

Data Mining

Project #1 Association Analysis

電機所碩一 謝宜廷 N26094396



The requirements of homework:

- Test the algorithm on at least 2 Datasets
- Implement the Apriori Algorithm and FP-growth
- The output of program must include the Frequent Patterns and Rules
- Hand in the analysis report

In this project, I tried to implement Apriori Algorithm and FP-growth. There are two attachments of this project, one is the implementation of Apriori Algorithm, and the other is the implementation of FP-growth. I used the data generated by IBM Quest Synthetic Data Generator, to test the performance of the program. The implementation and performance of the two algorithms will be detailed in the following part.

Figure 1 shows the steps of Association Analysis. First, we have to import and reshape the data from a .csv file. We delete the sets which supports are lower than the minimum support, and then get the frequent set. After getting frequent rules, we can generate the association rules.

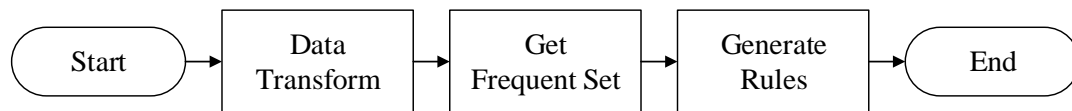


Fig. 1 The flow chart of Association Analysis

1. Apriori Algorithm

Figure 2 shows step “Get Frequent Set” of Apriori Algorithm. First, we delete the items which support lower than minimum support. And use the rest items to generate a bigger set. Repeat these steps until the rest items less than 1, then we can get the frequent items of this data.

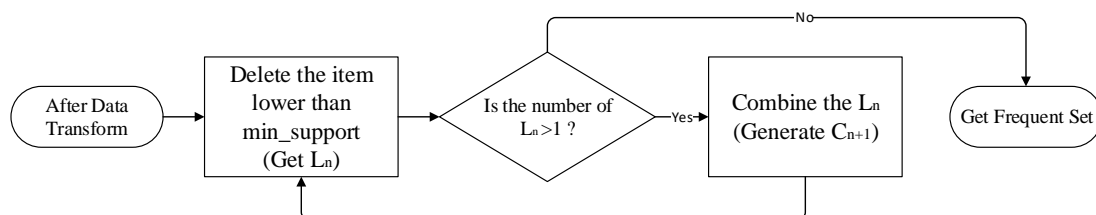


Fig. 2 The process “Get Frequent Set” of Apriori Algorithm

Figure 3 shows step “Generate Rules” of Apriori Algorithm. And I realize it by implement recursion function. First, I only choose an item from the frequent set to be the predicted set, and then calculate the confidence of the rule. If the confidence of the rule higher than minimum confidence, then another item in the frequent set will be added to be the predicted item. Repeat the steps until the confidence of the rule lower than the minimum confidence.

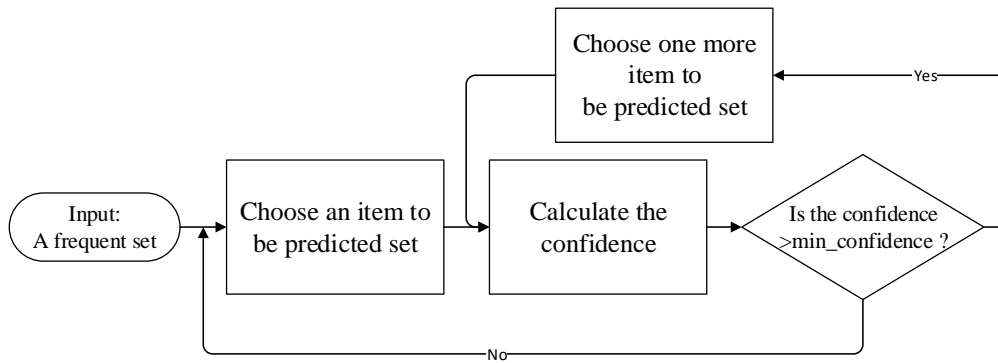


Fig. 3 The process of “Generate Rules”

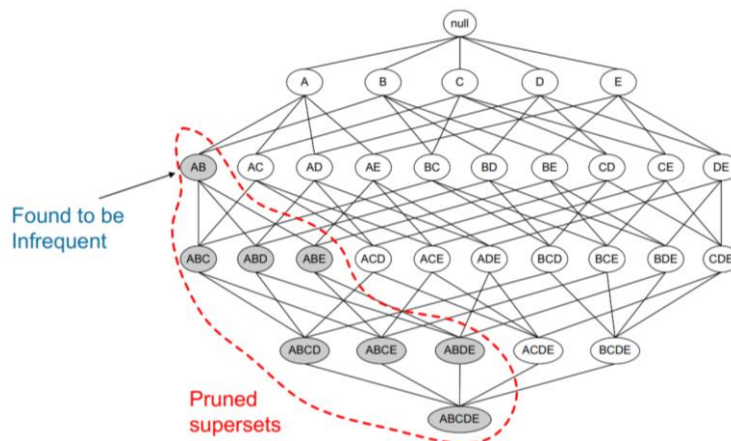


Fig.4 Generate Rules by using Apriori algorithm

2. FP-growth Algorithm

Figure 5 shows the process of FP-growth algorithm. After getting the pattern base of each item, I calculate the supports of the items in pattern base. And I generate the combinations of the item to get the frequent sets. In this project, I use Apriori algorithm to generate the rules after getting frequent sets by FP-growth.

