Data_mining HW1

NE6101034 AI碩一 柳譯筑

About Dataset:

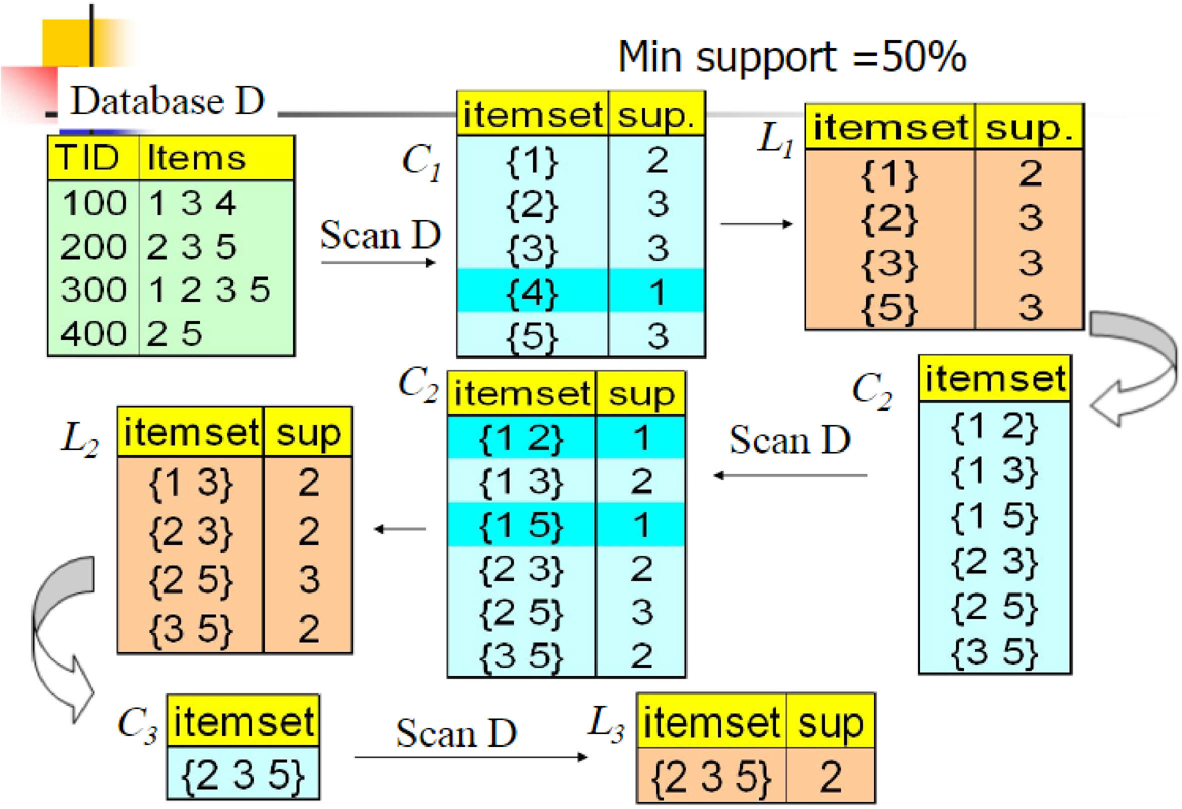
https://www.kaggle.com/irfanasrullah/groceries/activity

我用的dataset 是從kaggle上找的groceries dataset，這個dataset包含了9835 transaction跟169 unique items

程式架構：

```
.
|- Apriori.py
|       |- import_data()
|       |- oneitemlist(TDB)
|       |- get_L(itemsetlst,TDB,k, min_sup)
|       |- get_C2(L,length)
|       |- powerset(s)
|       |- rule_gen(global_L,global_freq,minConf)
|
|- FP_Growth.py
|       |- class treeNode
|       |- updateHeader(node, targetNode)
|       |- updateFPtree(items, inTree, headerTable, count)
|       |- createFPtree(dataSet, minSup)
|       |- createInitSet(dataSet)
|
|- mining.py
|       |- ascending(leafNode, prefix)
|       |- findPrefix(base, hdTable)
|       |- mineFPtree(FPtree, hdTable, minSup, freqItems)
|
|- draw_time.py
|- groceries _ groceries.csv
```

