國立成功大學-資工所

Data Mining 課程

**Project 1**

**Association Analysis**

課程教授: 高宏宇

學生: 李俊賢

學號: P96074121

1. FP-Growth程式實作
2. **作業環境**

作業系統: Window 10 Pro

處理器: intel i7-7700 @3.60GHz

記憶體: 8GB

程式語言: python

編譯器: Spyder3.6

額外工具: WEKA

1. **概略程式碼**

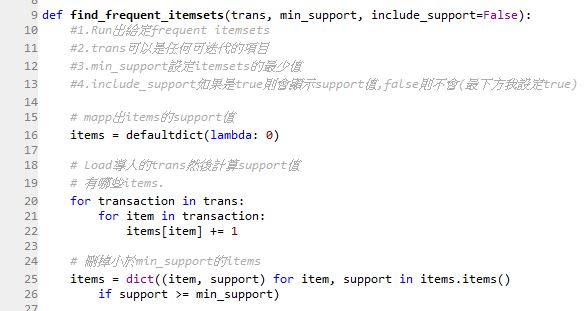


圖1.Run出給定frequent itemsets並且去除少於min\_suport的item



圖2.對於每個頻繁項，通過以下步驟求它的條件頻繁項集：找出它的條件模式並把條件模式當做itemsets去建造一棵樹，這棵樹不叫FP-Tree，而叫做該頻繁項的條件FP-Tree。對這棵條件FP-Tree遞歸以上操作，即找這棵條件FP-Tree上的子條件頻繁項集。以上找到的都是該頻繁項的條件頻繁項集而已，所以每次遞歸都需要把條件頻繁項集和該頻繁項拼接起來才是我們最終要求的頻繁項集





圖3.讀取CSV檔案內容以迴圈輸出每一列用find\_frequent\_itemsets()產生frequent\_itemsets設定min\_support後輸出結果

1. **IBM的data**

先從IBM產生data，按照老師給的步驟先進入IBM Quest Data Generator存放的資料夾然後打” IBM Quest Data Generator.exe” -lit -help就能參考設定參數，我是設定-ascii -ntrans 10 -tlen 10 -nitems 3下去跑最後產生如下圖資料

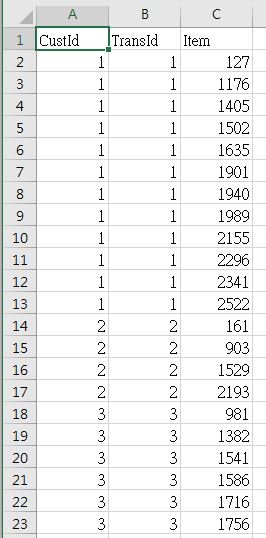


圖4.IBM產生的data總共有10000多筆不過只有一行隨機亂數

所以能找的frequent pattren很少

然後，進入程式碼內找到如下圖的部分，下圖中，綠色字體的部分為data路徑，讀取csv檔後已回圈的方式存在陣列裡

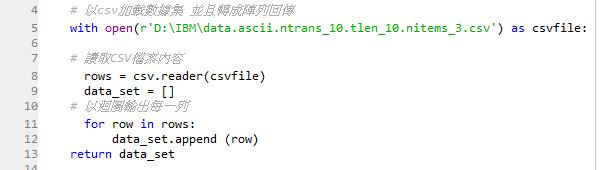


圖5.程式碼data匯入部分

1. **運算結果**

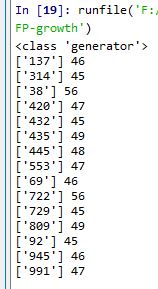


圖6.因為IBM設定我只設定一行的隨機亂數，所以運算結果只有數字出現的總量這邊我min\_spupport設定45，因為都是數字所以我把他們當成遊戲腳色的編號，那我就能知道那些腳色的出場率最高，如過數據完整我還能知道那些腳色勝率高