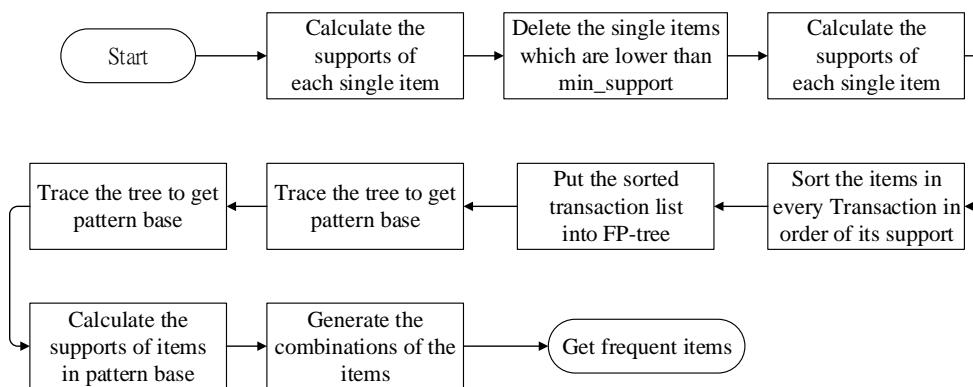


Fig.5 The process “Get Frequent Set” of FP-growth algorithm

3. Experiment Result

As shown in the following figure, the result of the experiment shows that the performance of the Apriori algorithm is better than the FP-growth. I think I didn’t do well on practicing FP-growth algorithm.

(1) High support, high confidence (min_support=400, min_confidence=0.5)

➤ testdata.csv

Time to find the frequent sets:

Apriori Algorithm: 1.43550705909729sec

FP-growth Algorithm: 1.2936785221099854sec

```

(frozenset({28}), 582)
(frozenset({14}), 485)
(frozenset({8}), 603)
(frozenset({11}), 625)
(frozenset({17}), 542)
(frozenset({9}), 492)
(frozenset({3}), 576)
(frozenset({29}), 478)
(frozenset({8, 11}), 415)
{frozenset({28}): 582, frozenset({14}): 485, frozenset({8}): 603, frozenset({11}): 625, frozenset({17}): 542, frozenset({9}): 492, frozenset({3}): 576, frozenset({29}): 478, frozenset({8, 11}): 415}
  
```

Fig.6 Frequent sets of high support, high confidence in testdata

```

(frozenset({11}), frozenset({8})) 0.664
(frozenset({8}), frozenset({11})) 0.6882255389718076
  
```

Fig.7 Rules of high support, high confidence in testdata

➤ testdata2.csv

Time to find the frequent sets:

Apriori Algorithm: 2.847632646560669 sec

FP-growth Algorithm: 4.545107364654541 sec

```

33 # print ('Length of Rules',len(rules_dic))
34 # for key,values in rules_dic.items():
35 #     print (key,values)
36
Length of frequent sets 87
{frozenset({28}): 1175, frozenset({25}): 578, frozenset({10}): 506, frozenset({18}): 412, frozenset({23}): 752, frozenset({1
4}): 967, frozenset({8}): 1180, frozenset({13}): 484, frozenset({11}): 1238, frozenset({17}): 1125, frozenset({9}): 966, frozen
set({26}): 622, frozenset({5}): 618, frozenset({12}): 509, frozenset({7}): 630, frozenset({3}): 1135, frozenset({29}): 954, fro
zenset({6}): 450, frozenset({19}): 461, frozenset({21}): 765, frozenset({0}): 598, frozenset({29, 14}): 507, frozenset({3, 2
1}): 462, frozenset({8, 7}): 408, frozenset({9, 21}): 401, frozenset({9, 14}): 497, frozenset({3, 11}): 756, frozenset({11, 1
4}): 622, frozenset({8, 23}): 464, frozenset({3, 23}): 469, frozenset({3, 14}): 589, frozenset({8, 3}): 719, frozenset({11, 2
3}): 493, frozenset({3, 28}): 702, frozenset({8, 28}): 761, frozenset({17, 29}): 587, frozenset({17, 7}): 404, frozenset({9, 2
9}): 511, frozenset({11, 29}): 657, frozenset({17, 9}): 562, frozenset({11, 5}): 407, frozenset({9, 11}): 636, frozenset({28, 2
1}): 478, frozenset({17, 14}): 576, frozenset({17, 28}): 702, frozenset({8, 29}): 624, frozenset({9, 28}): 610, frozenset({28, 2
3}): 482, frozenset({9, 3}): 593, frozenset({26, 11}): 408, frozenset({17, 3}): 696, frozenset({28, 14}): 598, frozenset({11, 7
}): 420, frozenset({8, 11}): 778, frozenset({8, 9}): 625, frozenset({17, 23}): 456, frozenset({8, 14}): 607, frozenset({17, 2
1}): 468, frozenset({8, 21}): 474, frozenset({3, 29}): 583, frozenset({11, 21}): 506, frozenset({11, 28}): 769, frozenset({17, 11
}): 744, frozenset({8, 0}): 402, frozenset({28, 29}): 617, frozenset({8, 17}): 722, frozenset({11, 17, 3}): 485, frozenset
({3, 11, 29}): 417, frozenset({17, 3, 28}): 440, frozenset({9, 11, 28}): 417, frozenset({11, 9, 3}): 412, frozenset({8, 17, 3
}): 470, frozenset({8, 9, 11}): 426, frozenset({11, 28, 29}): 439, frozenset({8, 17, 11}): 500, frozenset({8, 11, 29}): 438, f
rozenset({17, 11, 28}): 476, frozenset({8, 9, 28}): 420, frozenset({17, 11, 29}): 423, frozenset({8, 11, 28}): 511, frozenset
({8, 3, 28}): 475, frozenset({8, 17, 29}): 407, frozenset({8, 11, 14}): 415, frozenset({8, 28, 29}): 419, frozenset({3, 8, 1
1}): 491, frozenset({8, 17, 28}): 477, frozenset({3, 11, 28}): 478

