HW2 Association Rules ¶

`NCU MIS ECT  106403551  呂晟維`

1. 使用weka回答以下問題：
    - (a)請嘗試著修改 adult.csv 的欄位與上圖相同，並轉換成 arff 檔使 weka可以執行 Association Rule，請說明使用方法以及解釋原來的檔案不能執行的原因 ?(10%)
    - (b) 請將 numRules 設成 5 和 10，其各別執行後的 Minimum support 為何，請比較兩者並說明造成其差異的原因。(15%)
    - (c) 將 numRule 設成 10，列出前 5 條 rule(15%)
    - (d) 如何在 Associator output 產生 Itemset，請截圖說明並附上 Itemset 結果。(15%)
2. 使用python回答以下問題：
    - (e) 使用已修改過的 adult.csv 檔，使用 Apriori 演算法進行分析,設定 confidence = 0.9、minimum support = 0.2，過程中對所有重要程式步驟進行截圖並加以說明，越詳盡越好。(15%)
    - (f) 產生與 (c) 小題一樣的結果，列出前五條best rules，截圖並加以說明(15%)

1. Weka

參考資料：[以Weka對資料集進行關聯式規則之實作 (https://medium.com/@bt2011aa/%E4%BB%A5weka%E5%B0%8D%E8%B3%87%E6%96%99%E9%9B%86%Ee7a87c2005a9)](https://medium.com/@bt2011aa/%E4%BB%A5weka%E5%B0%8D%E8%B3%87%E6%96%99%E9%9B%86%Ee7a87c2005a9)、[Mining Association Rule with WEKA Explorer (https://storm.cis.fordham.edu/~yli/documents/CISC4631Spring16/Weka_LabTwo.pdf)](https://storm.cis.fordham.edu/~yli/documents/CISC4631Spring16/Weka_LabTwo.pdf)

(a) 請嘗試著修改 adult.csv 的欄位與上圖相同，並轉換成 arff 檔使 weka可以執行 Association Rule，請說明使用方法以及解釋原來的檔案不能執行的原因? (10%)

純Python作法

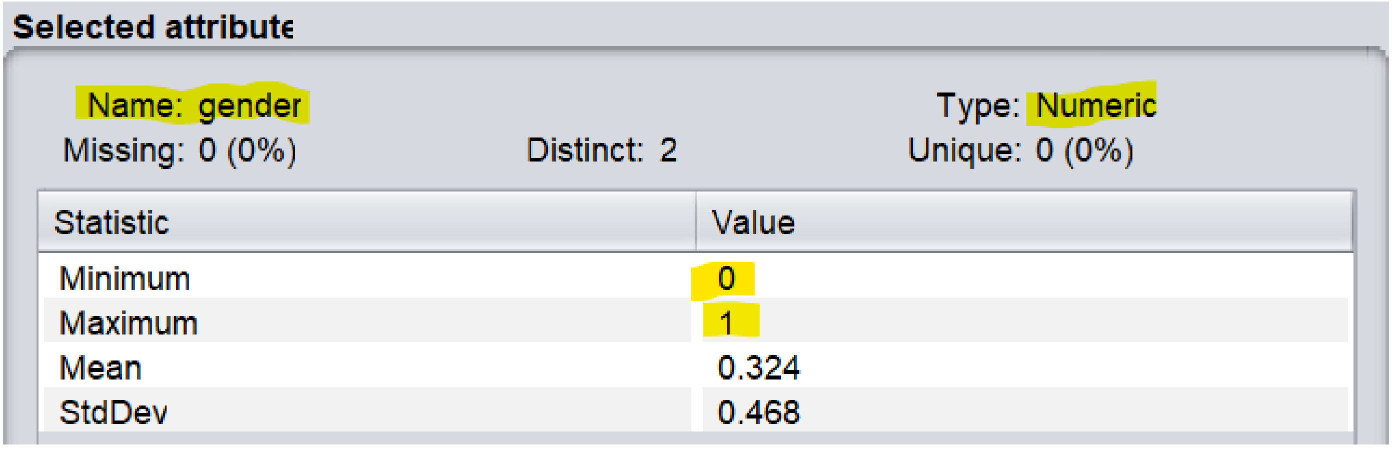
我也手刻一隻python程式 `Csv to Arff.ipynb` 可以將.csv轉換成.arff，因為考量通用性，所有的attribute都會是nominal型態。沒有提供改名功能。

