  {'coke'}  with support: 35.00 %, and confidence: 87.50 %
{'coke'}  ===>  {'burger'}  with support: 30.00 %, and confidence: 66.67 %
{'burger'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 75.00 %
{'burger'}  ===>  {'frenchfries'}  with support: 30.00 %, and confidence: 75.00 %
{'frenchfries'}  ===>  {'burger'}  with support: 30.00 %, and confidence: 75.00 %
{'coke', 'detergent'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries'}  ===>  {'detergent', 'towel'}  with support: 20.00 %, and confidence: 50.00 %
{'detergent', 'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries'}  ===>  {'coke', 'detergent'}  with support: 20.00 %, and confidence: 50.00 %
{'coke', 'detergent'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'frenchfries'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 57.14 %
{'detergent', 'frenchfries'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries'}  ===>  {'coke', 'towel'}  with support: 20.00 %, and confidence: 50.00 %
{'coke', 'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 57.14 %
{'coke', 'towel'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
{'coke'}  ===>  {'burger', 'frenchfries'}  with support: 25.00 %, and confidence: 55.56 %
{'burger'}  ===>  {'coke', 'frenchfries'}  with support: 25.00 %, and confidence: 62.50 %
{'frenchfries'}  ===>  {'coke', 'burger'}  with support: 25.00 %, and confidence: 62.50 %
{'coke', 'burger'}  ===>  {'frenchfries'}  with support: 25.00 %, and confidence: 83.33 %
{'coke', 'frenchfries'}  ===>  {'burger'}  with support: 25.00 %, and confidence: 71.43 %
{'burger', 'frenchfries'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 83.33 %
{'frenchfries'}  ===>  {'coke', 'detergent', 'towel'}  with support: 20.00 %, and confidence: 50.00 %
{'coke', 'frenchfries'}  ===>  {'detergent', 'towel'}  with support: 20.00 %, and confidence: 57.14 %
{'coke', 'detergent'}  ===>  {'frenchfries', 'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'towel'}  ===>  {'detergent', 'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries'}  ===>  {'coke', 'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'coke', 'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent', 'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'frenchfries', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent', 'towel'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries', 'towel'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
```

Support = 20%, confidence = 85%

```
     Association Rules:
 ========================
{'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 100.00 %
{'detergent'}  ===>  {'towel'}  with support: 45.00 %, and confidence: 90.00 %
{'towel'}  ===>  {'detergent'}  with support: 45.00 %, and confidence: 90.00 %
{'frenchfries'}  ===>  {'coke'}  with support: 35.00 %, and confidence: 87.50 %
{'coke', 'detergent'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'towel'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent'}  ===>  {'frenchfries', 'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'towel'}  ===>  {'detergent', 'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries'}  ===>  {'coke', 'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'frenchfries', 'towel'}  ===>  {'coke', 'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent', 'frenchfries'}  ===>  {'towel'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'frenchfries', 'towel'}  ===>  {'detergent'}  with support: 20.00 %, and confidence: 100.00 %
{'coke', 'detergent', 'towel'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'detergent', 'frenchfries', 'towel'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 100.00 %
```

5. transactions4

```
1 burger coke frenchfries
2 towel detergent earings
3 bracelets rings earings
4 frenchfries coke flashlight
5 batteries flashlight
6 rings bracelets
7 detergent towel burger coke
8 flashlight frenchfries coke batteries
9 bracelets rings earings batteries
10 detergent towel frenchfries coke burger
11 batteries flashlight towel
12 coke frenchfries batteries
13 coke rings bracelets earings
14 earings detergent towel
15 burger frenchfries
16 flashlight batteries earings rings
17 coke burger
18 frenchfries burger coke towel detergent
19 rings bracelets towel
20 flashlight batteries burger coke
```

```
     list of transactions:
 ==============================

ID: 1 items: burger, coke, frenchfries
ID: 2 items: towel, detergent, earings
ID: 3 items: bracelets, rings, earings
ID: 4 items: frenchfries, coke, flashlight
ID: 5 items: batteries, flashlight
ID: 6 items: rings, bracelets
ID: 7 items: detergent, towel, burger, coke
ID: 8 items: flashlight, frenchfries, coke, batteries
ID: 9 items: bracelets, rings, earings, batteries
ID: 10 items: detergent, towel, frenchfries, coke, burger
ID: 11 items: batteries, flashlight, towel
ID: 12 items: coke, frenchfries, batteries
ID: 13 items: coke, rings, bracelets, earings
ID: 14 items: earings, detergent, towel
ID: 15 items: burger, frenchfries
ID: 16 items: flashlight, batteries, earings, rings
ID: 17 items: coke, burger
ID: 18 items: frenchfries, burger, coke, towel, detergent
ID: 19 items: rings, bracelets, towel
ID: 20 items: flashlight, batteries, burger, coke
```

Support = 20%, confidence = 50%