- **Apriori.py**
    - 負責 apriori 實作



Min support =50%

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

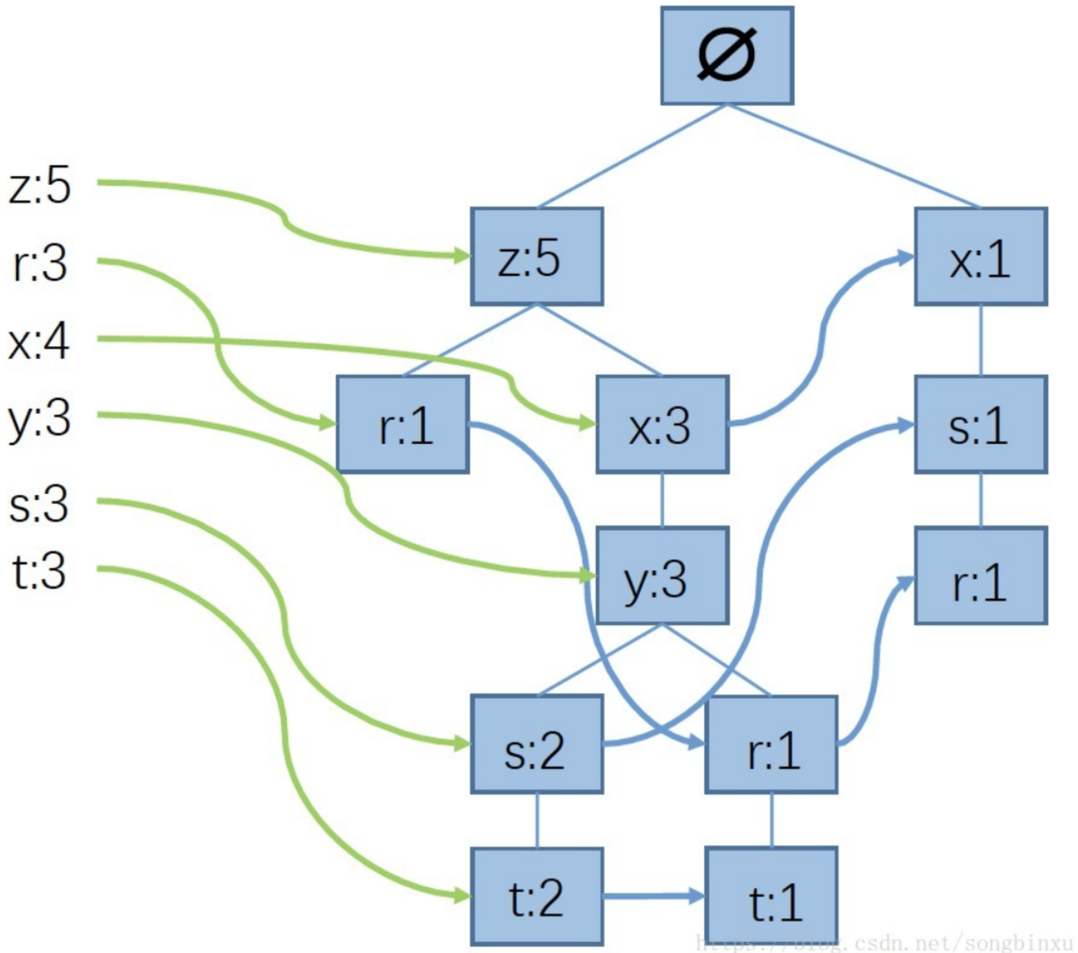
L_3

itemset	sup
{2 3 5}	2

- functions：
  - import_data() 負責先把item encode成 int型態，計算support 後接著將 groceries _ groceries.csv的資料存一個dictionary, dictionary形式：{ Item_number : support }
  - oneitemlist(TDB) : 找到one item set (C1)
  - get_C(L, length): 負責做combination , brute force 列出所有可能
  - get_L(itemsetlst, TDB, k, min_sup): 把沒有過min support 的 prune 掉
  - powerset(s)
  - rule_gen(global_L, global_freq, minConf)

