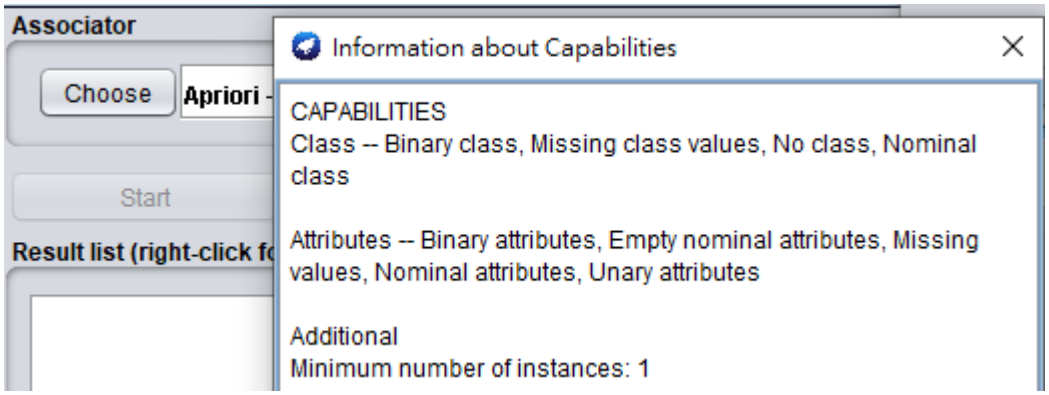
ECT HW2

Q1.

(a)

Part 1： 為何原來的檔案不能執行?

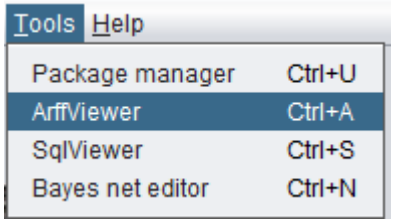


➔ 如圖中所示，Apriori 這個方式不能使用 numeric 的 data，但原始數據中

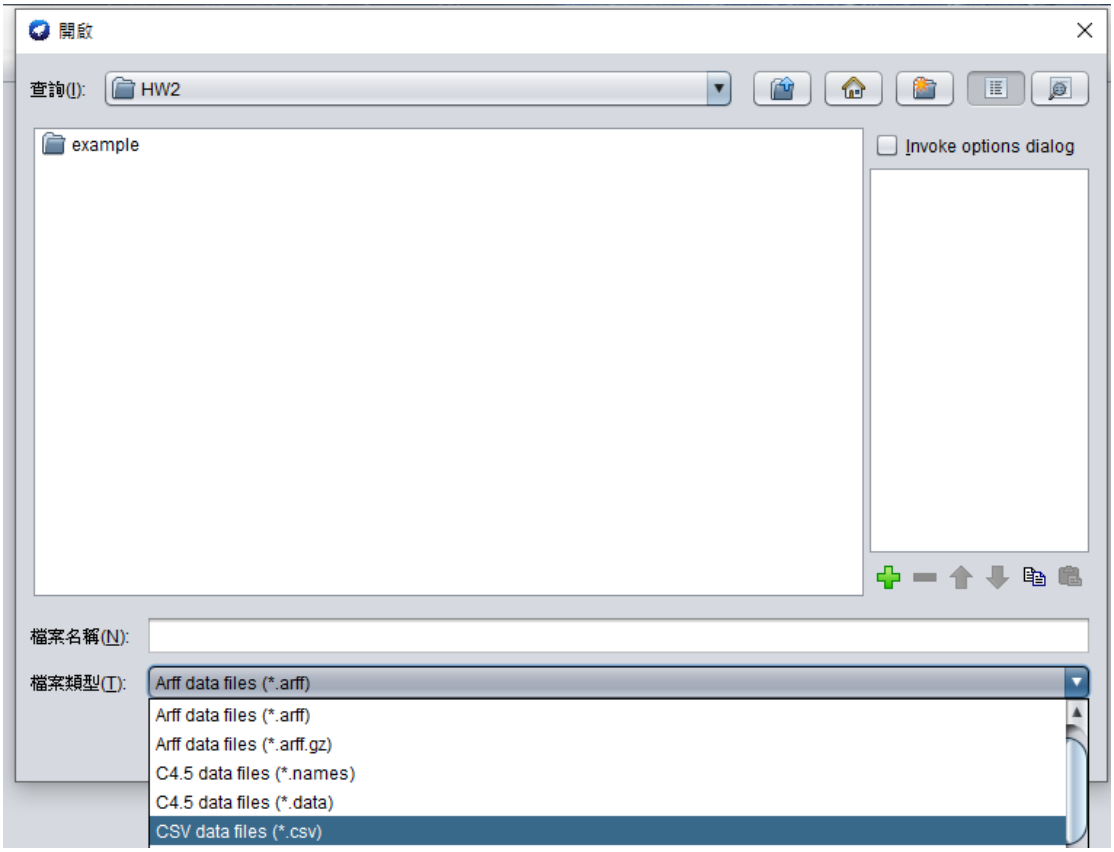
7: gender Numeric
0.0
0.0
0.0
0.0
0.0
0.0
1.0
0.0
0.0

