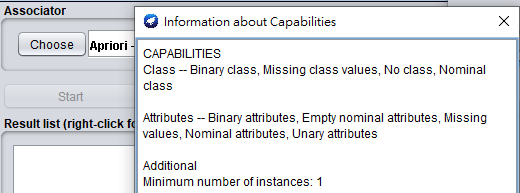
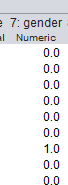
ECT HW2

**Q1.**

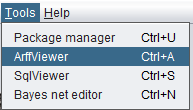
**(a)**

**Part 1 : 為何原來的檔案不能執行?**

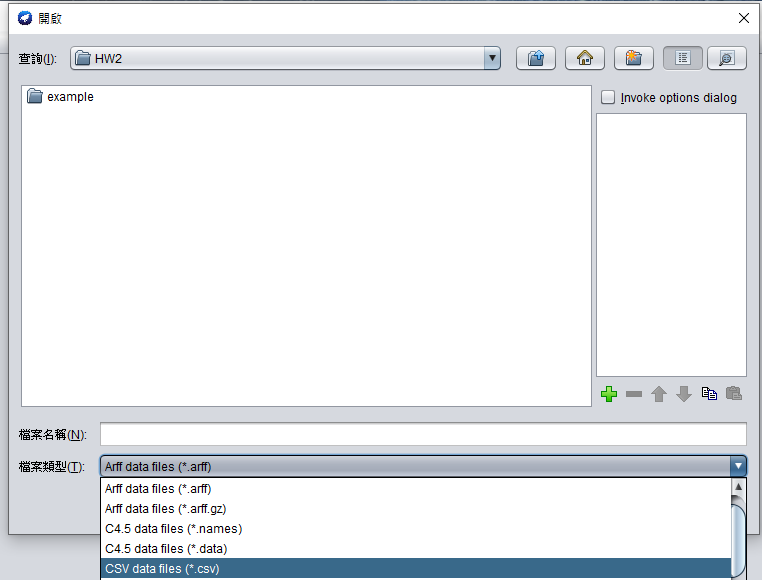
* 如圖中所示，Apriori這個方式不能使用numeric的data，但原始數據中

gender這一個屬性值為0、1，所以無法使用。

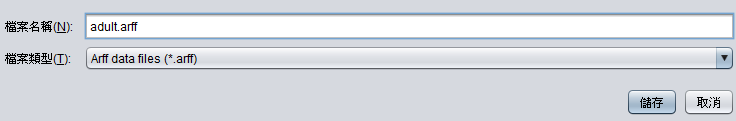
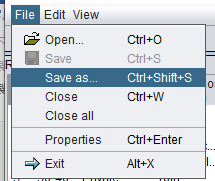
**Part 2 : 轉換成.arff**



* 先用ArffViewer，他可以開啟csv並另存為arff

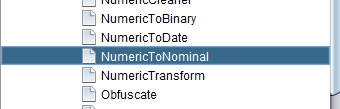


* 開啟檔案時，記得選CSV格式，不然會找不到檔案

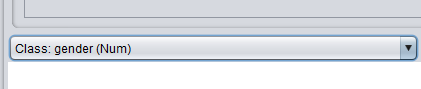


* 開啟後直接另存為.arff就可以了

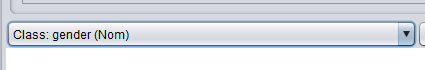
**Part 3 : 把numeric的0、1轉換成nominal的Male、Female**

****

* Weka有提供這個工具，把Numeric轉成Nominal

****

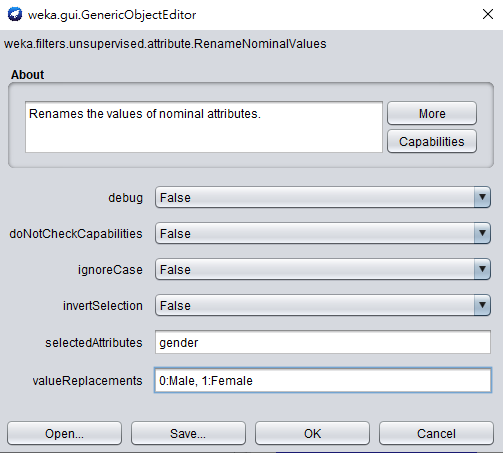
* 使用前gender為Numeric



* 使用後為Nominal，但value仍為0、1所以要改value



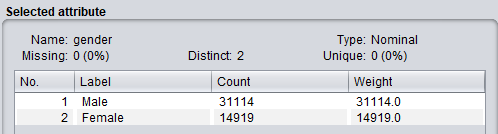
* Weka一樣有內建改Nominal Value的工具

.

