

# 資料探勘專題作業四

使用 Python 進行交易資料關聯規則分析

指導教授:

許中川 教授

成員:

M11123026 林宥昇

M11123047 劉穎謙

M11123055 蕭旭朝

日期:

2023 年 01 月 03 號

# 摘要

本研究使用 Python 進行交易資料關聯規則分析，使用交易資料集，先行剔除數量為零或是負值的資料再使用 Apriori 和 FP-growth 演算法設定不同的支持度以及信心度，記錄下規則的數量以及執行的時間，此外還進行了剔除冗餘規則以減少規則數量。

關鍵字：交易資料集、關聯規則、Apriori、支持度、信心度、FP-growth

## 一、緒論

### 1.1 動機

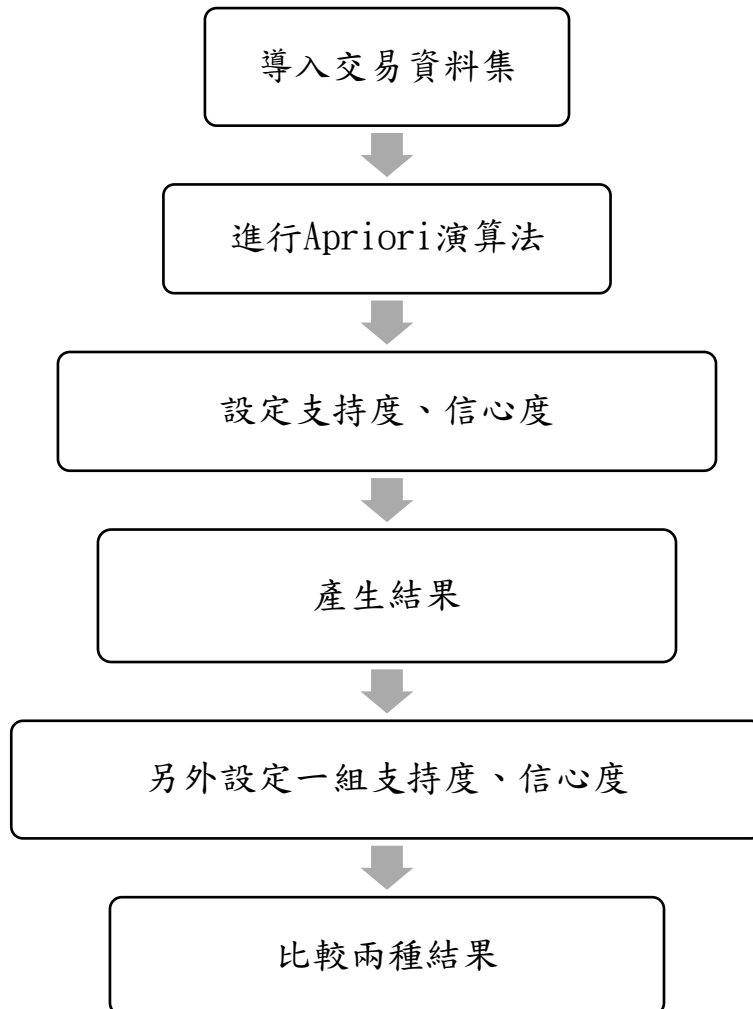
關聯分析用於發現用戶購買不同的商品之間存在關聯和相關聯繫本研究使用 Apriori 和 FP-growth 進行關聯資料分析通過大量銷售數據找到經常在一起購買的商品組合，以了解用戶的購買行為。