```

Fig.8 Frequent sets of high support, high confidence in testdata2

```

30
31 # 印出結果
32 print ('Length of Rules',len(rules_dic))
33 for key,values in rules_dic.items():
34     print (key,values)
35
Length of Rules 135
(frozenset({17, 3}), frozenset({11})) 0.6968398804597702
(frozenset({3, 11}), frozenset({17})) 0.6415343915343915
(frozenset({17, 11}), frozenset({3})) 0.6518817204301075
(frozenset({11, 29}), frozenset({3})) 0.634703196347032
(frozenset({3, 29}), frozenset({11})) 0.7152658662092625
(frozenset({11, 3}), frozenset({29})) 0.5515873015873016
(frozenset({3, 28}), frozenset({17})) 0.6267806267806267
(frozenset({17, 28}), frozenset({3})) 0.6267806267806267
(frozenset({17, 3}), frozenset({28})) 0.632183908045977
(frozenset({11, 28}), frozenset({9})) 0.5422626788036411
(frozenset({9, 28}), frozenset({11})) 0.6836065573770492
(frozenset({9, 11}), frozenset({28})) 0.6556603773584906
(frozenset({9, 3}), frozenset({11})) 0.6947723440134908
(frozenset({3, 11}), frozenset({9})) 0.544973544973545
(frozenset({9, 11}), frozenset({3})) 0.6477987421383647
(frozenset({17, 3}), frozenset({8})) 0.6752873563218391
(frozenset({8, 3}), frozenset({17})) 0.6536856745479833
(frozenset({8, 17}), frozenset({3})) 0.6509695290858726
(frozenset({9, 11}), frozenset({8})) 0.6698113207547169

```

Fig.9 Rules of high support, high confidence in testdata2

(2) High support, low confidence (min_support=400, min_coffidence=0.1)

➤ testdata.csv

Time to find the frequent sets:

Apriori Algorithm: 1.43550705909729sec

FP-growth Algorithm: 1.2936785221099854sec

```

(frozenset({28}), 582)
(frozenset({14}), 485)
(frozenset({8}), 603)
(frozenset({11}), 625)
(frozenset({17}), 542)
(frozenset({9}), 492)
(frozenset({3}), 576)
(frozenset({29}), 478)
(frozenset({8, 11}), 415)
{frozenset({28}): 582, frozenset({14}): 485, frozenset({8}): 603, frozenset({11}): 625, frozenset({17}): 542, frozenset({9}): 4
92, frozenset({3}): 576, frozenset({29}): 478, frozenset({8, 11}): 415}

```

Fig.10 Frequent sets of high support, low confidence in testdata

```
(frozenset({11}), frozenset({8})) 0.664
(frozenset({8}), frozenset({11})) 0.6882255389718076
```

Fig.11 Rules of of High support, low confidence in testdata

➤ testdata2.csv

Time to find the frequent sets:

Apriori Algorithm: 2.847632646560669 sec

FP-growth Algorithm: 4.545107364654541 sec

```
33 # print ('Length of Rules',len(rules_dic))
34 # for key,value in rules_dic.items():
35 #     print (key,value)
36
```

Length of frequent sets 87

```
{frozenset({28}): 1175, frozenset({25}): 578, frozenset({10}): 506, frozenset({18}): 412, frozenset({23}): 752, frozenset({1
4}): 967, frozenset({8}): 1180, frozenset({13}): 484, frozenset({11}): 1238, frozenset({17}): 1125, frozenset({9}): 966, fro
set({26}): 622, frozenset({5}): 618, frozenset({12}): 509, frozenset({7}): 630, frozenset({3}): 1135, frozenset({29}): 954, fro
zenset({6}): 450, frozenset({19}): 461, frozenset({21}): 765, frozenset({0}): 598, frozenset({29, 14}): 507, frozenset({3, 2
1}): 462, frozenset({8, 7}): 408, frozenset({9, 21}): 401, frozenset({9, 14}): 497, frozenset({3, 11}): 756, frozenset({11, 1
4}): 622, frozenset({8, 23}): 464, frozenset({3, 23}): 469, frozenset({3, 14}): 589, frozenset({8, 3}): 719, frozenset({11, 2
3}): 495, frozenset({3, 28}): 702, frozenset({8, 28}): 761, frozenset({17, 29}): 587, frozenset({17, 7}): 404, frozenset({9, 2
9}): 511, frozenset({11, 29}): 657, frozenset({17, 9}): 562, frozenset({11, 5}): 407, frozenset({9, 11}): 636, frozenset({28, 2
1}): 478, frozenset({17, 14}): 576, frozenset({17, 28}): 702, frozenset({8, 29}): 624, frozenset({9, 28}): 610, frozenset({28,
23}): 482, frozenset({9, 3}): 593, frozenset({26, 11}): 408, frozenset({17, 3}): 696, frozenset({28, 14}): 598, frozenset({11,
7}): 420, frozenset({8, 11}): 778, frozenset({8, 9}): 625, frozenset({17, 23}): 456, frozenset({8, 14}): 607, frozenset({17, 2
1}): 468, frozenset({8, 21}): 474, frozenset({3, 29}): 583, frozenset({11, 21}): 506, frozenset({11, 28}): 769, frozenset({17,
11}): 744, frozenset({8, 0}): 402, frozenset({28, 29}): 617, frozenset({8, 17}): 722, frozenset({11, 17, 3}): 485, frozenset
({3, 11, 29}): 417, frozenset({17, 3, 28}): 440, frozenset({9, 11, 28}): 417, frozenset({11, 9, 3}): 412, frozenset({8, 17,
3}): 470, frozenset({8, 9, 11}): 426, frozenset({11, 28, 29}): 439, frozenset({8, 17, 11}): 500, frozenset({8, 11, 29}): 438, f
rozenset({17, 11, 28}): 476, frozenset({8, 9, 28}): 420, frozenset({17, 11, 29}): 423, frozenset({8, 11, 28}): 511, frozenset
({8, 3, 28}): 475, frozenset({8, 17, 29}): 407, frozenset({8, 11, 14}): 415, frozenset({8, 28, 29}): 419, frozenset({3, 8, 1
1}): 491, frozenset({8, 17, 28}): 477, frozenset({3, 11, 28}): 478}
```