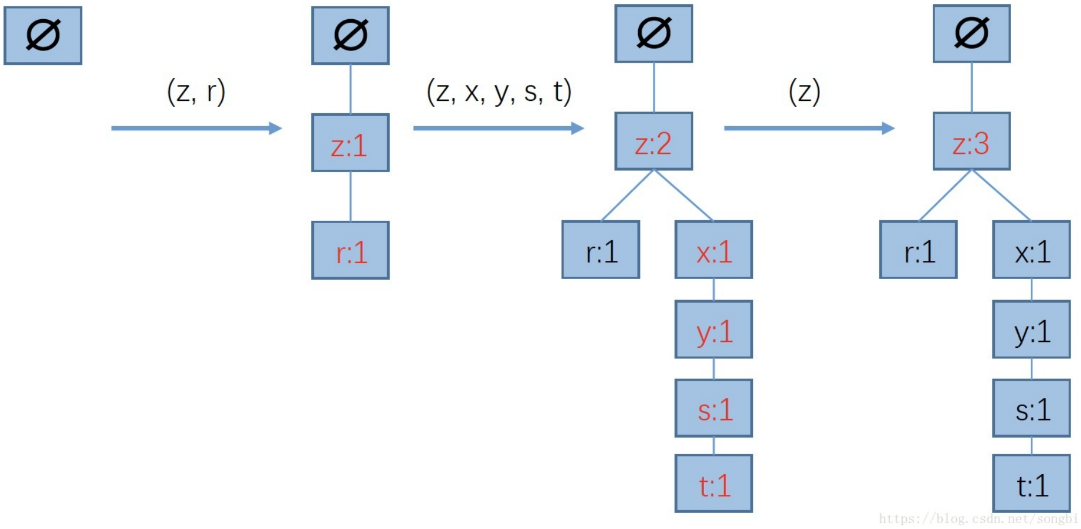
- **FP_Growth.py**
  - 負責 FP_Growth實作
  - function 功能：

- class treeNode
    - name
    - acc_cnt: 累積次數
    - nodeLink: 橫向的link
    - parent
    - children
- updateHeader(node, targetNode)
    - 負責update Header table，做橫向的更新



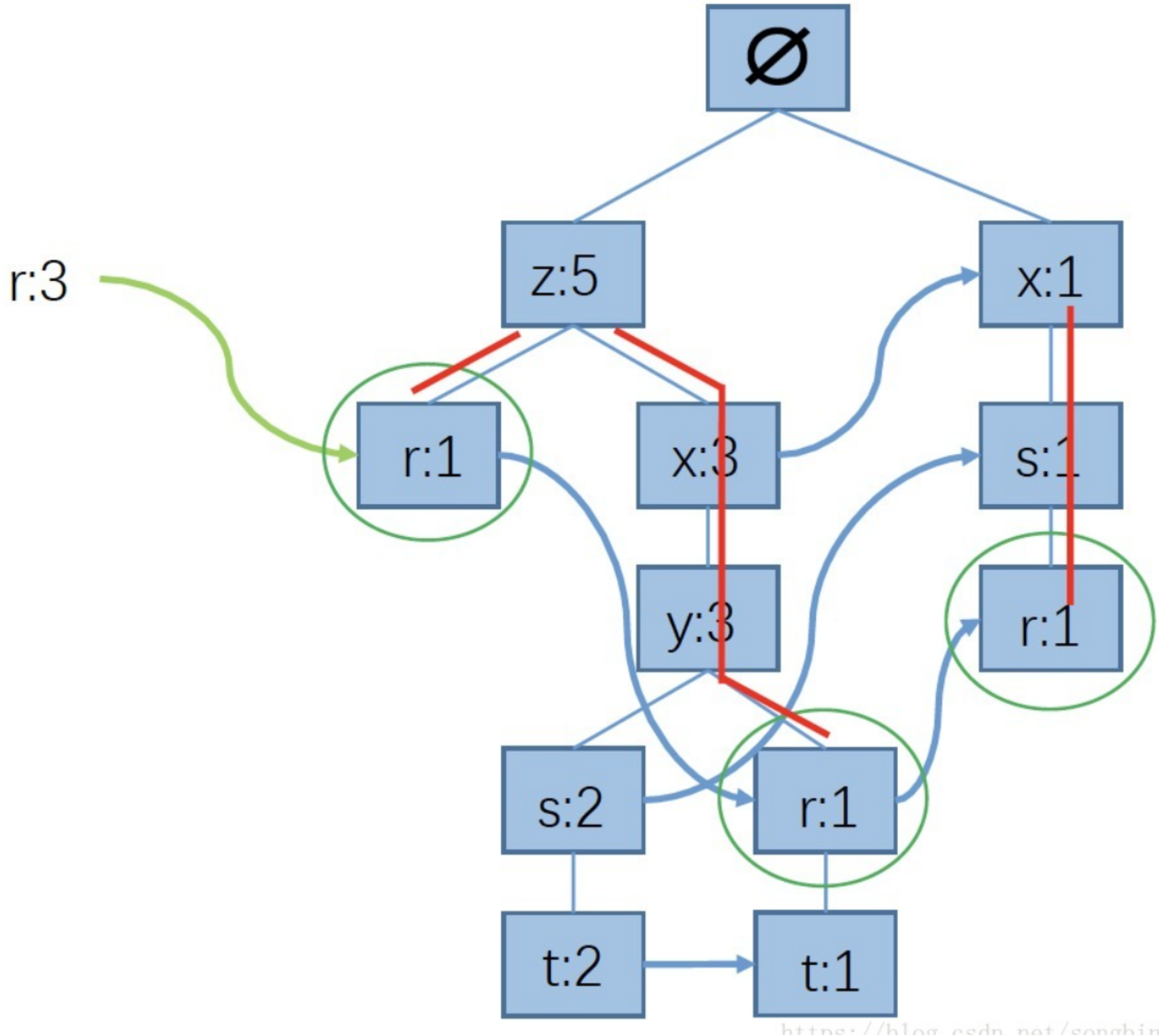
- updateHeader(node, targetNode)
    - 遍歷TDB，把小於minimum support 的濾掉，每條itemset 做一次 update