### 1.2 目的

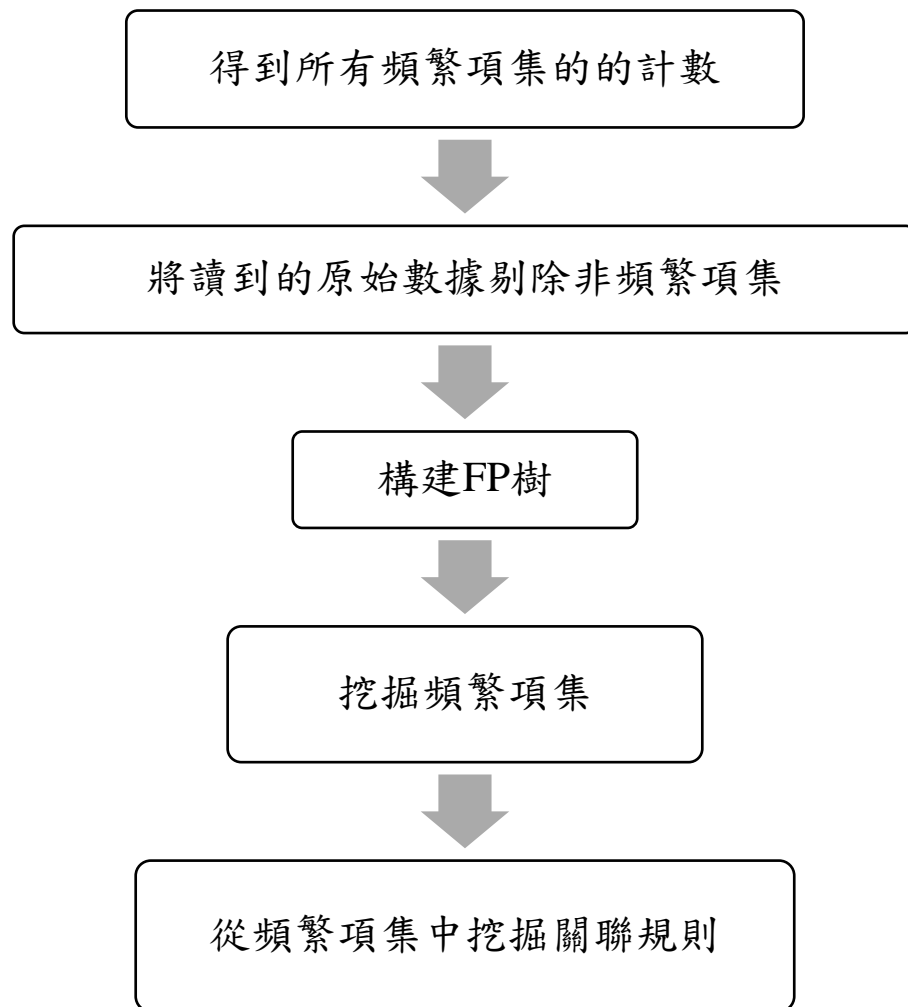
本研究使用 Apriori 和 FP-growth 進行關聯資料分析通過大量銷售數據找到經常在一起購買的商品組合，可以了解用戶的購買行為，根據銷售的商品推薦關聯商品從而給出購買建議，尋找銷售新的增長點。並且比較兩者所需時間的多寡

## 二、方法

### 2.1 Apriori 演算法



## 2.2 FP-growth 演算法



### 三、實驗

#### 3.1 資料集

交易資料集

ITEM_ID	物品編號
ITEM_NO	編號
PRODUCT_TYPE	產品類別
CUST_ID	客戶編號
TRX_DATE	日期
INVOICE_NO	發票號碼
QUANTITY	數量

#### 3.2 實驗設計

首先針對資料集進行導入，接著使用 Apriori 演算法進行支持度與信心度數值的設定，觀察其結果；進行第二次 Apriori 演算法，再次調整支持度與信心度數值調整，觀察其結果與第一次進行 Apriori 演算法所得出結果上之不同。進行 FP-growth 演算法，得出奇演算法結果後將 Apriori 演算法與 FP-growth 演算法進行比較，從而得出結論。

### 3.3 實驗結果

#### Apriori 演算法

##### 第一次

```
['Y', 'U']  
Rule: Y → U  
Support: 0.2857142857142857  
Lift: 2.3333333333333335  
=====
```

```
['U', 'C', 'D']  
Rule: U → C  
Support: 0.2857142857142857  
Lift: 2.3333333333333335  
=====
```

```
['C', 'O', 'E']  
Rule: C → O  
Support: 0.2857142857142857  
Lift: 2.3333333333333335  
=====
```

```
['T', 'C', 'U']  
Rule: T → C  
Support: 0.2857142857142857  
Lift: 2.3333333333333335
```