gender 這一個屬性值為 0、1，所以無法使用。

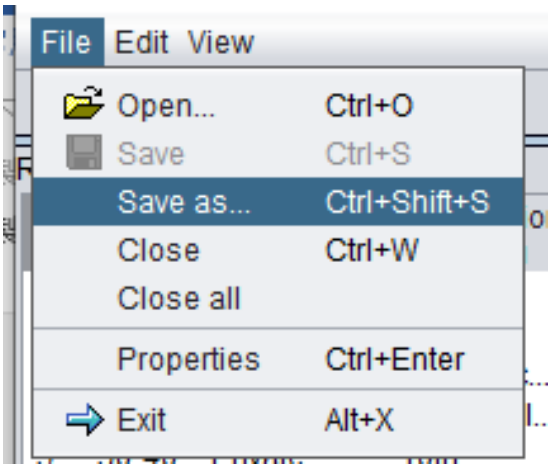
Part 2： 轉換成.arff

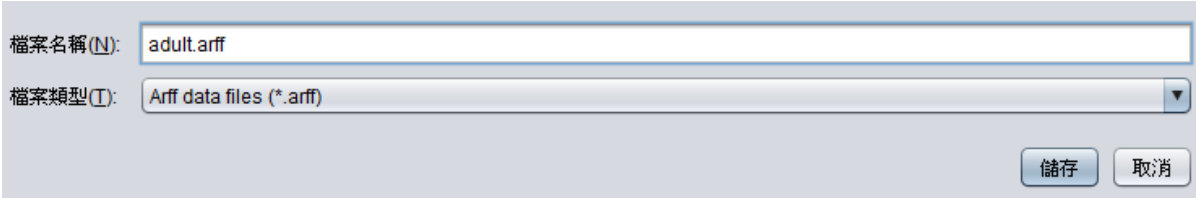


➔ 先用 ArffViewer，他可以開啟 csv 並另存為 arff



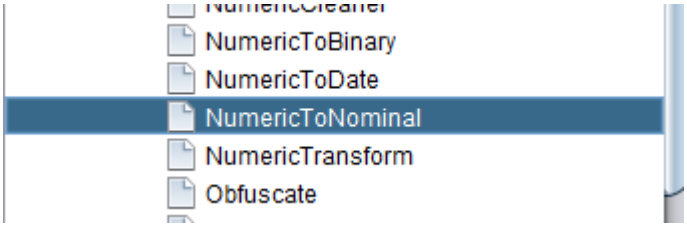
➔ 開啟檔案時，記得選 CSV 格式，不然會找不到檔案



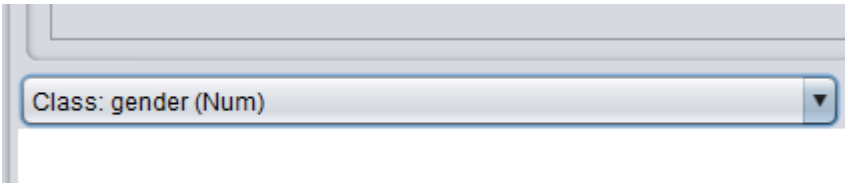


➔ 開啟後直接另存為.arff 就可以了

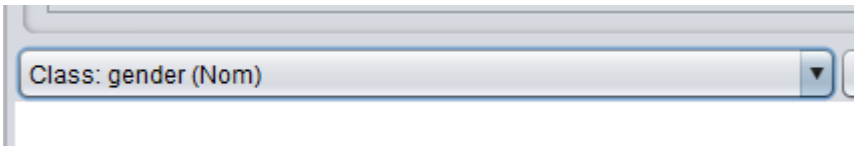
Part 3： 把 numeric 的 0、1 轉換成 nominal 的 Male、Female