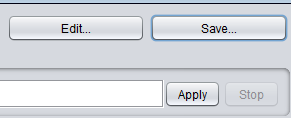
* 最下面2個欄位，分別用來指定attribute，並為對應的值做轉換。在此我指定gender屬性，並把0轉成Male，1轉成Female。



* 設定完之後記得按Apply，不然什麼事都不會發生

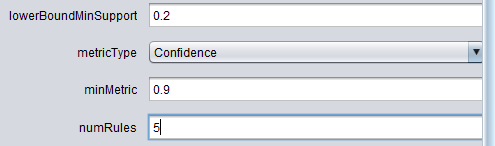


* 檢查一下，0的確變Male，1也變成Female了。

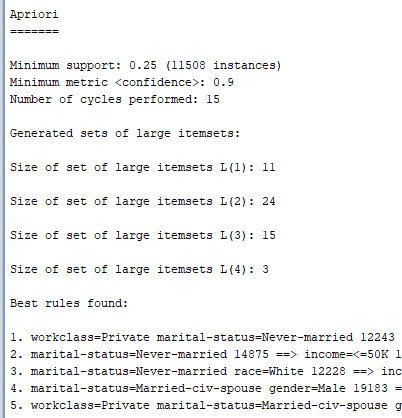


* 最最最重要的一步，請把他另存一份檔案，因為這裡做的修改都是暫時的，不儲存下次就都沒了。

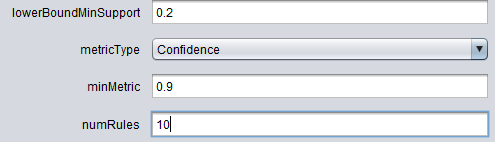
**(b)**



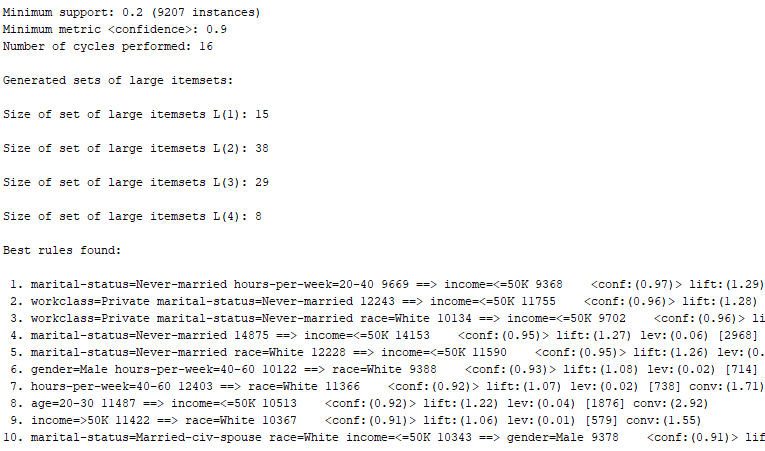
* 先設定為numRules = 5，Confidence和Support都是依照題意設定



* Minimum Support = 0.25



* 設定為numRules = 10



* Minimum Support = 0.2

造成原因:

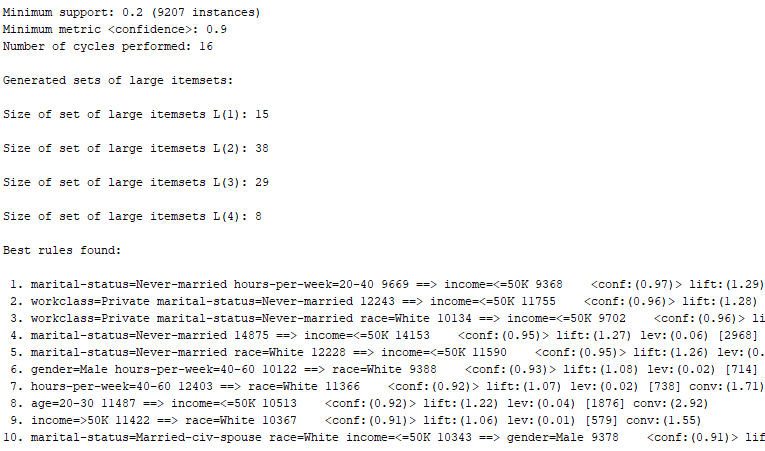




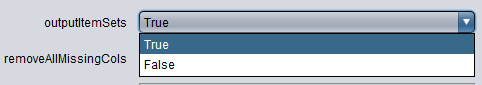
* 由上圖可知，我們設定最低Support = 0.2，並從Support = 1開始找Rule，若沒有找到每次Support就 – 0.05 (delta)。