純Weka作法

一開始我們只有adult.csv檔案，匯入時發現Association Rules的start鍵按不下去，因為csv裡面gender的欄位資料只有0和1；0代表Male，1代表Female，但是weka判斷此欄位為numeric型態，所以執行不了。

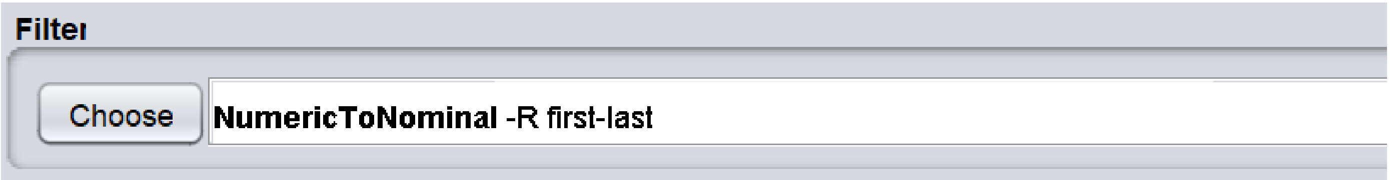
Relation: adult

No.	1: age Nominal	2: workclass Nominal	3: education Nominal	4: marital-status Nominal	5: occupation Nominal	6: race Nominal	7: gender Numeric	8: hours-per-week Nominal	9: income Nominal
1	20-30	Private	11th	Never-married	Machine-o...	Black	0.0	20-40	(=50K
2	30-40	Private	HS-grad	Married-civ-s...	Farming-fi...	White	0.0	40-60	(=50K
3	20-30	Local-gov	Assoc-ac...	Married-civ-s...	Protective-...	White	0.0	20-40	)50K
4	40-50	Private	Some-col...	Married-civ-s...	Machine-o...	Black	0.0	20-40	)50K



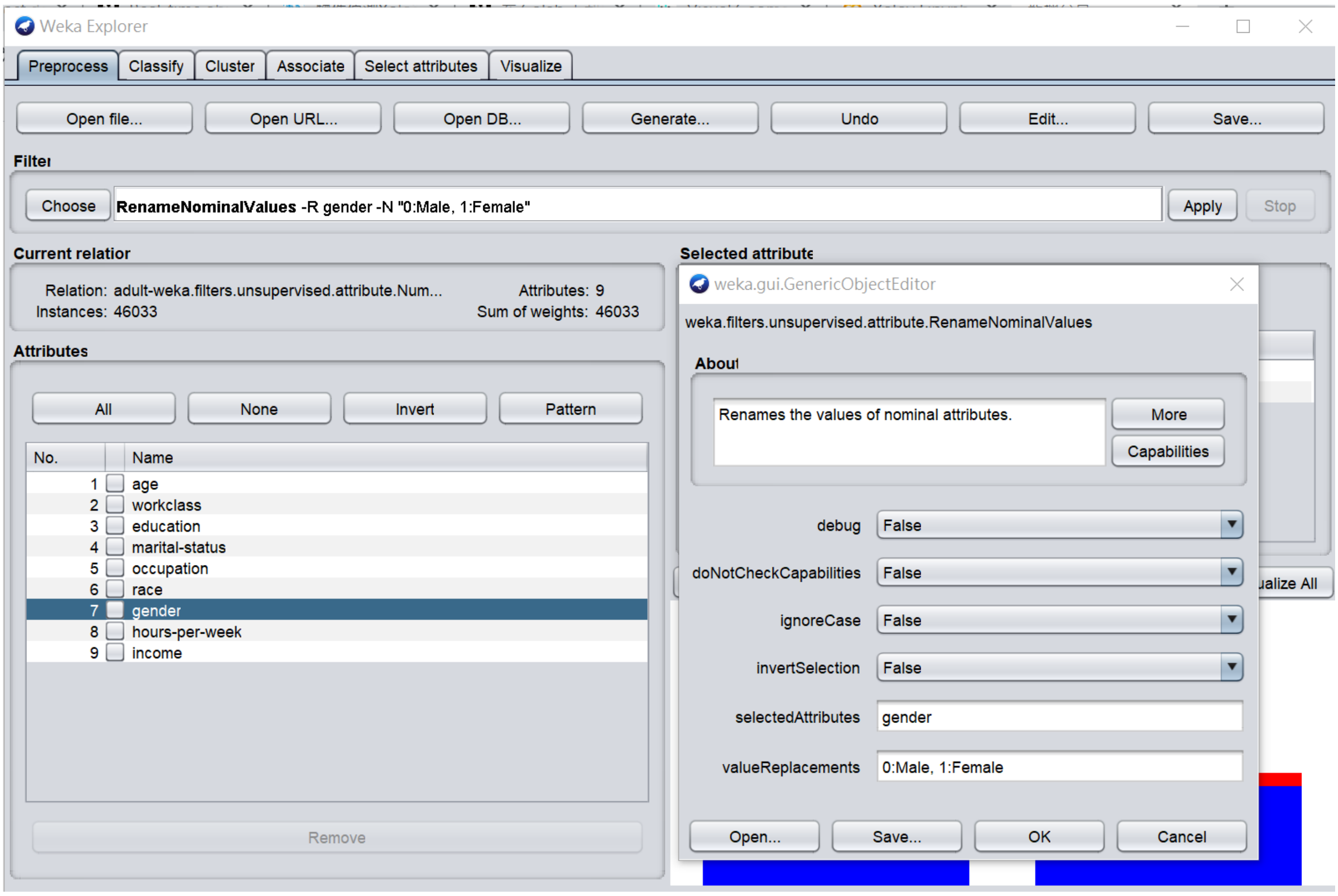
因此需要先將numeric型態，轉成nominal型態。Weka有內鍵功能來轉換attribute的型態，我們先選取gender欄位，點選filter > nonsupervise > numeric to nominal按下執行即可。如此一來就可將0和1轉成nominal的形式。

```
@attribute gender {0, 1}
```