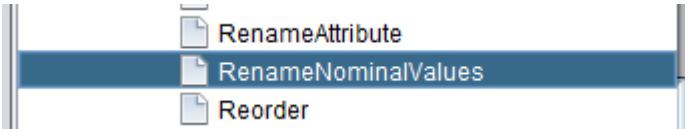
➔ Weka 有提供這個工具，把 Numeric 轉成 Nominal



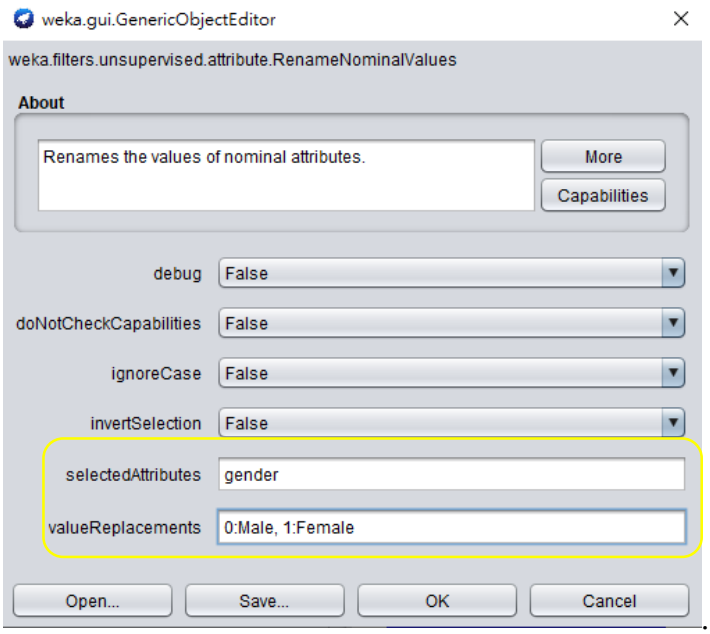
➔ 使用前 gender 為 Numeric



➔ 使用後為 Nominal，但 value 仍為 0、1 所以要改 value

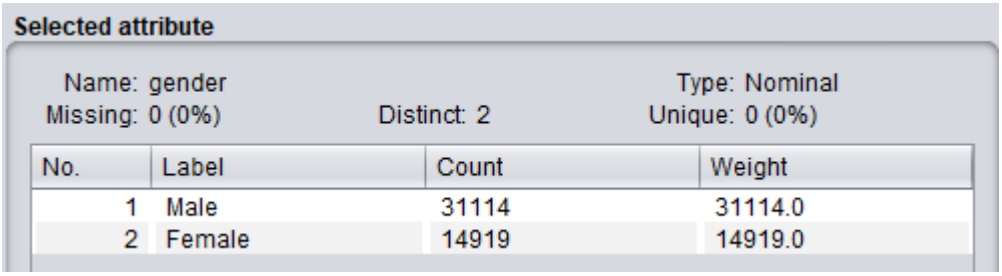


➔ Weka 一樣有內建改 Nominal Value 的工具

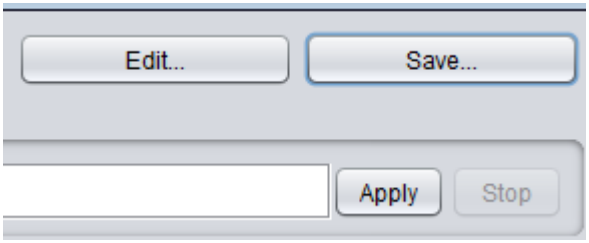


➔ 最下面 2 個欄位，分別用來指定 attribute，並為對應的值做轉換。在此我指定 gender 屬性，並把 0 轉成 Male，1 轉成 Female。

➔ 設定完之後記得按 Apply，不然什麼事都不會發生

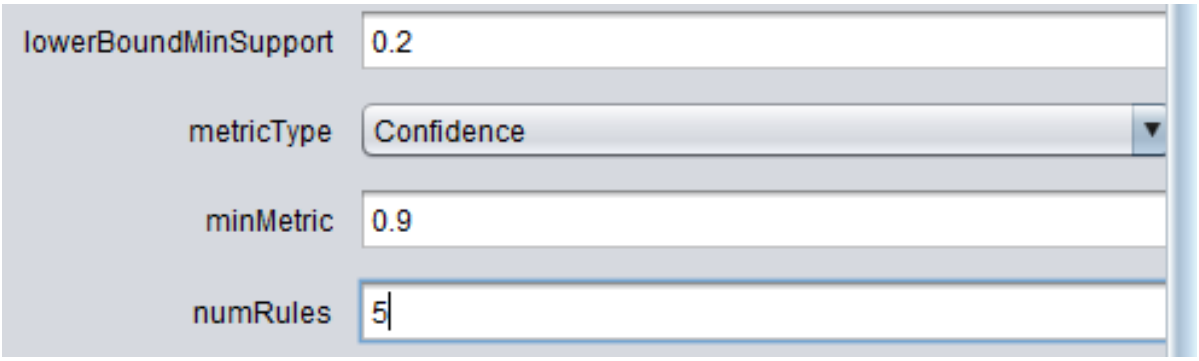


➔ 檢查一下，0 的確變 Male，1 也變成 Female 了。



➔ 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。

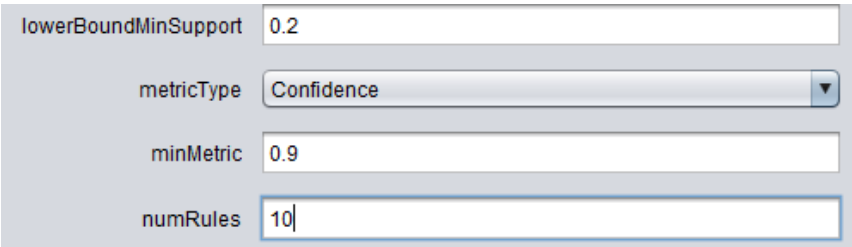
(b)



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

```
Apriori
=======

Minimum support: 0.25 (11508 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 24

Size of set of large itemsets L(3): 15

Size of set of large itemsets L(4): 3

Best rules found:

1. workclass=Private marital-status=Never-married 12243
2. marital-status=Never-married 14875 ==> income=<=50K 1
3. marital-status=Never-married race=White 12228 ==> inc
4. marital-status=Married-civ-spouse gender=Male 19183 =
5. workclass=Private marital-status=Married-civ-spouse g
```