因此我們可推知，在numRules = 5的條件下，Support遞減至0.25時就找到5條Rules了，但在numRules = 10的條件下，因為要找的Rules數量變多了，導致它在Support = 0.25時並未找完10條Rules，因此又減了一次0.05讓Support = 0.25 – 0.05 = 0.2去找剩下的Rules。

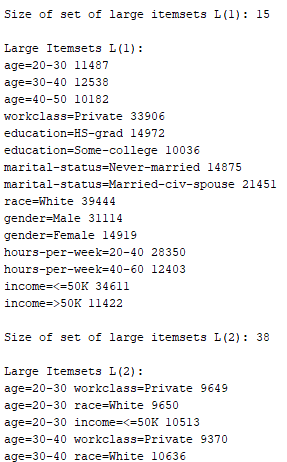
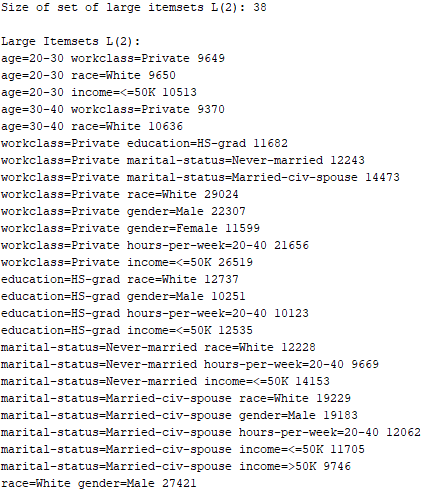
**(c)**

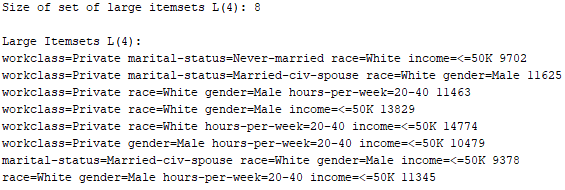
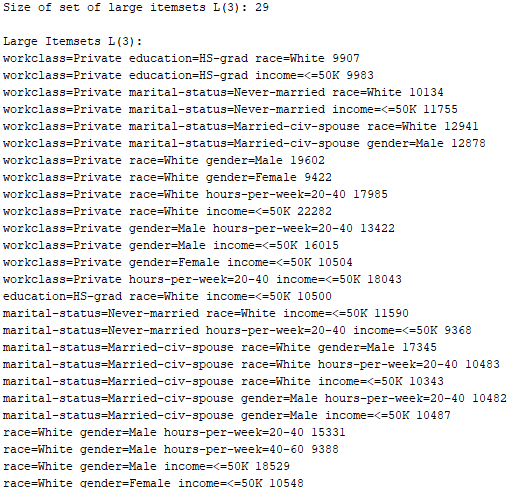