Fig.12 Frequent sets of high support, low confidence in testdata2

```
36
```

Length of Rules 216

```
(frozenset({17, 3}), frozenset({11})) 0.6968390804597702
(frozenset({3}), frozenset({17, 11})) 0.42731277533059647
(frozenset({17}), frozenset({3, 11})) 0.4311111111111111
(frozenset({3, 11}), frozenset({17})) 0.6415343915343915
(frozenset({11}), frozenset({17, 3})) 0.39176990468497576
(frozenset({17, 11}), frozenset({3})) 0.6518817204301075
(frozenset({11, 29}), frozenset({3})) 0.634703196347032
(frozenset({29}), frozenset({11, 3})) 0.4371069182389937
(frozenset({11}), frozenset({3, 29})) 0.3368336025848142
(frozenset({3, 29}), frozenset({11})) 0.7152658662092625
(frozenset({3}), frozenset({11, 29})) 0.3674008810572687
(frozenset({11, 3}), frozenset({29})) 0.5515873015873016
(frozenset({3, 28}), frozenset({17})) 0.6267806267806267
(frozenset({28}), frozenset({17, 3})) 0.37446808510638296
(frozenset({3}), frozenset({17, 28})) 0.3876651982378855
(frozenset({17, 28}), frozenset({3})) 0.6267806267806267
(frozenset({17}), frozenset({3, 28})) 0.3911111111111113
(frozenset({17, 3}), frozenset({28})) 0.632183908045977
(frozenset({11, 28}), frozenset({9})) 0.5422626788036411
```

Fig.13 Rules of high support, low confidence in testdata2

(3) Low support, low confidence (min_support=100, min_coffidence=0.1)

➤ testdata.csv

Time to find the frequent sets:

Apriori Algorithm: 2.565826416015625sec

FP-growth Algorithm: 4.4978346824646sec


```

32 # print (len(rules_dic))
33 # for key,values in rules_dic.items():
34 #     print (key,values)
35
Length of frequent sets 552
(frozenset({28}): 582, frozenset({25}): 296, frozenset({10}): 247, frozenset({18}): 191, frozenset({23}): 369, frozenset({1
4}): 485, frozenset({27}): 104, frozenset({24}): 141, frozenset({8}): 603, frozenset({13}): 246, frozenset({22}): 181, frozens
et({11}): 625, frozenset({17}): 542, frozenset({9}): 492, frozenset({26}): 311, frozenset({5}): 325, frozenset({15}): 120, frozens
et({12}): 248, frozenset({4}): 195, frozenset({20}): 111, frozenset({7}): 302, frozenset({3}): 576, frozenset({29}): 478,
frozenset({6}): 231, frozenset({19}): 221, frozenset({21}): 384, frozenset({0}): 302, frozenset({3, 4}): 116, frozenset({25,
7}): 115, frozenset({9, 13}): 124, frozenset({0, 29}): 169, frozenset({11, 14}): 322, frozenset({8, 3}): 375, frozenset({3, 2
8}): 341, frozenset({25, 28}): 180, frozenset({9, 5}): 172, frozenset({26, 23}): 127, frozenset({3, 12}): 156, frozenset({8,
5}): 216, frozenset({28, 21}): 239, frozenset({28, 13}): 157, frozenset({19, 3}): 145, frozenset({8, 13}): 166, frozenset({25,
21}): 125, frozenset({9, 28}): 308, frozenset({9, 3}): 303, frozenset({14, 23}): 192, frozenset({28, 22}): 110, frozenset({19,
28}): 135, frozenset({25, 23}): 116, frozenset({11, 4}): 133, frozenset({10, 29}): 128, frozenset({8, 11}): 415, frozenset({9,
10}): 133, frozenset({18, 11}): 141, frozenset({0, 11}): 203, frozenset({8, 21}): 235, frozenset({3, 29}): 303, frozenset({17,
4}): 122, frozenset({8, 0}): 216, frozenset({28, 29}): 305, frozenset({12, 14}): 143, frozenset({8, 17}): 354, frozenset({10,
3}): 150, frozenset({28, 12}): 155, frozenset({29, 14}): 261, frozenset({3, 21}): 239, frozenset({18, 28}): 112, frozenset({1
9, 14}): 109, frozenset({14, 7}): 180, frozenset({9, 14}): 262, frozenset({3, 11}): 388, frozenset({3, 23}): 232, frozenset({1
9, 29}): 112, frozenset({11, 23}): 238, frozenset({0, 9}): 168, frozenset({3, 22}): 113, frozenset({17, 19}): 131, frozenset
({17, 7}): 185, frozenset({26, 21}): 132, frozenset({9, 29}): 268, frozenset({29, 6}): 121, frozenset({17, 9}): 276, frozenset
({11, 15}): 207, frozenset({9, 26}): 165, frozenset({12, 23}): 113, frozenset({3, 5}): 177, frozenset({17, 14}): 274, frozenset
({8, 12}): 155, frozenset({17, 13}): 130, frozenset({5, 7}): 116, frozenset({17, 3}): 337, frozenset({21, 13}): 103, frozenset
({8, 26}): 195, frozenset({28, 14}): 291, frozenset({12, 29}): 146, frozenset({0, 28}): 185, frozenset({11, 7}): 196, frozenset