➔ Minimum Support = 0.25

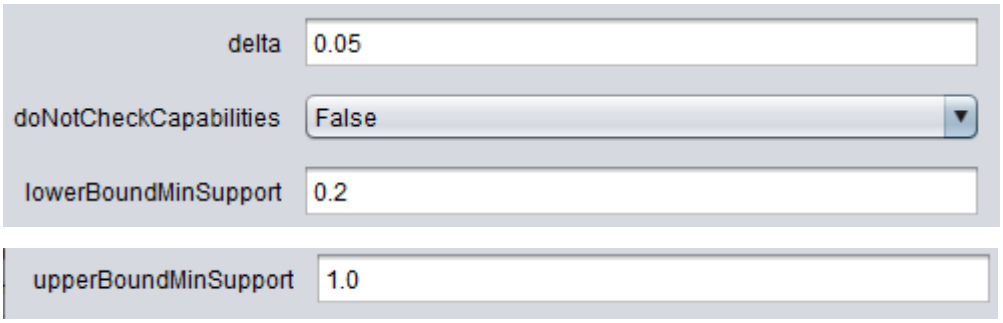


➔ 設定為 numRules = 10

```
Minimum support: 0.2 (9207 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 15

Size of set of large itemsets L(2): 38

Size of set of large itemsets L(3): 29

Size of set of large itemsets L(4): 8

Best rules found:

 1. marital-status=Never-married hours-per-week=20-40 9669 ==> income=<=50K 9368    <conf:(0.97)> lift:(1.29)
 2. workclass=Private marital-status=Never-married 12243 ==> income=<=50K 11755    <conf:(0.96)> lift:(1.28)
 3. workclass=Private marital-status=Never-married race=White 10134 ==> income=<=50K 9702    <conf:(0.96)> li
 4. marital-status=Never-married 14875 ==> income=<=50K 14153    <conf:(0.95)> lift:(1.27) lev:(0.06) [2968]
 5. marital-status=Never-married race=White 12228 ==> income=<=50K 11590    <conf:(0.95)> lift:(1.26) lev:(0.
 6. gender=Male hours-per-week=40-60 10122 ==> race=White 9388    <conf:(0.93)> lift:(1.08) lev:(0.02) [714]
 7. hours-per-week=40-60 12403 ==> race=White 11366    <conf:(0.92)> lift:(1.07) lev:(0.02) [738] conv:(1.71)
 8. age=20-30 11487 ==> income=<=50K 10513    <conf:(0.92)> lift:(1.22) lev:(0.04) [1876] conv:(2.92)
 9. income=>50K 11422 ==> race=White 10367    <conf:(0.91)> lift:(1.06) lev:(0.01) [579] conv:(1.55)
10. marital-status=Married-civ-spouse race=White income=<=50K 10343 ==> gender=Male 9378    <conf:(0.91)> lif
```

➔ Minimum Support = 0.2

造成原因:



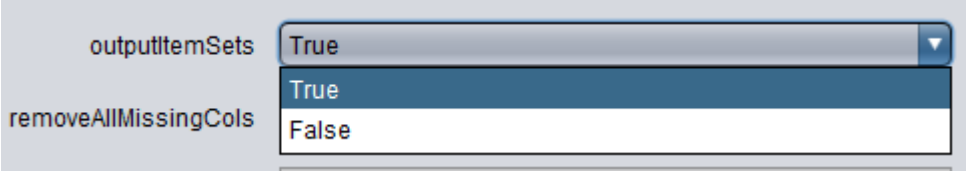
➔ 由上圖可知，我們設定最低 Support = 0.2，並從 Support = 1 開始找 Rule，若沒有找到每次 Support 就 – 0.05 (delta)。