```
    Association Rules:
 ========================
{'rings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 83.33 %
{'bracelets'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 100.00 %
{'rings'}  ===>  {'earings'}  with support: 20.00 %, and confidence: 66.67 %
{'earings'}  ===>  {'rings'}  with support: 20.00 %, and confidence: 66.67 %
{'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 83.33 %
{'batteries'}  ===>  {'flashlight'}  with support: 25.00 %, and confidence: 71.43 %
{'detergent'}  ===>  {'towel'}  with support: 25.00 %, and confidence: 100.00 %
{'towel'}  ===>  {'detergent'}  with support: 25.00 %, and confidence: 71.43 %
{'coke'}  ===>  {'frenchfries'}  with support: 30.00 %, and confidence: 60.00 %
{'frenchfries'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
{'coke'}  ===>  {'burger'}  with support: 30.00 %, and confidence: 60.00 %
{'burger'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
{'burger'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 57.14 %
{'frenchfries'}  ===>  {'burger'}  with support: 20.00 %, and confidence: 57.14 %
```

```
remained 2-itemsets:
{'rings', 'bracelets'}
{'rings', 'earings'}
{'flashlight', 'batteries'}
{'detergent', 'towel'}
{'coke', 'frenchfries'}
{'coke', 'burger'}
{'burger', 'frenchfries'}
```

When the threshold for support is increased at the beginning, most items are removed in the steps to get to the resulting rules.

Support = 25%, confidence = 50%

```
    Association Rules:
 ========================
{'rings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 83.33 %
{'bracelets'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 83.33 %
{'batteries'}  ===>  {'flashlight'}  with support: 25.00 %, and confidence: 71.43 %
{'detergent'}  ===>  {'towel'}  with support: 25.00 %, and confidence: 100.00 %
{'towel'}  ===>  {'detergent'}  with support: 25.00 %, and confidence: 71.43 %
{'coke'}  ===>  {'frenchfries'}  with support: 30.00 %, and confidence: 60.00 %
{'frenchfries'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
{'coke'}  ===>  {'burger'}  with support: 30.00 %, and confidence: 60.00 %
{'burger'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
```

```
remained 2-itemsets:
{'rings', 'bracelets'}
{'flashlight', 'batteries'}
{'detergent', 'towel'}
{'coke', 'frenchfries'}
{'coke', 'burger'}
```

Support = 25%, confidence = 85%

```
     Association Rules:
 ========================
{'rings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 83.33 %
{'bracelets'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 83.33 %
{'detergent'}  ===>  {'towel'}  with support: 25.00 %, and confidence: 100.00 %
{'frenchfries'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
{'burger'}  ===>  {'coke'}  with support: 30.00 %, and confidence: 85.71 %
```

Here the predictions can be positive on probability of "rings, bracelets" and also "burger, coke" and "flashlight, batteries" to be bought together most of the times.

6. transactions5

```
1 burger frenchfries
2 rings earings bracelets
3 batteries flashlight coke frenchfries
4 frenchfries coke burger rings earings
5 towel frenchfries coke flashlight batteries
6 batteries flashlight coke burger
7 coke burger frenchfries towel
8 earings bracelets
9 rings earings bracelets coke frenchfries
10 rings earings bracelets
11 burger coke flashlight batteries
12 towel coke frenchfries burger
13 burger frenchfries coke rings earings
14 flashlight batteries
15 detergent towel frenchfries burger
16 burger frenchfries
17 rings bracelets earings
18 burger coke
19 bracelets rings earings flashlight batteries coke
20 coke rings earings burger
```

```
    list of transactions:
 ==============================

ID: 1 items: burger, frenchfries
ID: 2 items: rings, earings, bracelets
ID: 3 items: batteries, flashlight, coke, frenchfries
ID: 4 items: frenchfries, coke, burger, rings, earings
ID: 5 items: towel, frenchfries, coke, flashlight, batteries
ID: 6 items: batteries, flashlight, coke, burger
ID: 7 items: coke, burger, frenchfries, towel
ID: 8 items: earings, bracelets
ID: 9 items: rings, earings, bracelets, coke, frenchfries
ID: 10 items: rings, earings, bracelets
ID: 11 items: burger, coke, flashlight, batteries
ID: 12 items: towel, coke, frenchfries, burger
ID: 13 items: burger, frenchfries, coke, rings, earings
ID: 14 items: flashlight, batteries
ID: 15 items: detergent, towel, frenchfries, burger
ID: 16 items: burger, frenchfries
ID: 17 items: rings, bracelets, earings
ID: 18 items: burger, coke
ID: 19 items: bracelets, rings, earings, flashlight, batteries, coke
ID: 20 items: coke, rings, earings, burger
```

Support = 20%, confidence = 50%