```

Fig.14 Frequent sets of low support, low confidence in testdata

```

28 for item in freqSet_list:
29     produceRule(item,item,0.1)
30
31 # 印出結果
32 print ('Length of Rules',len(rules_dic))
33 for key,values in rules_dic.items():
34     print (key,values)
35
Length of Rules 3466
(frozenset({8, 9, 11, 28}), frozenset({3})) 0.6688311688311688
(frozenset({9, 11, 28}), frozenset({8, 3})) 0.4858490566037736
(frozenset({11, 28}), frozenset({8, 9, 3})) 0.2717678100263852
(frozenset({28}), frozenset({8, 9, 3, 11})) 0.17697594501718214
(frozenset({11}), frozenset({8, 9, 3, 28})) 0.1648
(frozenset({9, 28}), frozenset({8, 11, 3})) 0.3344155844155844
(frozenset({9}), frozenset({8, 11, 3, 28})) 0.20934959349593496
(frozenset({9, 11}), frozenset({8, 3, 28})) 0.31482439024390244
(frozenset({8, 11, 28}), frozenset({9, 3})) 0.39768339768339767
(frozenset({8, 28}), frozenset({11, 9, 3})) 0.2761394181876676
(frozenset({8}), frozenset({11, 9, 3, 28})) 0.17081269364842454
(frozenset({8, 11}), frozenset({9, 3, 28})) 0.24819277108433735
(frozenset({8, 9, 28}), frozenset({11, 3})) 0.4798697674418605
(frozenset({8, 9}), frozenset({11, 3, 28})) 0.3074626865671642
(frozenset({8, 9, 11}), frozenset({3, 28})) 0.4364406779661017
(frozenset({11, 9, 3, 28}), frozenset({8})) 0.7304964539007093
(frozenset({9, 3, 28}), frozenset({8, 11})) 0.533678756476684
(frozenset({3, 28}), frozenset({8, 9, 11})) 0.3078527850237537

```

Fig.15 Rules of low support, low confidence in testdata

➤ testdata2.csv

Time to find the frequent sets:

Apriori Algorithm: 18.514791011810303 sec

FP-growth Algorithm: 305.7357885837555sec

```

29 for item in freqSet_list:
30     produceRule(item,item,0.1)
31
32 # 印出結果
33 print ('Length of Rules',len(rules_dic))
34 # for key,values in rules_dic.items():
35 #     print (key,values)
36
Length of frequent sets 2451
(frozenset({28}): 1175, frozenset({25}): 578, frozenset({10}): 506, frozenset({18}): 412, frozenset({23}): 752, frozenset({1
4}): 967, frozenset({27}): 199, frozenset({24}): 301, frozenset({8}): 1180, frozenset({13}): 484, frozenset({16}): 186, frozens
et({22}): 358, frozenset({11}): 1238, frozenset({17}): 1125, frozenset({9}): 966, frozenset({26}): 622, frozenset({5}): 618,
frozenset({15}): 231, frozenset({12}): 509, frozenset({4}): 390, frozenset({20}): 226, frozenset({7}): 630, frozenset({3}): 11
35, frozenset({29}): 954, frozenset({6}): 450, frozenset({19}): 461, frozenset({21}): 765, frozenset({0}): 598, frozenset({25,
4}): 117, frozenset({10, 13}): 114, frozenset({9, 22}): 198, frozenset({3, 4}): 234, frozenset({16, 17}): 126, frozenset({25,
7}): 219, frozenset({9, 13}): 244, frozenset({0, 29}): 318, frozenset({11, 14}): 622, frozenset({0, 18}): 130, frozenset({3, 2
7}): 118, frozenset({8, 3}): 719, frozenset({21, 22}): 146, frozenset({10, 12}): 137, frozenset({18, 5}): 115, frozenset({18,
26}): 154, frozenset({13, 22}): 100, frozenset({3, 28}): 702, frozenset({19, 5}): 143, frozenset({25, 28}): 347, frozenset({2
5, 22}): 108, frozenset({4, 5}): 130, frozenset({28, 20}): 146, frozenset({9, 5}): 321, frozenset({26, 23}): 249, frozenset
({3, 12}): 320, frozenset({8, 5}): 394, frozenset({28, 21}): 478, frozenset({28, 13}): 308, frozenset({17, 22}): 208, frozenset
({19, 3}): 293, frozenset({8, 13}): 309, frozenset({25, 21}): 237, frozenset({9, 28}): 610, frozenset({9, 3}): 593, frozenset
({24, 14}): 166, frozenset({14, 23}): 387, frozenset({28, 22}): 229, frozenset({10, 18}): 135, frozenset({19, 28}): 287, frozenset
({25, 23}): 232, frozenset({11, 4}): 259, frozenset({10, 19}): 122, frozenset({14, 22}): 184, frozenset({10, 29}): 251, frozenset
({6, 23}): 196, frozenset({24, 29}): 157, frozenset({20, 29}): 121, frozenset({8, 11}): 778, frozenset({9, 10}): 261, frozenset
({18, 11}): 284, frozenset({8, 27}): 124, frozenset({9, 20}): 119, frozenset({0, 11}): 397, frozenset({18, 21}): 474, frozenset
({3, 29}): 583, frozenset({18, 7}): 158, frozenset({17, 4}): 246, frozenset({25, 10}): 161, frozenset({25, 12}): 158, frozenset
({8, 0}): 402, frozenset({3, 15}): 141, frozenset({28, 20}): 617, frozenset({12, 14}): 274, frozenset({8, 17}): 722,