接下來，由於gender的值還是0和1，雖然可以run但可讀性還是差了一些，所以接下來我們再把0和1改名成Male和Female，選取filter > nonsupervise > rename nominal values。按照下圖輸入參數。最後再save成.arff檔即可。

```
@attribute gender {Male, Female}
```



轉檔後的結果

Relation: adult-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.RenameN

No.	1: age Nominal	2: workclass Nominal	3: education Nominal	4: marital-status Nominal	5: occupation Nominal	6: race Nominal	7: gender Nominal	8: hours-per-week Nominal	9: income Nominal
1	20-30	Private	11th	Never-married	Machine-o...	Black	Male	20-40	(=50K
2	30-40	Private	HS-grad	Married-civ-s...	Farming-fi...	White	Male	40-60	(=50K
3	20-30	Local-gov	Assoc-ac...	Married-civ-s...	Protective-...	White	Male	20-40	)50K
4	40-50	Private	Some-col...	Married-civ-s...	Machine-o...	Black	Male	20-40	)50K

(b) 請將 numRules 設成 5 和 10，其各別執行後的 Minimum support 為何，請比較兩者並說明造成其差異的原因。(15%)

numRules設成5的結果：Minimum support為0.25

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

numRules設成10的結果：Minimum support為0.2

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

解讀一下這份報表，weka一開始從support = 1找confidence > 0.9的rules。演算法一共重複執行了Number of cycles performed次才停下來，停下來的情形有2種，一是找到指定數目筆的rules(5或10個)，二是Minimum support下降至指定值(0.2)時還未找齊rules。

當numrule = 5時，演算法重複執行了15次便集齊5個rules了(大於等於5個)，第15次的Minimum support是0.25；而要再進入第16次cycle才能找到10個rules，Minimum support為0.25 - 0.05 = 0.2。

另外由於rules的排序是照confidence的大小來排，因此每條規則的rank會不太一樣，像是rules5的第一條規則在rule10中是第二條。

- **Rule 5**的全部截圖

```
Best rules found:

1. workclass=Private marital-status=Never-married 12243 ==> income=<=50K 11755    <conf:(0.96)> lift:(1.28) lev:
2. marital-status=Never-married 14875 ==> income=<=50K 14153    <conf:(0.95)> lift:(1.27) lev:(0.06) [2968] conv
3. marital-status=Never-married race=White 12228 ==> income=<=50K 11590    <conf:(0.95)> lift:(1.26) lev:(0.05)
4. marital-status=Married-civ-spouse gender=0 19183 ==> race=White 17345    <conf:(0.9)> lift:(1.06) lev:(0.02)
5. workclass=Private marital-status=Married-civ-spouse gender=0 12878 ==> race=White 11625    <conf:(0.9)> lift:
```