```
    Association Rules:
 ========================
{'rings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 62.50 %
{'bracelets'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 83.33 %
{'rings'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 62.50 %
{'rings'}  ===>  {'earings'}  with support: 40.00 %, and confidence: 100.00 %
{'earings'}  ===>  {'rings'}  with support: 40.00 %, and confidence: 88.89 %
{'earings'}  ===>  {'bracelets'}  with support: 30.00 %, and confidence: 66.67 %
{'bracelets'}  ===>  {'earings'}  with support: 30.00 %, and confidence: 100.00 %
{'batteries'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 83.33 %
{'flashlight'}  ===>  {'batteries'}  with support: 30.00 %, and confidence: 100.00 %
{'batteries'}  ===>  {'flashlight'}  with support: 30.00 %, and confidence: 100.00 %
{'towel'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 100.00 %
{'coke'}  ===>  {'frenchfries'}  with support: 35.00 %, and confidence: 58.33 %
{'frenchfries'}  ===>  {'coke'}  with support: 35.00 %, and confidence: 70.00 %
{'earings'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 55.56 %
{'coke'}  ===>  {'burger'}  with support: 40.00 %, and confidence: 66.67 %
{'burger'}  ===>  {'coke'}  with support: 40.00 %, and confidence: 72.73 %
{'flashlight'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 83.33 %
{'burger'}  ===>  {'frenchfries'}  with support: 35.00 %, and confidence: 63.64 %
{'frenchfries'}  ===>  {'burger'}  with support: 35.00 %, and confidence: 70.00 %
{'rings'}  ===>  {'bracelets', 'earings'}  with support: 25.00 %, and confidence: 62.50 %
{'earings'}  ===>  {'rings', 'bracelets'}  with support: 25.00 %, and confidence: 55.56 %
{'bracelets'}  ===>  {'rings', 'earings'}  with support: 25.00 %, and confidence: 83.33 %
{'rings', 'earings'}  ===>  {'bracelets'}  with support: 25.00 %, and confidence: 62.50 %
{'rings', 'bracelets'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 100.00 %
{'bracelets', 'earings'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 83.33 %
{'rings'}  ===>  {'coke', 'earings'}  with support: 25.00 %, and confidence: 62.50 %
{'earings'}  ===>  {'rings', 'coke'}  with support: 25.00 %, and confidence: 55.56 %
{'rings', 'coke'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 100.00 %
{'rings', 'earings'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 62.50 %
{'coke', 'earings'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'coke', 'batteries'}  with support: 25.00 %, and confidence: 83.33 %
{'batteries'}  ===>  {'coke', 'flashlight'}  with support: 25.00 %, and confidence: 83.33 %
{'coke', 'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 100.00 %
{'coke', 'batteries'}  ===>  {'flashlight'}  with support: 25.00 %, and confidence: 100.00 %
{'flashlight', 'batteries'}  ===>  {'coke'}  with support: 25.00 %, and confidence: 83.33 %
{'coke', 'burger'}  ===>  {'frenchfries'}  with support: 20.00 %, and confidence: 50.00 %
{'coke', 'frenchfries'}  ===>  {'burger'}  with support: 20.00 %, and confidence: 57.14 %
{'burger', 'frenchfries'}  ===>  {'coke'}  with support: 20.00 %, and confidence: 57.14 %
```

Like transactions3 this file also consists of more items in its records. The items "towel" and "detergent" are occurred rarely and therefore, they are not appeared in the final rules more than once, even with low support and confidence values. By increasing these values the final rules become more meaningful.

Support = 25%, confidence = 90%

```
    Association Rules:
 ========================
{'rings'}  ===>  {'earings'}  with support: 40.00 %, and confidence: 100.00 %
{'bracelets'}  ===>  {'earings'}  with support: 30.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 30.00 %, and confidence: 100.00 %
{'batteries'}  ===>  {'flashlight'}  with support: 30.00 %, and confidence: 100.00 %
{'rings', 'bracelets'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 100.00 %
{'rings', 'coke'}  ===>  {'earings'}  with support: 25.00 %, and confidence: 100.00 %
{'coke', 'earings'}  ===>  {'rings'}  with support: 25.00 %, and confidence: 100.00 %
{'coke', 'flashlight'}  ===>  {'batteries'}  with support: 25.00 %, and confidence: 100.00 %
{'coke', 'batteries'}  ===>  {'flashlight'}  with support: 25.00 %, and confidence: 100.00 %
```

Support = 30%, confidence = 90%

```
     Association Rules:
 ========================
{'rings'}  ===>  {'earings'}  with support: 40.00 %, and confidence: 100.00 %
{'bracelets'}  ===>  {'earings'}  with support: 30.00 %, and confidence: 100.00 %
{'flashlight'}  ===>  {'batteries'}  with support: 30.00 %, and confidence: 100.00 %
{'batteries'}  ===>  {'flashlight'}  with support: 30.00 %, and confidence: 100.00 %
```

As seen in the results, one can predict that customers usually buy "rings" and "earings" together. In addition, items "flashlight" and "batteries" are commonly bought together.