```

Fig.16 Frequent sets of low support, low confidence in testdata2

```

31
32 # 印出結果
33 print ('Length of Rules',len(rules_dic))
34 for key,values in rules_dic.items():
35     print (key,values)
36
Length of Rules 28927
(frozenset({8, 11, 23, 28, 29}), frozenset({17})) 0.7112676056338029
(frozenset({11, 28, 29, 23}), frozenset({8, 17})) 0.5824875621890548
(frozenset({28, 29, 23}), frozenset({8, 17, 11})) 0.36727272727272725
(frozenset({29, 23}), frozenset({8, 17, 11, 28})) 0.25569620253164554
(frozenset({23}), frozenset({8, 11, 17, 28, 29})) 0.13430851063829788
(frozenset({29}), frozenset({8, 11, 17, 23, 28})) 0.10587002096436059
(frozenset({28, 23}), frozenset({8, 17, 11, 29})) 0.2095435684647303
(frozenset({28, 29}), frozenset({8, 17, 11, 23})) 0.16369529983792544
(frozenset({11, 29, 23}), frozenset({8, 17, 28})) 0.35815602836879434
(frozenset({11, 23}), frozenset({8, 17, 28, 29})) 0.20486815415821502
(frozenset({11, 29}), frozenset({8, 17, 28, 23})) 0.1537290715372907
(frozenset({11, 28, 23}), frozenset({8, 17, 29})) 0.3098159509202454
(frozenset({11, 28}), frozenset({8, 17, 29, 23})) 0.13133940182054615
(frozenset({11, 28, 29}), frozenset({8, 17, 23})) 0.23006833712984054
(frozenset({8, 28, 29, 23}), frozenset({17, 11})) 0.531578947368421
(frozenset({8, 29, 23}), frozenset({17, 11, 28})) 0.38257575757575757
(frozenset({8, 23}), frozenset({17, 11, 28, 29})) 0.21767241379310345
(frozenset({8, 29}), frozenset({17, 11, 28, 23})) 0.16185897435897437
(frozenset({8, 28, 23}), frozenset({17, 11, 29})) 0.3166144200626959

```

Fig.17 Rules of low support, low confidence in testdata2

(4) Low support, high confidence (min_support=100, min_confidence=0.5)

➤ testdata.csv

Time to find the frequent sets:

Apriori Algorithm: 2.565826416015625sec

FP-growth Algorithm: 4.4978346824646sec

```

34 #
35     print (key,values)
36
Length of frequent sets 552
(frozenset({28}): 582, frozenset({25}): 296, frozenset({10}): 247, frozenset({18}): 191, frozenset({23}): 369, frozenset({1
4}): 485, frozenset({27}): 104, frozenset({24}): 141, frozenset({8}): 603, frozenset({13}): 246, frozenset({22}): 181, frozens
et({11}): 625, frozenset({17}): 542, frozenset({9}): 492, frozenset({26}): 311, frozenset({5}): 325, frozenset({15}): 120, fro
zenset({12}): 248, frozenset({4}): 195, frozenset({20}): 111, frozenset({7}): 302, frozenset({3}): 576, frozenset({29}): 478,
frozenset({6}): 231, frozenset({19}): 221, frozenset({21}): 384, frozenset({0}): 302, frozenset({3, 4}): 116, frozenset({25,
7}): 115, frozenset({9, 13}): 124, frozenset({0, 29}): 169, frozenset({11, 14}): 322, frozenset({8, 3}): 375, frozenset({3, 2
8}): 341, frozenset({25, 28}): 180, frozenset({9, 5}): 172, frozenset({26, 23}): 127, frozenset({3, 12}): 156, frozenset({8,
5}): 216, frozenset({28, 21}): 239, frozenset({28, 13}): 157, frozenset({19, 3}): 145, frozenset({8, 13}): 166, frozenset({25,
21}): 125, frozenset({9, 28}): 308, frozenset({9, 3}): 303, frozenset({14, 23}): 192, frozenset({28, 22}): 110, frozenset({19,
28}): 135, frozenset({25, 23}): 116, frozenset({11, 4}): 133, frozenset({10, 29}): 128, frozenset({8, 11}): 415, frozenset({9,
10}): 133, frozenset({18, 11}): 141, frozenset({0, 11}): 203, frozenset({8, 21}): 235, frozenset({3, 29}): 303, frozenset({17,
4}): 122, frozenset({8, 0}): 216, frozenset({28, 29}): 305, frozenset({12, 14}): 143, frozenset({8, 17}): 354, frozenset({10,
3}): 150, frozenset({28, 12}): 155, frozenset({29, 14}): 261, frozenset({3, 21}): 239, frozenset({18, 28}): 112, frozenset({1
9, 14}): 109, frozenset({14, 7}): 180, frozenset({9, 14}): 262, frozenset({3, 11}): 388, frozenset({3, 23}): 232, frozenset({1
9, 29}): 112, frozenset({11, 23}): 238, frozenset({0, 9}): 168, frozenset({3, 22}): 113, frozenset({17, 19}): 131, frozenset
({17, 7}): 185, frozenset({26, 21}): 132, frozenset({9, 29}): 268, frozenset({29, 6}): 121, frozenset({17, 9}): 276, frozenset
({11, 5}): 207, frozenset({9, 26}): 165, frozenset({12, 23}): 113, frozenset({3, 5}): 177, frozenset({17, 14}): 274, frozenset
({8, 12}): 155, frozenset({17, 13}): 130, frozenset({5, 7}): 116, frozenset({17, 3}): 337, frozenset({21, 13}): 103, frozenset
({8, 26}): 195, frozenset({28, 14}): 291, frozenset({12, 29}): 146, frozenset({0, 28}): 185, frozenset({11, 7}): 196, frozenset