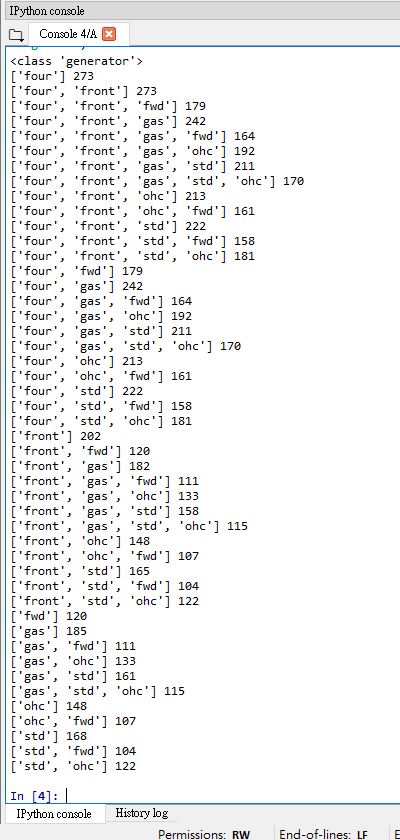
圖7.將關聯結果畫出並轉換成遊戲腳色名稱

1. **Kaggle的data**

****

這是我從Kaggle找的汽車數據有廠牌，汽油種類，門的數量，車身風格，傳動情形，引擎位置，引擎種類，汽缸數量，燃油系統，轉速大約200筆因為weka跑不了太多筆

1. **運算結果**



1.(引擎種類:ohc，傳動:fwd) 🡺(汽缸數:4)

2.(傳動:fwd) 🡺(汽缸數:4)

3.(傳動:fwd) 🡺(引擎種類:ohc)

4.(引擎種類:ohc) 🡺(汽缸數:4)

1. Aprior-程式實作
2. **作業環境**

作業系統: Window 10 Pro

處理器: intel i7-7700 @3.60GHz

記憶體: 8GB

程式語言: python

編譯器: Spyder3.6

額外工具: WEKA

1. **概略程式碼**

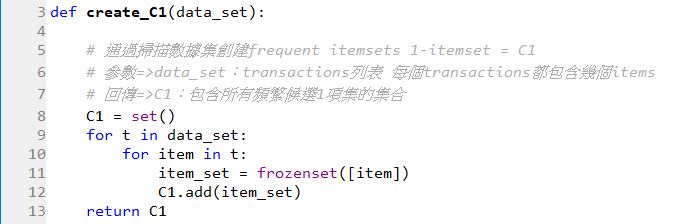


圖.通過掃描數據集創建frequent itemsets 1-itemset = C1

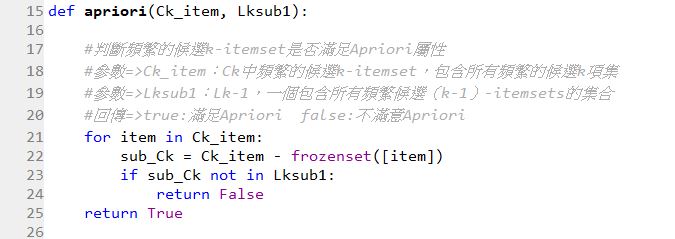


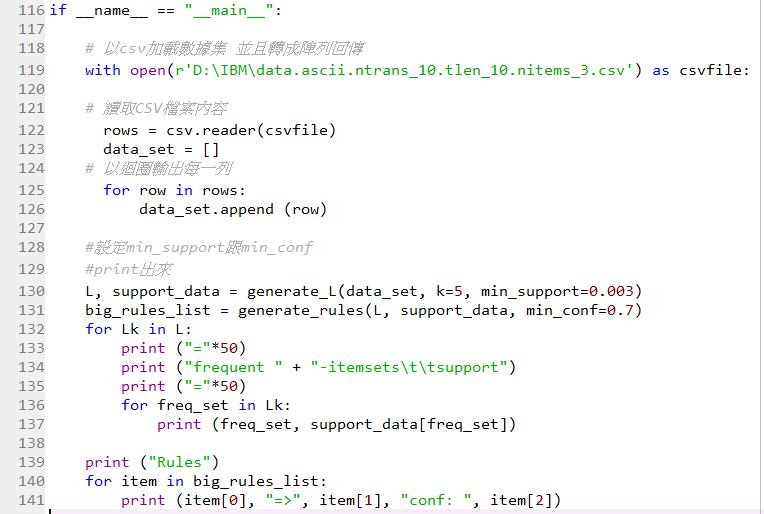
圖. 判斷頻繁的候選k-itemset是否滿足Apriori屬性

圖3.以csv加載數據集並且轉成陣列回傳，設定min\_support跟min\_conf並且印出結果

1. **IBM的data**

先從IBM產生data，按照老師給的步驟先進入IBM Quest Data Generator存放的資料夾然後打” IBM Quest Data Generator.exe” -lit -help就能參考設定參數，我是設定-ascii -ntrans 10 -tlen 10 -nitems 3下去跑最後產生如下圖資料