        - 在一開始跟後來的mining都使用這個function



    - updateFPtree(items, curTree, headerTable, acc_cnt)
        - 每次新的itemset進來，負責update FP-tree，做直向的更新
    - createInitSet(dataSet)

- **mining.py**

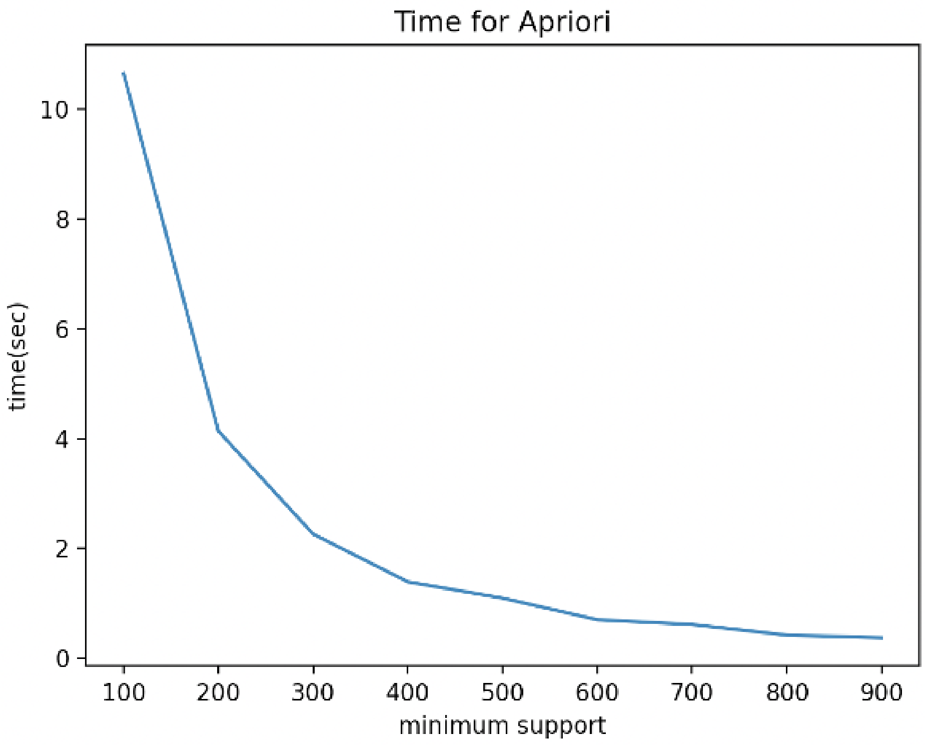


- ascending(leafNode, prefix)
  - 儲存紅色path
- find_prefix(base, hdTable)
  - 以這張tree為例， find_prefix(‘r‘, hdTable)