```

Fig.10 Frequent sets of low support, high confidence in testdata

```

34 print ('Length of Rules',len(rules_dic))
35 for key,values in rules_dic.items():
36     print (key,values)
37
Length of Rules 1194
(frozenset({8, 9, 11, 28}), frozenset({3})) 0.6688311688311688
(frozenset({11, 9, 3, 28}), frozenset({8})) 0.7304964539007093
(frozenset({9, 3, 28}), frozenset({8, 11})) 0.533678756476684
(frozenset({8, 11, 3, 28}), frozenset({9})) 0.6242424242424243
(frozenset({8, 9, 3, 28}), frozenset({11})) 0.7183448275862069
(frozenset({8, 9, 3, 11}), frozenset({28})) 0.6645161290322581
(frozenset({8, 9, 11, 29}), frozenset({3})) 0.6754966887417219
(frozenset({9, 11, 29}), frozenset({8, 3})) 0.5125628148703518
(frozenset({8, 9, 29}), frozenset({11, 3})) 0.5125628148703518
(frozenset({11, 9, 3, 29}), frozenset({8})) 0.7338129496402878
(frozenset({9, 3, 29}), frozenset({8, 11})) 0.576271186440678
(frozenset({8, 11, 3, 29}), frozenset({9})) 0.6375
(frozenset({8, 9, 3, 29}), frozenset({11})) 0.7611940298507462
(frozenset({8, 9, 3, 11}), frozenset({29})) 0.6580645161290323
(frozenset({8, 11, 28, 29}), frozenset({17})) 0.6753246753246753
(frozenset({17, 11, 28, 29}), frozenset({8})) 0.7482014388489209
(frozenset({17, 28, 29}), frozenset({8, 11})) 0.5502645502645502
(frozenset({17, 11, 29}), frozenset({8, 28})) 0.5048543689320388
(frozenset({8, 17, 28, 29}), frozenset({11})) 0.7536231884057971

```

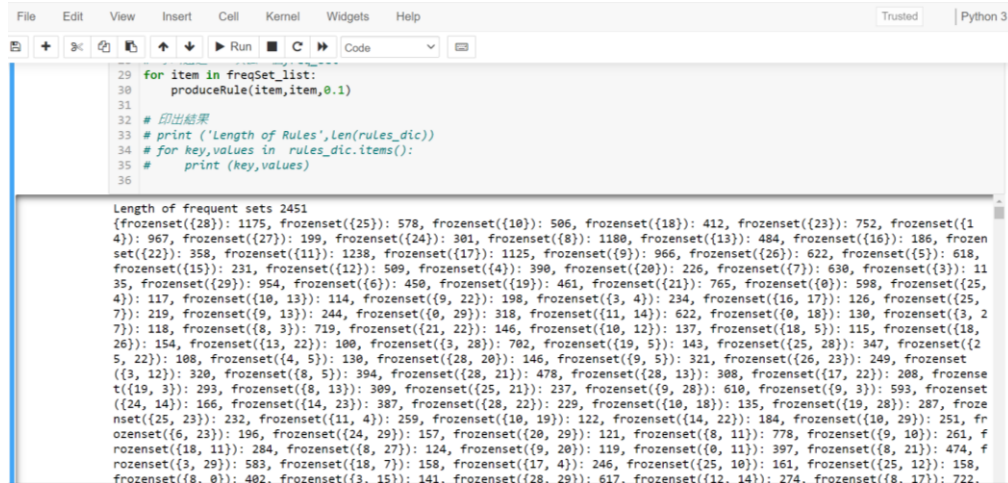
Fig.10 Rules of low support, high confidence in testdata

➤ testdata2.csv

Time to find the frequent sets:

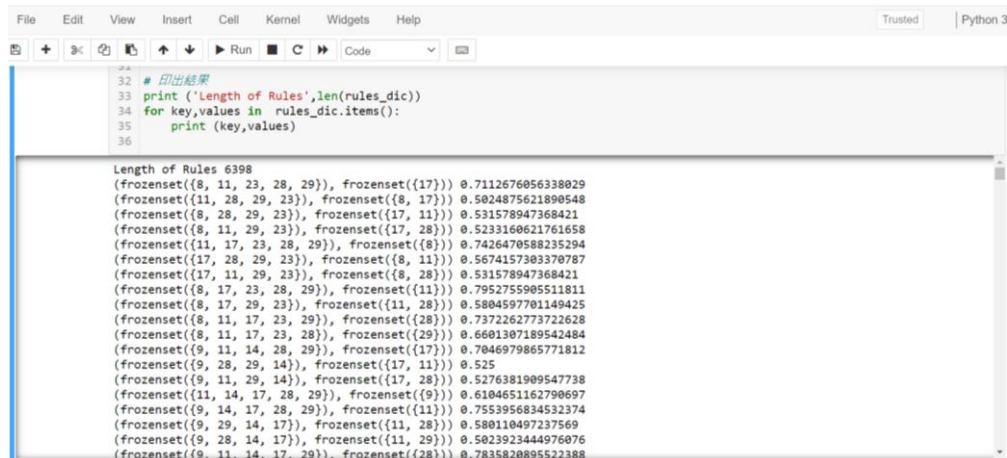
Apriori Algorithm: 18.514791011810303 sec