**(d)**



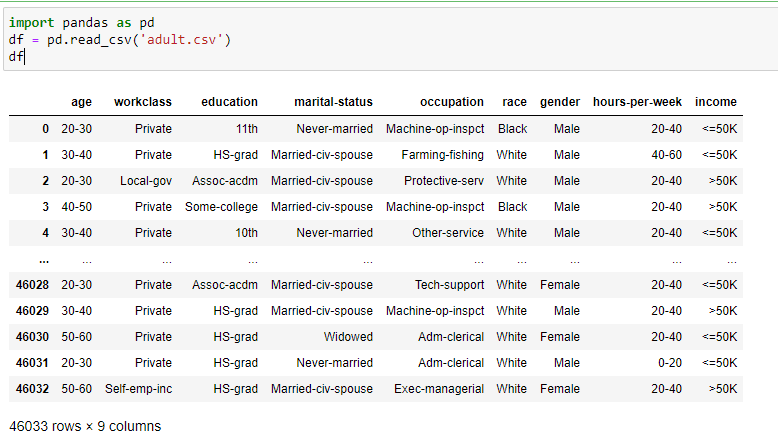
* 把outputItemSets調成True



**Q2.**

**(e)**

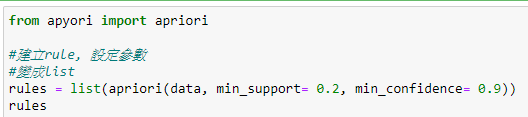


* 讀檔，並看資料大致樣貌



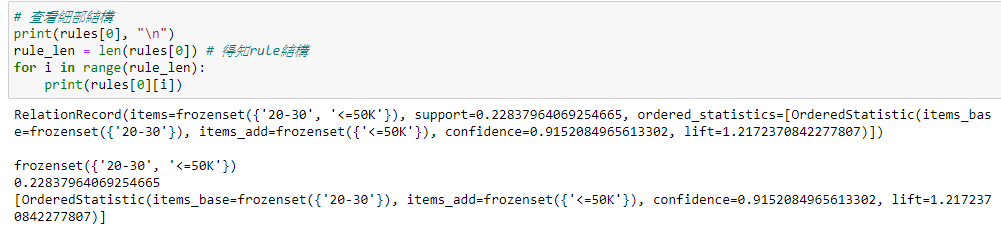
* 先把型態都轉成string避免之後有型態轉換的問題

再把資料弄成List型態，之後要當作input



* 用apriori做關聯式分析，第一個參數就是使用的資料，第二個參數是最低support值，第三個參數是最低confidence值

做完之後轉成list，是為了方便觀察分析完的資料結構為何。

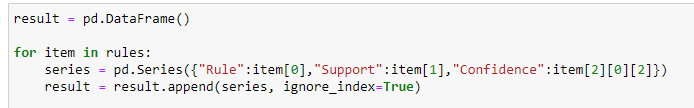


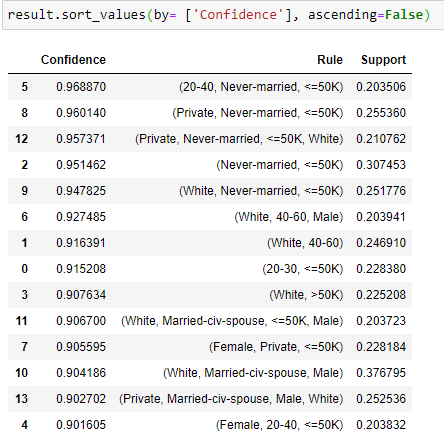
* 事先了解資料存儲的結構，在此可得知每條rule由3個部分組成。

Index = 0 : 代表rule的部分

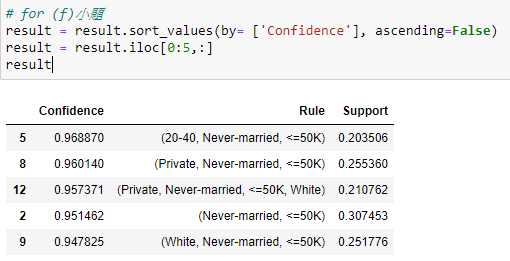
Index = 1 : 代表support為多少

Index = 2 : 整體的rule結構，裡面有一個元素，在此元素中又包含數個元數，其中我們在意的元素Confidence在第3個位置。



* 剛剛已經分析完結構了，接著我們要把我們關心的資料 : Rule、Support、Confidence放進一個Series中(就是一個很像list的東西)，可以想像成儲存成一筆instance，再把每筆Series存進DataFrame中(一個2維表格)，Rule、Support、Confidence對應的位置就如同上述所分析。
* 然後用sort\_values排序，by = [] 代表要用甚麼屬性排序，ascending = ?代表是否要升冪排列，若設False代表降冪排列。

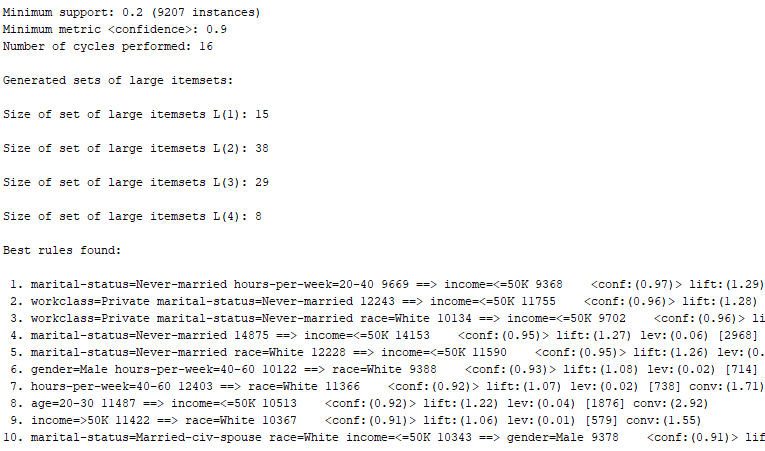
**(f)**



* 先用sort\_values按照Confidence由高到低排序，因為Weka默認就是依照Confidence的高低排序。

再用iloc函數挑出前5個instance，iloc是用index來挑選元素的方法。

In Weka:



可以看出，Weka Confidence的數值應是四捨五入制小數點後第二位，將python計算出的數值做此運算後，正好相符。仔細對比也可看出，前5條的rule都是一樣的，因此兩者是相符的。