**科⽬：1071\_資料探勘 DATA MINING**

**學⽣：楊沛霖**

**學號：Q36071156**

**日期：2018-10-23**

**前言：**

1. **本次作業利用不同的工具( 自己方法、WEKA )以及資料（老師課堂資料、IBM、Kaggle）使用FP-growth來尋找freqItems 並比較不同工具的結果。**
2. **本次作業的inputData都放在use\_data資料夾內，outputData則放置在save\_data資料夾。**

**Topic 1:**

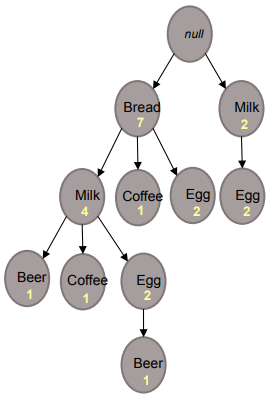
先以老師上課資料確認程式運行是否正確。

Min\_sup : 2

執行時間 : 1.1854257

**使用資料(IBM Quest Data Generator):**

**FP-growth:**

 老師講義上的圖(圖 一)與這次程式實作出來的圖(圖 二)

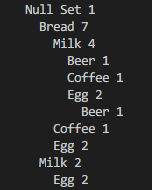
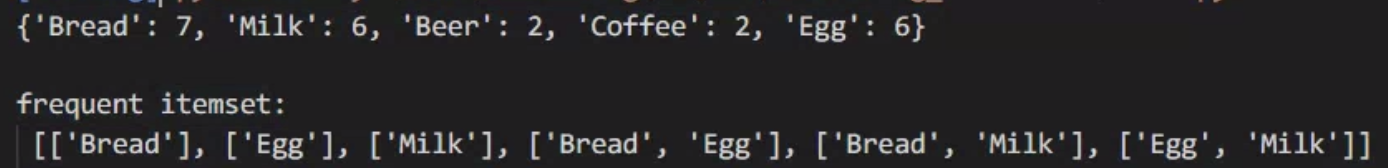


圖 二

圖 一

**Apriori:**



**Association analysis:**

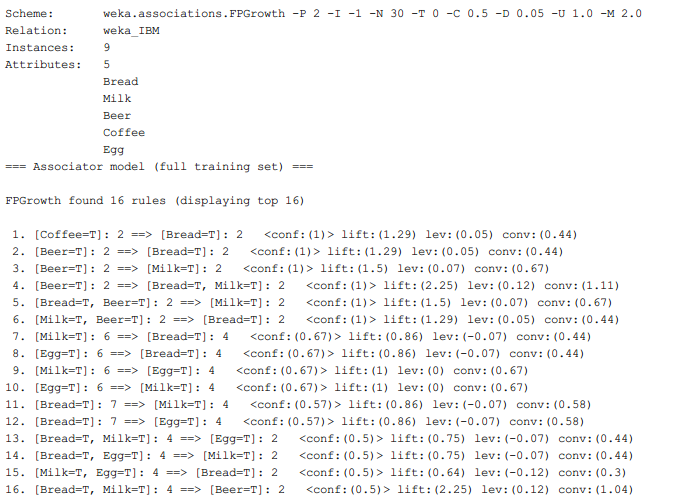
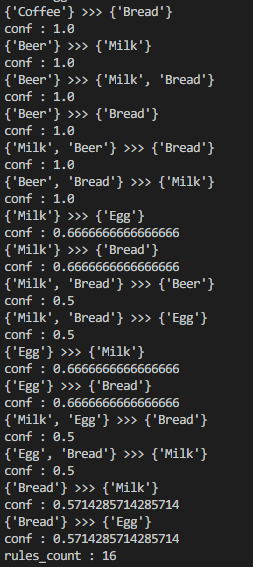
 分別為自己寫的程式(圖 四)與WEKA的FP-tree(圖 三)的關聯顯示,可以看到兩邊答案相同. 對應function為start.py中的associate(freqItems, simpDat)

圖 三

圖 四

**比較結果:**

程式經過老師範例的洗禮過後，能正常運行並且和分析工具WEKA所顯示的資料相同，因此可以加入IBM資料去處理了。可以看到程式運作出來的顯示和WEKA所出現的Rule是一致的，因此程式應該是成功。

**Topic 2:**

實際運行IBM Quest Data Generator所生成的資料

Min\_sup : 2

執行時間 : 2.124941

**使用資料(IBM Quest Data Generator):**

|  |  |
| --- | --- |
| **TID** | **ItemSet** |
| 1 | {247,506,578} |
| 2 | {247,506,578} |
| 3 | {247,506,247} |
| 4 | {247,506,578} |
| 5 | {247,506,578} |
| 6 | {247,506,578} |

**使用參數 :**

lit -ntrans 1 -ascii -tlen 5 -nitems 10 -npats 3

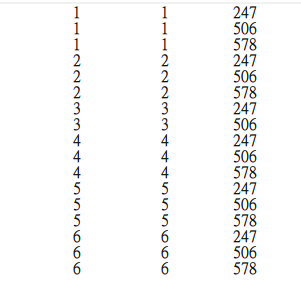
圖 五為IBM Quest Data Generator， 表一 為切換成老師投影片的形式。