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)

```
Minimum support: 0.2 (9207 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 15

Size of set of large itemsets L(2): 38

Size of set of large itemsets L(3): 29

Size of set of large itemsets L(4): 8

Best rules found:

 1. marital-status=Never-married hours-per-week=20-40 9669 ==> income=<=50K 9368    <conf:(0.97)> lift:(1.29)
 2. workclass=Private marital-status=Never-married 12243 ==> income=<=50K 11755    <conf:(0.96)> lift:(1.28)
 3. workclass=Private marital-status=Never-married race=White 10134 ==> income=<=50K 9702    <conf:(0.96)> li
 4. marital-status=Never-married 14875 ==> income=<=50K 14153    <conf:(0.95)> lift:(1.27) lev:(0.06) [2968]
 5. marital-status=Never-married race=White 12228 ==> income=<=50K 11590    <conf:(0.95)> lift:(1.26) lev:(0.
 6. gender=Male hours-per-week=40-60 10122 ==> race=White 9388    <conf:(0.93)> lift:(1.08) lev:(0.02) [714]
 7. hours-per-week=40-60 12403 ==> race=White 11366    <conf:(0.92)> lift:(1.07) lev:(0.02) [738] conv:(1.71)
 8. age=20-30 11487 ==> income=<=50K 10513    <conf:(0.92)> lift:(1.22) lev:(0.04) [1876] conv:(2.92)
 9. income=>50K 11422 ==> race=White 10367    <conf:(0.91)> lift:(1.06) lev:(0.01) [579] conv:(1.55)
10. marital-status=Married-civ-spouse race=White income=<=50K 10343 ==> gender=Male 9378    <conf:(0.91)> lif
```

(d)



➔ 把 outputItemSets 調成 True

```
Size of set of large itemsets L(1): 15

Large Itemsets L(1):
age=20-30 11487
age=30-40 12538
age=40-50 10182
workclass=Private 33906
education=HS-grad 14972
education=Some-college 10036
marital-status=Never-married 14875
marital-status=Married-civ-spouse 21451
race=White 39444
gender=Male 31114
gender=Female 14919
hours-per-week=20-40 28350
hours-per-week=40-60 12403
income=<=50K 34611
income=>50K 11422

Size of set of large itemsets L(2): 38

Large Itemsets L(2):
age=20-30 workclass=Private 9649
age=20-30 race=White 9650
age=20-30 income=<=50K 10513
age=30-40 workclass=Private 9370
age=30-40 race=White 10636
Size of set of large itemsets L(3): 29

Large Itemsets L(3):
workclass=Private education=HS-grad race=White 9907
workclass=Private education=HS-grad income=<=50K 9983
workclass=Private marital-status=Never-married race=White 10134
workclass=Private marital-status=Never-married income=<=50K 11755
workclass=Private marital-status=Married-civ-spouse race=White 12941
workclass=Private marital-status=Married-civ-spouse gender=Male 12878
workclass=Private race=White gender=Male 19602
workclass=Private race=White gender=Female 9422
workclass=Private race=White hours-per-week=20-40 17985
workclass=Private race=White income=<=50K 22282
workclass=Private gender=Male hours-per-week=20-40 13422
workclass=Private gender=Male income=<=50K 16015
workclass=Private gender=Female income=<=50K 10504
workclass=Private hours-per-week=20-40 income=<=50K 18043
education=HS-grad race=White income=<=50K 10500
marital-status=Never-married race=White income=<=50K 11590
marital-status=Never-married hours-per-week=20-40 income=<=50K 9368
marital-status=Married-civ-spouse race=White gender=Male 17345
marital-status=Married-civ-spouse race=White hours-per-week=20-40 10483
marital-status=Married-civ-spouse race=White income=<=50K 10343
marital-status=Married-civ-spouse gender=Male hours-per-week=20-40 10482
marital-status=Married-civ-spouse gender=Male income=<=50K 10487
race=White gender=Male hours-per-week=20-40 15331
race=White gender=Male hours-per-week=40-60 9388
race=White gender=Male income=<=50K 18529
race=White gender=Female income=<=50K 10548
```


```
Size of set of large itemsets L(2): 38

Large Itemsets L(2):
age=20-30 workclass=Private 9649
age=20-30 race=White 9650
age=20-30 income=<=50K 10513
age=30-40 workclass=Private 9370
age=30-40 race=White 10636
workclass=Private education=HS-grad 11682
workclass=Private marital-status=Never-married 12243
workclass=Private marital-status=Married-civ-spouse 14473
workclass=Private race=White 29024
workclass=Private gender=Male 22307
workclass=Private gender=Female 11599
workclass=Private hours-per-week=20-40 21656
workclass=Private income=<=50K 26519
education=HS-grad race=White 12737
education=HS-grad gender=Male 10251
education=HS-grad hours-per-week=20-40 10123
education=HS-grad income=<=50K 12535
marital-status=Never-married race=White 12228
marital-status=Never-married hours-per-week=20-40 9669
marital-status=Never-married income=<=50K 14153
marital-status=Married-civ-spouse race=White 19229
marital-status=Married-civ-spouse gender=Male 19183
marital-status=Married-civ-spouse hours-per-week=20-40 12062
marital-status=Married-civ-spouse income=<=50K 11705
marital-status=Married-civ-spouse income=>50K 9746
race=White gender=Male 27421
```

```
Size of set of large itemsets L(4): 8

Large Itemsets L(4):
workclass=Private marital-status=Never-married race=White income=<=50K 9702
workclass=Private marital-status=Married-civ-spouse race=White gender=Male 11625
workclass=Private race=White gender=Male hours-per-week=20-40 11463
workclass=Private race=White gender=Male income=<=50K 13829
workclass=Private race=White hours-per-week=20-40 income=<=50K 14774
workclass=Private gender=Male hours-per-week=20-40 income=<=50K 10479
marital-status=Married-civ-spouse race=White gender=Male income=<=50K 9378
race=White gender=Male hours-per-week=20-40 income=<=50K 11345
```

