摘要

本實驗基於一份 2016 年 7 月份的交易資料集內容中所進行資料關聯規則分析，為探求出交易間的關聯性，找出消費者買了 A 產品之後還會買 B 產品的規則，提供推薦產品。實驗中，透過前置處理將資料集整理完，分別利用 Apriori 演算法和 FP-Growth 演算法設定不同的參數，如支持度及信心度，比較出不同演算法所呈現之推薦產品數量多寡之關係與比較使用該種演算法所花費時間，最後將規則輸出存檔。最後結果顯示，若需要較多關聯規則時，使用 FP-Growth 演算法為較好的選擇。

關鍵字：Apriori；FP-Growth；資料分析；關聯分析；機器學習

一、緒論

1.1 動機

在以往都是給予消費者問卷作答，從消費者問卷回答中提供的答案做產品銷售規劃，然而本實驗將利用消費者的交易紀錄，從中求得購買產品之間的關聯性，正確找出購買產品的關聯規則，以供推薦給消費者相關產品。

1.2 目的

在以問卷調查的方式詢問消費者意見時，可能會因為一些因素影響，例如消費者主觀或外在資訊接收影響，也會因為部分的消費者不想填寫而沒有辦法得到資訊。而透過實際的交易紀錄求得交易關聯規則之方法能夠排除上述所提的問題，本實驗將使用交易資料集.xlsx 檔案，求出關聯規則，為消費者提供相關的推薦產品。

二、資料集

2.1 真實資料集

本實驗所使用資料集為交易資料集.xlsx，包含了七個欄位，分別是 ITEM_ID、ITEM_NO、PRODUCT_TYPE、CUST_ID、TRX_DATE、INVOICE_NO 及 QUANTITY ，資料總筆數為 157396 筆，其中，相同 INVOICE_NO 代表同一筆交易，而交易紀錄日期為 2016 年的 7 月 1 日至同年的 7 月 29 日，由於筆數過多無法全部呈現，部分資料內容如表 1 所示。

表 1

交易資料集

ITEM_ID	ITEM_NO	PRODUCT_TYPE	…
3217532	M25P40-VMN6TPB	MEMORY_EMBEDED	…
3326781	AU80610006237AASLBX9	CPU / MPU	…
…	…	…	…

三、方法

3.1 實作說明

本實驗針對交易資料集.xlsx 的處理分為了 五個階段。第一階段做前置處理，將資料集裡數量為 0 或負值的交易代表退貨或註銷，因此透過 (df[['QUANTITY']] > 0).all ，移除數量小於等於 0 的交易。第二階段將屬於同一筆交易的商品 ID 整理在一起，利用 set() 的不重複特性整理出所有不同交易，將屬於不同交易的資料整理出。第三階段透過 sorted 排序資料後，將冗餘規則剔除，減少其規則數量，以利後續進行實驗。第四階段將前處理過的資料集使用 Apriori 與 FP-Growth 演算法，透過兩種演算法設定不同的支持度 (support)與信心度 (confidence) 後，產生出關聯規則表。第五階段將產生的關聯規則表輸出存檔成 xlsx 檔案。

3.2 操作說明

透過讀取關聯規則表檔案允許使用者輸入產品數量與產品代號，如圖 1 所示。並透過關聯規則推薦產品，如圖 2 所示。若使用者輸入超過產品數量的值會顯示使用者輸入錯誤的提醒。

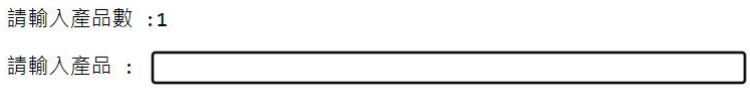

圖 1 輸入產品數量與產品代號

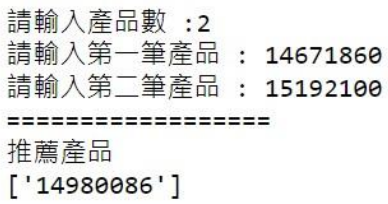

圖 2 輸入成功並推薦產品

四、實驗

4.1 前置處理

將資料集放入 df 變數中，透過 (df[['QUANTITY']] > 0).all 保留數量大於 0 的交易，利用 set() 集合將同一筆交易的商品 ID 整理在一起，找出所有不重複的交易，共有 15042 筆，並將屬於同一筆交易的資料整理出來，以利套入 Apriori 演算法和 FP-Growth 演算法中。

4.2 實驗設計

分別建構 Apriori 和 FP-Growth 模型，將整理好的資料放入模型中，其中，29 天中每天至少會買 3 次的話，支持度相當於 29（天）乘上 3（次）再除以 15042（所有不重複的交易筆數）。因此，本實驗在 Apriori 模型中設計了 3 個不同的支持度情況，分別為 29 天中至少會買 3 次、29 天中至少會買 2 次、29 天中至少會買 1 次的三個支持度。

4.3 實驗結果

以下表 2 及表 3 分別根據 Apriori 演算法和 FP-Growth 演算法所設定不同的支持度及信心度，記錄規則數量及執行時間之變化。

表 2

Apriori 演算法之規則數量與執行時間

支持度	信心度	規則數量	執行時間 (s)
.0019	.2	234	.3734
.0038	.2	6	.1097
.0050	.2	1	.0508

表 3

FP-Growth 演算法之規則數量與執行時間

支持度	信心度	規則數量	執行時間 (s)
.0019	.2	864	.7823
.0038	.2	11	.6729
.0050	.2	2	.7682

表 4 為 Apriori 演算法與 FP-Growth 演算法的執行時間比較。

表 4

Apriori 演算法與 FP-Growth 演算法之花費時間

支持度	信心度	Apriori 執行時間 (s)	FP-Growth 執行時間 (s)
.0019	.2	.3734	.7823
.0038	.2	.1097	.6729
.0050	.2	.0508	.7682

五、結論

本實驗中，透過設定支持度與信心度可以決定關聯規則產生的數量。當支持度設定越小時會產生越多的關聯規則。在 Apriori 演算法與 FP-Growth 演算法比較後可以得知在執行時間上 Apriori 演算法耗費的時間較 FP-Growth 演算法來的少。在同等支持度與信心度的情況下產生的關聯規則 FP-Growth 演算法比 Apriori 演算法來的多，因此可以由此得出若需要較多關聯規則時，使用 FP-Growth 演算法為較好的選擇。

六、參考文獻

Anonymous. (2014, March 6). *Pandas: Change Data Type of Series to String*. Stack Overflow. https://stackoverflow.com/questions/22231592/pandas-change-data-type-of-series-to-string

Anonymous. (2018, January 18). *Dropping Rows from a PANDAS Dataframe Where Some of the Columns Have Value 0*. Stack Exchange. https://codereview.stackexchange.com/questions/185389/dropping-rows-from-a-pandas-dataframe-where-some-of-the-columns-have-value-0

Bejamin naibei. (2021, November 4). *Getting Started with Apriori Algorithm in Python*. Section. https://www.section.io/engineering-education/apriori-algorithm-in-python/

Spencer guy . (2021, February 15). *Pandas Sort: Your Guide to Sorting Data in Python*. Real Python. https://realpython.com/pandas-sort-python/