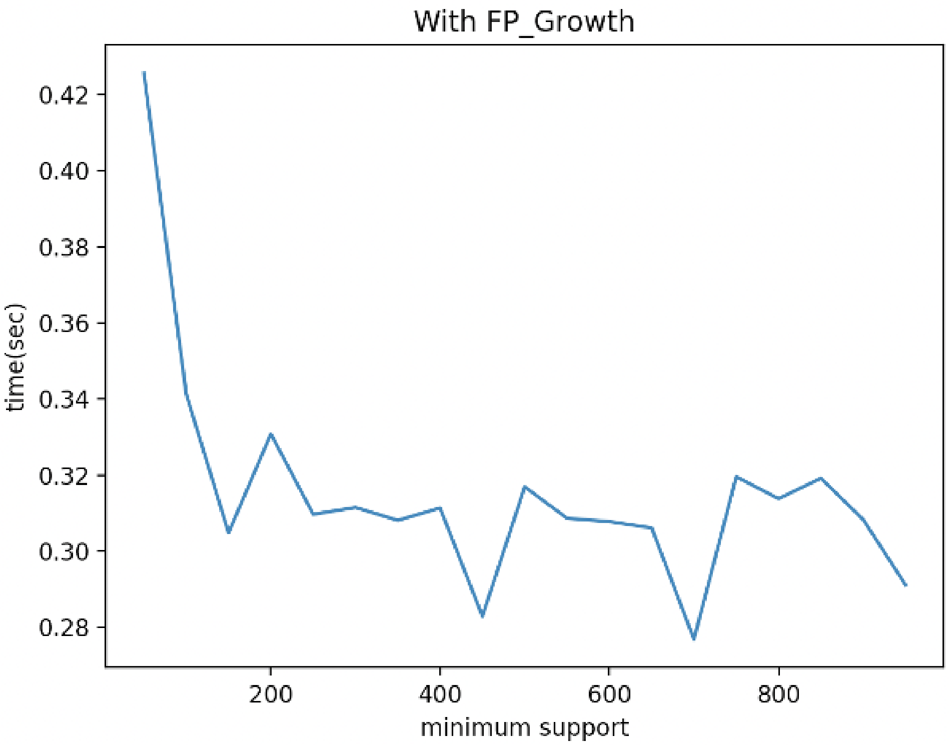
    → 會得到 {x,s}:1, {z,x,y}:1, {z}:1
- mineFPtree(FPtree, hdTable, minSup, freqItems)
  - rerurn 高於 minSup 的 freqItemsets