Q2.

(e)

```
import pandas as pd
df = pd.read_csv('adult.csv')
df
```

	age	workclass	education	marital-status	occupation	race	gender	hours-per-week	income
0	20-30	Private	11th	Never-married	Machine-op-inspct	Black	Male	20-40	<=50K
1	30-40	Private	HS-grad	Married-civ-spouse	Farming-fishing	White	Male	40-60	<=50K
2	20-30	Local-gov	Assoc-acdm	Married-civ-spouse	Protective-serv	White	Male	20-40	>50K
3	40-50	Private	Some-college	Married-civ-spouse	Machine-op-inspct	Black	Male	20-40	>50K
4	30-40	Private	10th	Never-married	Other-service	White	Male	20-40	<=50K
...	...	...	...	...	...	...	...	...	...
46028	20-30	Private	Assoc-acdm	Married-civ-spouse	Tech-support	White	Female	20-40	<=50K
46029	30-40	Private	HS-grad	Married-civ-spouse	Machine-op-inspct	White	Male	20-40	>50K
46030	50-60	Private	HS-grad	Widowed	Adm-clerical	White	Female	20-40	<=50K
46031	20-30	Private	HS-grad	Never-married	Adm-clerical	White	Male	0-20	<=50K
46032	50-60	Self-emp-inc	HS-grad	Married-civ-spouse	Exec-managerial	White	Female	20-40	>50K

46033 rows × 9 columns

➔ 讀檔，並看資料大致樣貌

```
In [2]: df = df.astype(str) # 確保所有型態都為string,避免到時候有型態轉換的問題
        data = df.values.tolist()
```

➔ 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

```
from apyori import apriori

#建立rule, 設定參數
#變成list
rules = list(apriori(data, min_support= 0.2, min_confidence= 0.9))
rules
```