圖 五

表 一

**FP-****growth:**

有程式生成的FP-growth(圖 六)

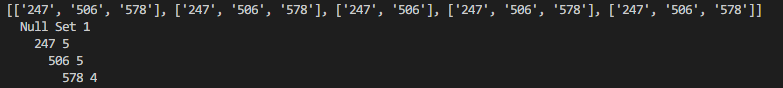
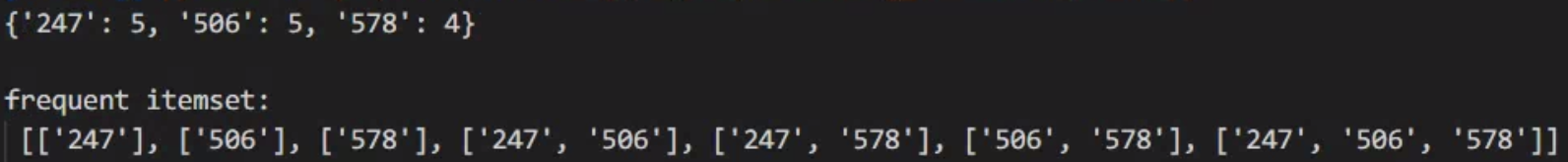


圖 六

**Apriori:**



**Association analysis:**

分別為自己寫的程式(圖 八)與WEKA的FP-tree(圖 七)的關聯顯示,可以看到兩邊答案相同。

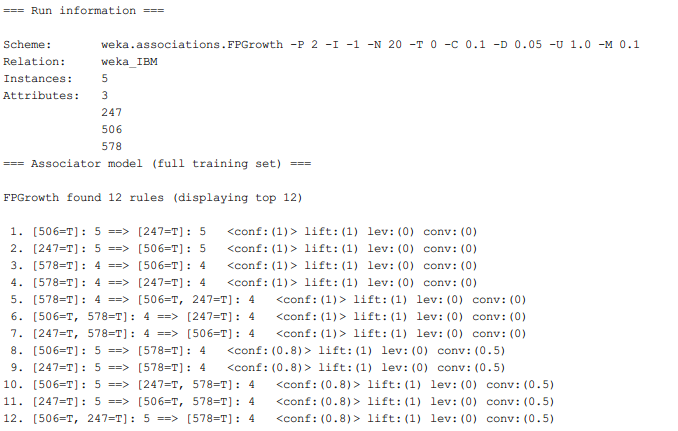
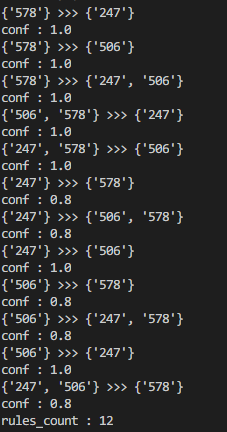


圖 七

圖 八

**比較結果:**

無論資料量大小皆與WEKA的分析結果一致，不過在自己associate的寫法中沒辦法像WEKA能夠把除了True或False的資料顯示出來，只能簡單的用存不存在去做對比。