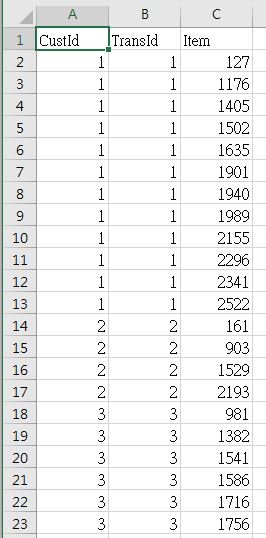
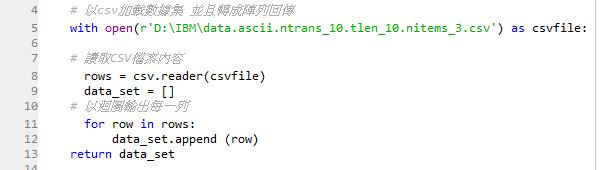


圖.IBM產生的data總共有10000多筆不過只有一行隨機亂數

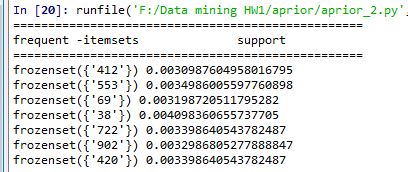
所以能找的frequent pattren很少

然後，進入程式碼內找到如下圖的部分，下圖中，綠色字體的部分為data路徑，讀取csv檔後已回圈的方式存在陣列裡



圖七. 程式碼data匯入部分

1. **運算結果**

****

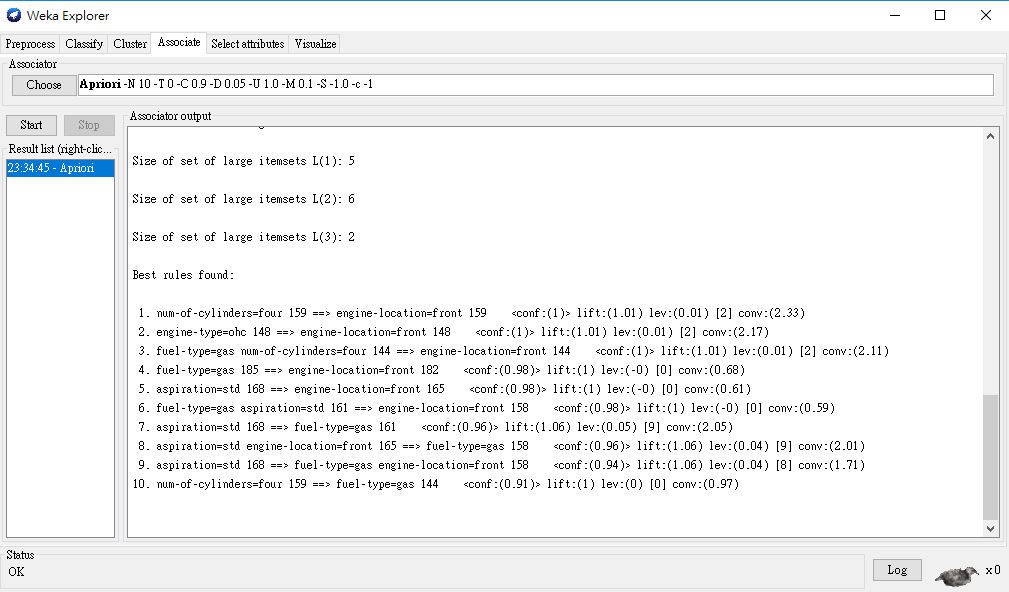
因為IBM設定我只設定一行的隨機亂數，所以運算結果只有數字出現的總量這邊我min\_spupport設定0.003大約是30這邊是在跟FP-growth比較，其實都大同小異只是我的aprior會算到CustID跟TransID的數字

1. **IBM的data**



這是我從Kaggle找的汽車數據有廠牌，汽油種類，門的數量，車身風格，傳動情形，引擎位置，引擎種類，汽缸數量，燃油系統，轉速大約200筆因為weka跑不了太多筆

1. **運算結果**

****

1.(引擎種類:ohc，傳動:fwd) 🡺(汽缸數:4)

2.(傳動:fwd) 🡺(汽缸數:4)

3.(傳動:fwd) 🡺(引擎種類:ohc)

4.(引擎種類:ohc) 🡺(汽缸數:4)

1. **結論**

IBM跑不出多行的亂數，導致FP跟aprior跑不出頻繁項

Weka跑不動數量太多的data

FP\_growth跟Aprior算出來的結果差了一點點