➔ 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。

```
# 查看細部結構
print(rules[0], "\n")
rule_len = len(rules[0]) # 得知rule結構
for i in range(rule_len):
    print(rules[0][i])
```

```
RelationRecord(items=frozenset({'20-30', '<=50K'}), support=0.22837964069254665, ordered_statistics=[OrderedStatistic(items_bas
e=frozenset({'20-30'}), items_add=frozenset({'<=50K'}), confidence=0.9152084965613302, lift=1.2172370842277807)])

frozenset({'20-30', '<=50K'})
0.22837964069254665
[OrderedStatistic(items_base=frozenset({'20-30'}), items_add=frozenset({'<=50K'}), confidence=0.9152084965613302, lift=1.217237
0842277807)]
```

➔ 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。

```
result = pd.DataFrame()

for item in rules:
    series = pd.Series({"Rule":item[0],"Support":item[1],"Confidence":item[2][0][2]})
    result = result.append(series, ignore_index=True)
```

➔ 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。

```
result.sort_values(by= ['Confidence'], ascending=False)
```

	Confidence	Rule	Support
5	0.968870	(20-40, Never-married, <=50K)	0.203506
8	0.960140	(Private, Never-married, <=50K)	0.255360
12	0.957371	(Private, Never-married, <=50K, White)	0.210762
2	0.951462	(Never-married, <=50K)	0.307453
9	0.947825	(White, Never-married, <=50K)	0.251776
6	0.927485	(White, 40-60, Male)	0.203941
1	0.916391	(White, 40-60)	0.246910
0	0.915208	(20-30, <=50K)	0.228380
3	0.907634	(White, >50K)	0.225208
11	0.906700	(White, Married-civ-spouse, <=50K, Male)	0.203723
7	0.905595	(Female, Private, <=50K)	0.228184
10	0.904186	(White, Married-civ-spouse, Male)	0.376795
13	0.902702	(Private, Married-civ-spouse, Male, White)	0.252536
4	0.901605	(Female, 20-40, <=50K)	0.203832

➔ 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)

```
# for (f)小題
result = result.sort_values(by= ['Confidence'], ascending=False)
result = result.iloc[0:5,:]
result
```

	Confidence	Rule	Support
5	0.968870	(20-40, Never-married, <=50K)	0.203506
8	0.960140	(Private, Never-married, <=50K)	0.255360
12	0.957371	(Private, Never-married, <=50K, White)	0.210762
2	0.951462	(Never-married, <=50K)	0.307453
9	0.947825	(White, Never-married, <=50K)	0.251776

➔ 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:

```
Minimum support: 0.2 (9207 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 15

Size of set of large itemsets L(2): 38

Size of set of large itemsets L(3): 29

Size of set of large itemsets L(4): 8

Best rules found:

 1. marital-status=Never-married hours-per-week=20-40 9669 ==> income=<=50K 9368    <conf:(0.97)> lift:(1.29)
 2. workclass=Private marital-status=Never-married 12243 ==> income=<=50K 11755    <conf:(0.96)> lift:(1.28)
 3. workclass=Private marital-status=Never-married race=White 10134 ==> income=<=50K 9702    <conf:(0.96)> li
 4. marital-status=Never-married 14875 ==> income=<=50K 14153    <conf:(0.95)> lift:(1.27) lev:(0.06) [2968]
 5. marital-status=Never-married race=White 12228 ==> income=<=50K 11590    <conf:(0.95)> lift:(1.26) lev:(0.
 6. gender=Male hours-per-week=40-60 10122 ==> race=White 9388    <conf:(0.93)> lift:(1.08) lev:(0.02) [714]
 7. hours-per-week=40-60 12403 ==> race=White 11366    <conf:(0.92)> lift:(1.07) lev:(0.02) [738] conv:(1.71)
 8. age=20-30 11487 ==> income=<=50K 10513    <conf:(0.92)> lift:(1.22) lev:(0.04) [1876] conv:(2.92)
 9. income=>50K 11422 ==> race=White 10367    <conf:(0.91)> lift:(1.06) lev:(0.01) [579] conv:(1.55)
10. marital-status=Married-civ-spouse race=White income=<=50K 10343 ==> gender=Male 9378    <conf:(0.91)> lif
```

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。