**Topic 3:**

**使用資料(Kaggle):**

**使用資料來源 :**

<https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings>

**資料目的：**

此資料為判斷每種Video Games主機銷量與顧客反應，在這裡可以看到不同國家或不同地區的銷售狀況還有評論狀況還有遊戲分數等等資訊。

而我所使用的是它的遊戲主機和用戶評分的來做比較，可以看到由於分數的分數極為零散因此整顆數非常的巨大。

**FP-growth（部分）:**



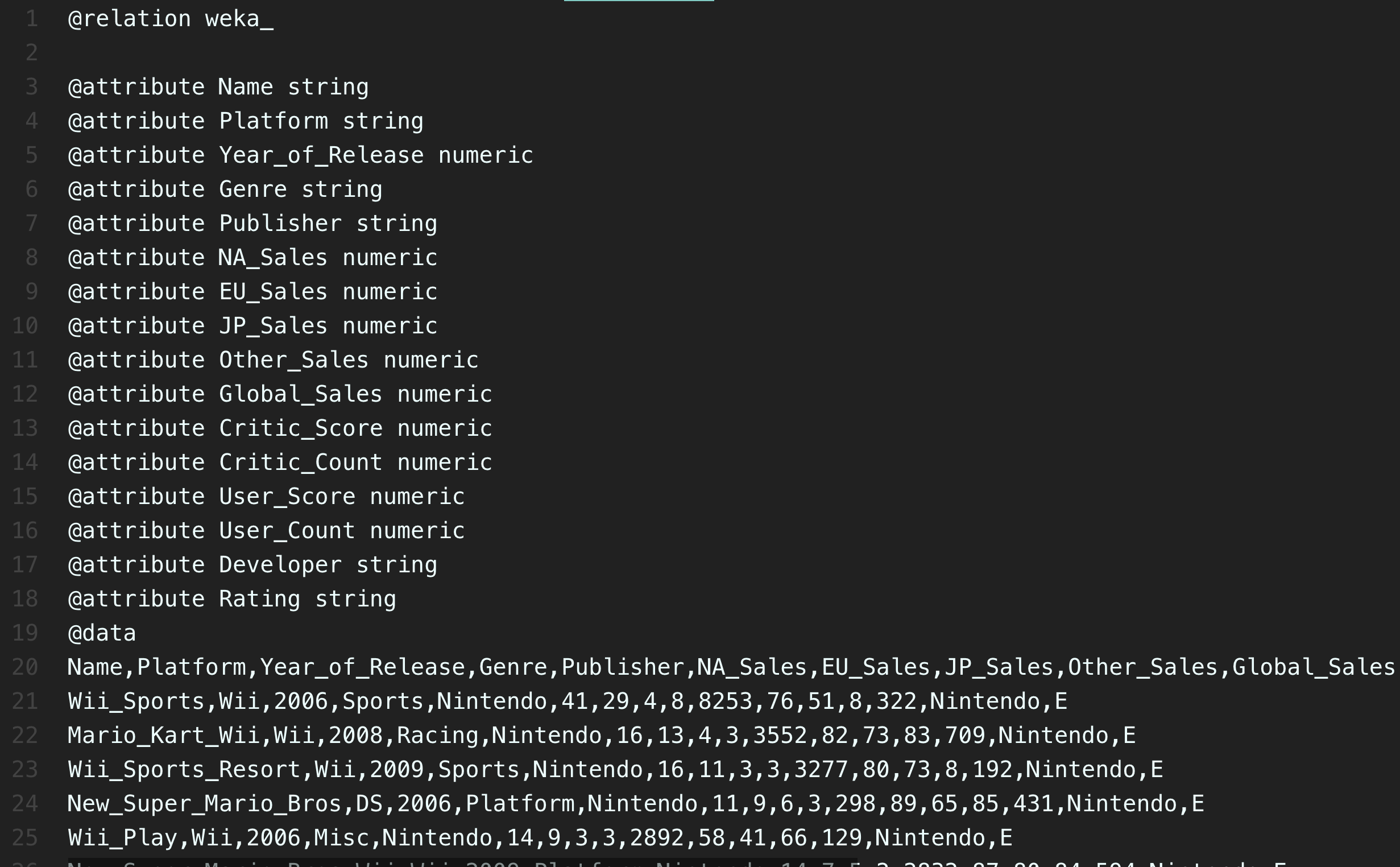
**經處理後的WEKA讀取文件（arff）:**

**文件名稱 ： @relation**

**Table Header 屬性 ： @attribute**

**屬性分為 :**string , numberic , nominal, date

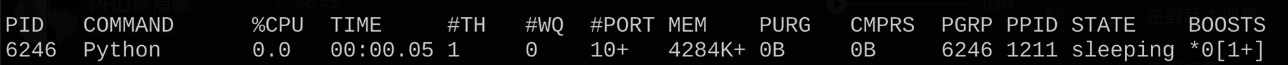
分別為 字串, 數字, 自定義屬性（ex. temperature {hot, mild, cool}）, 日期



**結果與討論**

**IBM Data(FP):**

Time : 0.04s user 0.02s system 70% cpu 0.088 total

Memory:

**IBM Data(Apriori):**

Time : 0.04s user 0.02s system 67% cpu 0.086 total

Memory: 速度太快拍不到＠＠

**Kaggle Data:**

Time : 0.04s user 0.03s system 35% cpu 0.206 total

Memory:

這裡可以發現由於Kaggle所給的資料比IBM的還要多因此可以看到在MEM的地方，Kaggle硬是比IBM多出280K左右的記憶體使用量。而運作時間也是Kaggle最慢筆IBM多了0.1秒左右而Apriori最快連memory變化都拍不到。不過這裡有一個很有趣的地方，就是雖然Kaggle的時間增加了但是在system cpu的方面卻比IBM少了一半的趴數。

**心得:**

由於樹的特性很適合利用LinkList的方式去寫因此，在建立Tree之前我創立了一個Node的資料型態，裡面可以放它的Name、Parents、Children、Value，如此一來建立一棵樹就會變得方便許多。在之後的FP-Growth的相關操作方面，把老師投影片的演算法看過一步步照著設計即可。

在IBM資料的產生方面由於完全不懂他的運作方式，一開始沒有看網路上解說直接讓他產生資料就跑了半個小時，當初還覺得很厲害，原來這就是為甚麼別人說『訓練資料要訓練很久』，當下還覺得很興奮!但是實際跑過一次程式後發現根本不能跑，還要處理Memory的問題，因此上網找了他的相關參數，用了一個可以跑得數據即可。

而且在WEKA的使用方面也找了不少網路上的影片or資訊去看，看完後發現雖然它的介面沒有很炫泡，但是裡面功能卻是十分完善的，只要將資料換成它所適合的型態再給他的型態做設定就可以幫我們做簡單的資料預測了。