min support	0.2
min confidence	0.2

##### 第二次 (進行調整支持度、信心度)

```
['Q', 'A']  
Rule: Q → A  
Support: 0.14285714285714285  
Lift: 3.5  
=====
```

```
['X', 'A']  
Rule: X → A  
Support: 0.14285714285714285  
Lift: 3.5  
=====
```

```
['P', 'C']  
Rule: P → C  
Support: 0.14285714285714285  
Lift: 2.3333333333333335  
=====
```

```
['C', 'S']  
Rule: C → S  
Support: 0.14285714285714285  
Lift: 2.3333333333333335
```

min support	0.1
min confidence	0.1

## FP-Growth 演算法

```
- 6
  T 5
  E 4
    I 2
      D 1
      N 1
      O 1
      D 2
      C 1
      O 1
      U 1
    I 1
    D 1
    C 1
    U 1
  E 1
  I 1
  N 1
  C 1
  O 1
T 1
I 1
N 1
U 1
conditional tree for: {'N'}
Null Set 1
I 3
conditional tree for: {'O'}
Null Set 1
- 3
  E 3
conditional tree for: {'O', 'E'}
Null Set 1
- 3
conditional tree for: {'C'}
Null Set 1
- 3
conditional tree for: {'U'}
Null Set 1
T 3
conditional tree for: {'D'}
Null Set 1
- 4
  T 4
  E 3
conditional tree for: {'D', 'E'}
Null Set 1
- 3
  T 3
conditional tree for: {'T', 'D', 'E'}
Null Set 1
- 3
conditional tree for: {'T', 'D'}
Null Set 1
- 4
conditional tree for: {'E'}
Null Set 1
- 5
  T 4
conditional tree for: {'T', 'E'}
Null Set 1
- 4
conditional tree for: {'I'}
Null Set 1
- 4
  T 3
  E 2
  E 1
  T 1
conditional tree for: {'E', 'I'}
Null Set 1
- 3
conditional tree for: {'T', 'I'}
Null Set 1
- 3
conditional tree for: {'T'}
Null Set 1
- 5
[{'N'}, {'N', 'I'}, {'O'}, {'_', 'O'}, {'O', 'E'}, {'_', 'O', 'E'}, {'C'}, {'_', 'C'}, {'U'}, {'T', 'U'}, {'D'}, {'D', 'E'},
{'_', 'D', 'E'}, {'T', 'D', 'E'}, {'_', 'T', 'D', 'E'}, {'_', 'D'}, {'T', 'D'}, {'E'}, {'T', 'E'}, {'_', 'T',
'E'}, {'_', 'E'}, {'I'}, {'E', 'I'}, {'_', 'E', 'I'}, {'_', 'I'}, {'T', 'I'}, {'_', 'T', 'I'}, {'_', 'T'}, {'_', 'I'}]
```

## 四、結論

Apriori 在資料量大時，運算效率低下，但數據中只需要有關連數據即可，其他屬性不會用到。FP-Growth 演算法建立了一個像樹一樣的數據結構，它利用了頻繁項集之間的關聯性，大幅提高了速度。FP-Growth 演算法比 Apriori 演算法更快，它不需要排序候選項集，也不需要枚舉所有的候選項集。Apriori 和 FP-Growth 這兩個演算法當選取支持度很小時，計算時間明顯變長，性能影響很大，數據量大時可優先考慮 FP-Growth 演算法查找頻繁集

## 五、參考文獻

Chwang(2020 年 10 月 15 日)。Machine Learning -關聯分析-Apriori 演算法-  
詳細解說啤酒與尿布的背後原理 Python 實作-Scikit Learn 一步一步教學

<https://chwang12341.medium.com/machine-learning-%E9%97%9C%E8%81%AF%E5%88%86%E6%9E%90-apriori%E6%BC%94%E7%AE%97%E6%B3%95-%E8%A9%B3%E7%B4%B0%E8%A7%A3%E8%AA%AA%E5%95%A4%E9%85%92%E8%88%87%E5%B0%BF%E5%B8%83%E7%9A%84%E8%83%8C%E5%BE%8C%E5%8E%9F%E7%90%86-python%E5%AF%A6%E4%BD%9C-scikit-learn%E4%B8%80%E6%AD%A5%E4%B8%80%E6%AD%A5%E6%95%99%E5%AD%B8-76b7778f8f34>

ls 秦(2018 年 8 月 23 日)。Python 機器學習 — 關聯規則 (Apriori、FP-growth)

<https://www.twblogs.net/a/5b7dd3152b717768385411e0>