實驗結果

1. **Apriori with brute-force :**



2. FP Growth



```
Total Time:  0.6789960861206055  sec
minsup 600
minconf 0.1
num:  2

Total Time:  0.685157299041748  sec
minsup 600
minconf 0.2
num:  2

Total Time:  0.6842069625854492  sec
minsup 600
minconf 0.3
num:  1

Total Time:  0.6856534481048584  sec
minsup 600
minconf 0.4
num:  0

Total Time:  1.0889663696289062  sec
minsup 500
minconf 0.1
num:  6
```

```
Total Time:  1.0724871158599854  sec
minsup 500
minconf 0.5
num:  0

Total Time:  1.3605141639709473  sec
minsup 400
minconf 0.1
num:  16

Total Time:  1.3487548828125  sec
minsup 400
minconf 0.2
num:  14

Total Time:  1.360137939453125  sec
minsup 400
minconf 0.3
num:  7

Total Time:  1.355186939239502  sec
minsup 400
minconf 0.4
num:  4

Total Time:  1.3585259914398193  sec
minsup 400
minconf 0.5
num:  0

Total Time:  2.2040810585021973  sec
minsup 300
minconf 0.1
num:  36

Total Time:  2.2045092582702637  sec
minsup 300
minconf 0.2
num:  24

Total Time:  2.204713821411133  sec
minsup 300
minconf 0.3
num:  13

Total Time:  2.1939971446990967  sec
minsup 300
minconf 0.4
num:  5

Total Time:  2.195888042449951  sec
minsup 300
minconf 0.5
```

```
num:  0

Total Time:  4.035916805267334  sec
minsup 200
minconf 0.1
num:  122

Total Time:  4.0237650871276855  sec
minsup 200
minconf 0.2
num:  72

Total Time:  4.022386789321899  sec
minsup 200
minconf 0.3
num:  36

Total Time:  4.027678966522217  sec
minsup 200
minconf 0.4
num:  15

Total Time:  4.044141054153442  sec
minsup 200
minconf 0.5
num:  1


Total Time:  10.744132041931152  sec
minsup 100
minconf 0.5
num:  13

Total Time:  10.796161890029907  sec
minsup 100
minconf 0.3
num:  122

minsup 100
minconf 0.1
num:  448

Total Time:  12.675949096679688  sec
minsup 90
minconf 0.1
num:  568

Total Time:  12.767038822174072  sec
minsup 90
minconf 0.2
num:  286

Total Time:  12.652554035186768  sec
minsup 90
```

```
minconf 0.3
num:  160

Total Time:  12.797768115997314  sec
minsup 90
minconf 0.4
num:  78

Total Time:  12.692430019378662  sec
minsup 90
minconf 0.5
num:  23

Total Time:  17.906562089920044  sec
minsup 80
minconf 0.1
num:  720

Total Time:  17.481483936309814  sec
minsup 80
minconf 0.2
num:  359

Total Time:  17.289433002471924  sec
minsup 80
minconf 0.3
num:  200

Total Time:  17.336360216140747  sec
minsup 80
minconf 0.4
num:  99

Total Time:  17.449546813964844  sec
minsup 80
minconf 0.5
num:  27

Total Time:  25.939293146133423  sec
minsup 70
minconf 0.1
num:  915

Total Time:  25.383134126663208  sec
minsup 70
minconf 0.2
num:  459

Total Time:  25.86386013031006  sec
minsup 70
minconf 0.3
num:  253

Total Time:  26.434595823287964  sec
```

```
minsup 70
minconf 0.4
num:  127

Total Time:  26.35964608192444  sec
minsup 70
minconf 0.5
num:  39

Total Time:  251.12425708770752  sec
minsup 50
minconf 0.1
num:  1752

Total Time:  241.81170415878296  sec
minsup 50
minconf 0.2
num:  891

Total Time:  246.26084399223328  sec
minsup 50
minconf 0.4
num:  267

Total Time:  247.21111702919006  sec
minsup 50
minconf 0.5
num:  113
```

分析

1. 時間：

   可以從執行時間看到 Apriori 跟 FP Growth 的效能差異，FP Growth之所以厲害是因為他不需要去產生candidate，用 brute force 做 combination，for example 以我的dataset來產生C2，總共169種unique item ， C169取2 = 14196筆資料，在minimum support 設定在50的時候需要花到4分鐘才能做完所有rule 的 implies，minimum support 設70的時候26秒，隨著minSup越來越strict，筆數也隨之減少使得計算時間變快

2. 關於rules:

   根據我的觀察，two items → one item的confidence會大於 one item → two items，由 A → B = P(B)/P(A∩B)來推斷，是因為 one item 的 frequency 會高於 two item，分子比較大而分母一樣就會得到這個結果

   High support, high confidence ?

根據我的觀察，frequency 跟 confidence兩者皆高的data通常都是很相關的物品，比如說你買了牙刷就會順便買牙膏

High support, low confidence ?

根據觀察，我覺得這樣的資料通常都是one item互相關聯(freq較高)

Low support, high confidence ?

這種資料應該就相對較有意義，但是由groceries dataset 我還沒有看到比較明顯的特性

Low support, low confidence ?

這種資料就比較沒有什麼意義，只要稍微有關聯的就會被加入rules