FP-growth Algorithm: 305.7357885837555sec



```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
+ - * < > Run C Code
29 for item in freqSet_list:
30     produceRule(item,item,0.1)
31
32 # 输出结果
33 # print ('Length of Rules',len(rules_dic))
34 # for key,values in rules_dic.items():
35 #     print (key,values)
36
Length of frequent sets 2451
{frozenset({28}): 1175, frozenset({25}): 578, frozenset({10}): 506, frozenset({18}): 412, frozenset({23}): 752, frozenset({1
4}): 967, frozenset({17}): 199, frozenset({24}): 301, frozenset({8}): 1180, frozenset({13}): 484, frozenset({16}): 186, frozen
set({22}): 358, frozenset({11}): 1238, frozenset({17}): 1125, frozenset({9}): 966, frozenset({26}): 622, frozenset({5}): 618,
frozenset({15}): 231, frozenset({12}): 509, frozenset({4}): 390, frozenset({20}): 226, frozenset({7}): 630, frozenset({3}): 11
35, frozenset({29}): 954, frozenset({6}): 450, frozenset({19}): 461, frozenset({21}): 765, frozenset({0}): 598, frozenset({25,
4}): 117, frozenset({10, 13}): 114, frozenset({9, 22}): 198, frozenset({3, 4}): 234, frozenset({16, 17}): 126, frozenset({25,
7}): 219, frozenset({9, 13}): 244, frozenset({0, 29}): 318, frozenset({11, 14}): 622, frozenset({0, 18}): 130, frozenset({3, 2
7}): 118, frozenset({8, 3}): 719, frozenset({21, 22}): 146, frozenset({10, 12}): 137, frozenset({18, 5}): 115, frozenset({18,
26}): 154, frozenset({13, 22}): 100, frozenset({3, 28}): 702, frozenset({19, 5}): 143, frozenset({25, 28}): 347, frozenset({2
5, 22}): 108, frozenset({4, 5}): 130, frozenset({28, 20}): 146, frozenset({9, 5}): 321, frozenset({26, 23}): 249, frozenset
({3, 12}): 320, frozenset({8, 5}): 394, frozenset({28, 21}): 478, frozenset({28, 13}): 308, frozenset({17, 22}): 208, frozene
t({19, 3}): 293, frozenset({8, 13}): 309, frozenset({25, 21}): 237, frozenset({9, 28}): 610, frozenset({9, 3}): 593, frozenset
({24, 14}): 166, frozenset({14, 23}): 387, frozenset({28, 22}): 229, frozenset({10, 18}): 135, frozenset({19, 28}): 287, froze
nset({25, 23}): 232, frozenset({11, 4}): 259, frozenset({10, 19}): 122, frozenset({14, 22}): 184, frozenset({10, 29}): 251, fr
ozenset({6, 23}): 196, frozenset({24, 29}): 157, frozenset({20, 29}): 121, frozenset({8, 11}): 778, frozenset({9, 10}): 261, f
rozenset({18, 11}): 284, frozenset({8, 27}): 124, frozenset({9, 20}): 119, frozenset({0, 11}): 397, frozenset({8, 21}): 474, f
rozenset({3, 29}): 583, frozenset({18, 7}): 158, frozenset({17, 4}): 246, frozenset({25, 10}): 161, frozenset({25, 12}): 158,
frozenset({8, 0}): 402, frozenset({3, 15}): 141, frozenset({28, 29}): 617, frozenset({12, 14}): 274, frozenset({8, 17}): 722,
```

Fig.11 Frequent sets of low support, high confidence in testdata2



```
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
+ - * < > Run C Code
32 # 输出结果
33 # print ('Length of Rules',len(rules_dic))
34 # for key,values in rules_dic.items():
35 #     print (key,values)
36
Length of Rules 6398
(frozenset({8, 11, 23, 28, 29}), frozenset({17})) 0.7112676056338029
(frozenset({11, 28, 29, 23}), frozenset({8, 17})) 0.5024875621890548
(frozenset({8, 28, 29, 23}), frozenset({17, 11})) 0.531578947368421
(frozenset({8, 11, 29, 23}), frozenset({17, 28})) 0.5233160621761658
(frozenset({11, 17, 23, 28, 29}), frozenset({8})) 0.7426470588235294
(frozenset({17, 28, 29, 23}), frozenset({8, 11})) 0.5674157303370787
(frozenset({17, 11, 29, 23}), frozenset({8, 28})) 0.531578947368421
(frozenset({8, 17, 23, 28, 29}), frozenset({11})) 0.7952755905511811
(frozenset({8, 17, 29, 23}), frozenset({11, 28})) 0.5804597701149425
(frozenset({8, 11, 17, 23, 29}), frozenset({28})) 0.737226273722628
(frozenset({8, 11, 17, 23, 28}), frozenset({29})) 0.6601307189542484
(frozenset({9, 11, 14, 28, 29}), frozenset({17})) 0.7046979865771812
(frozenset({9, 28, 29, 14}), frozenset({17, 11})) 0.525
(frozenset({9, 11, 29, 14}), frozenset({17, 28})) 0.5276381909547738
(frozenset({11, 14, 17, 28, 29}), frozenset({9})) 0.6104651162790697
(frozenset({9, 14, 17, 28, 29}), frozenset({11})) 0.7553956834532374
(frozenset({9, 29, 14, 17}), frozenset({11, 28})) 0.580110497237569
(frozenset({9, 28, 14, 17}), frozenset({11, 29})) 0.5023923444976076
(frozenset({9, 11, 14, 17, 29}), frozenset({28})) 0.7835820895522388
```

Fig.12 Rules of low support, high confidence in testdata2

4. Experience

I'm a beginner of Python, I used to program in C & C++. I spend lots of time to learn how to program in Python style. After doing the homework, I got lots of strategy of programming in Python style. Also, I understand association analysis more.