- **Rule 10**的全部截圖

```
Best rules found:

 1. marital-status=Never-married hours-per-week=20-40 9669 ==> income=<=50K 9368    <conf:(0.97)> lift:(1.29) le
 2. workclass=Private marital-status=Never-married 12243 ==> income=<=50K 11755    <conf:(0.96)> lift:(1.28) lev
 3. workclass=Private marital-status=Never-married race=White 10134 ==> income=<=50K 9702    <conf:(0.96)> lift:
 4. marital-status=Never-married 14875 ==> income=<=50K 14153    <conf:(0.95)> lift:(1.27) lev:(0.06) [2968] con
 5. marital-status=Never-married race=White 12228 ==> income=<=50K 11590    <conf:(0.95)> lift:(1.26) lev:(0.05)
 6. gender=0 hours-per-week=40-60 10122 ==> race=White 9388    <conf:(0.93)> lift:(1.08) lev:(0.02) [714] conv:(
 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=0 9378    <conf:(0.91)> lift:(1.3
```

(c) 將 numRule 設成 10，列出前 5 條 rule(15%)

- 這5個rules的lift值(提升度)都 > 1，Conviction(conv確信度)值也很大，代表前因和後果之間的關係相當密切。
- 這5個rules的support count也都在10000筆資料上下，在46000筆樣本中算是多數了，因此結果值得信賴。

Rules如下：

1. marital-status=Never-married hours-per-week=20-40 9669 ==> income= <=50K 9368 conf:(0.97) lift:(1.29) lev:(0.05) [2098] conv:(7.94)

    → 沒有結婚 ∩ 工時20~40小時的人，有0.97的比例，他們的收入小於50K

2. workclass=Private marital-status=Never-married 12243 ==> income= <=50K 11755 conf:(0.96) lift:(1.28) lev:(0.06) [2549] conv:(6.21)

    → 在私人機構工作 ∩ 沒有結婚的人，有0.96的比例，他們的收入小於50K

3. workclass=Private marital-status=Never-married race=White 10134 ==> income= <=50K 9702 conf:(0.96) lift:(1.27) lev:(0.05) [2082] conv:(5.81)

    → 在私人機構工作 ∩ 沒有結婚 ∩ 是白人，有0.96的比例，他們的收入小於50K

4. marital-status=Never-married 14875 ==> income= <=50K 14153 conf:(0.95) lift:(1.27) lev:(0.06) [2968] conv:(5.1)

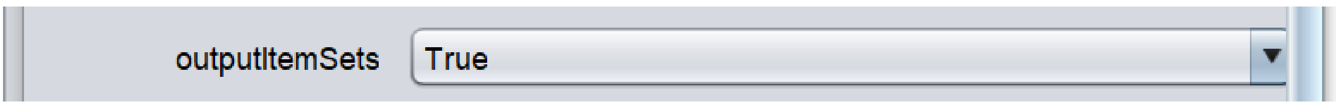
    → 沒有結婚的人，有0.95的比例，他們的收入小於50K

5. marital-status=Never-married race=White 12228 ==> income= <=50K 11590 conf:(0.95) lift:(1.26) lev:(0.05) [2396] conv:(4.75)

    → 沒有結婚 ∩ 是白人，有0.95的比例，他們的收入小於50K

(d) 如何在 Associator output 產生 Itemset，請截圖說明並附上 Itemset 結果。(15%)

將output itemset的選項設成True即可看到frequent itemset的output。Itemset會依support值高到低(從1.0開始)尋找support值 >= min_sup 的集合，每個循環會從一個元素且符合min_sup的集合找起，再找兩個元素的集合、三個元素的集合、四個元素的集合...。再依每種集合(itemsets)的confidence來找尋規則，confidence即是條件機率的意思，當confidence >= 0.9便是一條rule。



Rule10的輸出如下：

```
Generated sets of large itemsets:

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
workclass=Private education=HS-grad 11682
...

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
...

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
...
```

2. 使用Python回答以下問題：

(e) 使用已修改過的 **adult.csv** 檔，使用 **Apriori** 演算法進行分析,設定 **confidence =**

0.9、minimum support = 0.2，過程中對所有重要程式步驟進行截圖並加以說明，越詳盡越好。(15%)

可以直接跳至程式的.ipynb，有列出所有的參考資料和詳盡程式解說。

我們程式分成2步驟，資料前處理和演算法執行，最後我們會解析結果：

- 整理資料

利用pandas讀csv檔，apriori的輸入和輸出都是list型態的資料結構，所以把df轉成list。

Association rules

- Rules的解析法 Python Apriori (apyori library) 實戰篇
- Result項目的含意 result裡每一個項集的屬性介紹
- 超級完整的英文逐步程式範例 Association Rule Mining via Apriori Algorithm in Python

整理資料

In [1]:
```
import pandas as pd
df = pd.read_csv('adult.csv')
df
```

Out[1]:

	age	workclass	education	marital-status	occupation	race	gender	hours-per-week	income
0	20-30	Private	11th	Never-married	Machine-op-inspct	Black	0	20-40	<=50K
1	30-40	Private	HS-grad	Married-civ-spouse	Farming-fishing	White	0	40-60	<=50K
2	20-30	Local-gov	Assoc-acdm	Married-civ-spouse	Protective-serv	White	0	20-40	>50K
3	40-50	Private	Some-college	Married-civ-spouse	Machine-op-inspct	Black	0	20-40	>50K
4	30-40	Private	10th	Never-married	Other-service	White	0	20-40	<=50K

由於 Apriori library we are going to use requires our dataset to be in the form of a list of lists. Currently we have data in the form of a pandas dataframe. To convert our pandas dataframe into a list of lists, execute the following script:

In [2]:
```
df = df.astype(str)
data = df.values.tolist()
```

- apriori演算法

重點程式碼是apriori方法，輸入值有min_support= 0.2, min_confidence= 0.9。

```
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))
```

apriori演算法

作業規定 min_support= 0.2, min_confidence= 0.9

In [3]:
```
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
```

Out[3]:
```
[RelationRecord(items=frozenset({'<=50K', '20-30'}), support=0.22837964069254665, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'20-30'}), items_add=frozenset({'<=50K'}), confidence=0.91520
84965613302, lift=1.2172370842277807)]),
 RelationRecord(items=frozenset({'40-60', 'White'}), support=0.2469098255599244, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'40-60'}), items_add=frozenset({'White'}), confidence=0.91639
11956784649, lift=1.0694715523442546)]),
 RelationRecord(items=frozenset({'<=50K', 'Never-married'}), support=0.3074533486846393, ordered_stat
istics=[OrderedStatistic(items_base=frozenset({'Never-married'}), items_add=frozenset({'<=50K'}), con
```

Rules

- 解析結果

根據以下屬性可以整理出得出的規則們

```
- items_base - 關聯規則中的分母項集 (ex買了啤酒)
- items_add -  關聯規則中的分子項集 (ex跟著買尿布)
```

In [5]:
```python
result = pd.DataFrame()
for item in rules:
    item_base = item[2][0][0] # 分母
    item_add  = item[2][0][1] # 分子
    base = set([x for x in item_base])
    add  = set([x for x in item_add])
    series = pd.Series({"Rule":f"{base}->{add}","Support":item[1],"Confidence":item[2][0][2]})
    result = result.append(series, ignore_index=True)
```

In [6]:
```python
result.sort_values(by= ['Confidence'], ascending=False)
```

Out[6]:

	Confidence	Rule	Support
8	0.968870	{'Never-married', '20-40'}->{'<=50K'}	0.203506
9	0.960140	{'Never-married', 'Private'}->{'<=50K'}	0.255360
13	0.957371	{'Never-married', 'Private', 'White'}->{'<=50K'}	0.210762
2	0.951462	{'Never-married'}->{'<=50K'}	0.307453
10	0.947825	{'Never-married', 'White'}->{'<=50K'}	0.251776
4	0.927485	{'40-60', '0'}->{'White'}	0.203941
1	0.916391	{'40-60'}->{'White'}	0.246910

(f) 產生與 (c) 小題一樣的結果，列出前五條best rules，截圖並加以說明 (15%)

我們發現python程式的結果和weka的結果一模一樣，第一條rule的confidence = 0.968870 = 9368 / 9669，因為weka只四捨五入到0.97，若把樣本數相除則和python結果相符。關聯的規則是"沒有結婚 ∩ 工時20~40小時的人，有0.97的比例，他們的收入小於50K"。

其他四項不加贅述了。偷偷一提python的apriori產生的rules的結果超級難解讀(´_ゝ`)，規則也都沒有直接列出的因果關係。下表是額外查資料才做出的關係。

In [6]:
```python
result.sort_values(by= ['Confidence'], ascending=False)
```

Out[6]:

	Confidence	Rule	Support
8	0.968870	{'Never-married', '20-40'}->{'<=50K'}	0.203506
9	0.960140	{'Never-married', 'Private'}->{'<=50K'}	0.255360
13	0.957371	{'Never-married', 'Private', 'White'}->{'<=50K'}	0.210762
2	0.951462	{'Never-married'}->{'<=50K'}	0.307453
10	0.947825	{'Never-married', 'White'}->{'<=50K'}	0.251776