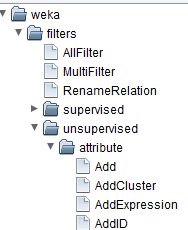
ECT HW7

**Weka Part:**

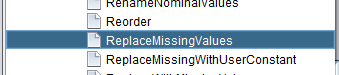
**(a)**

****

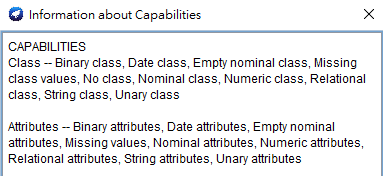
* 找到Filter的地方，點選Choose



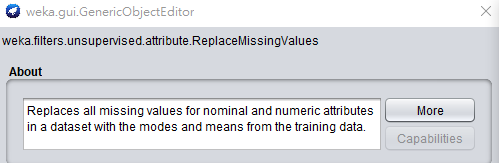
* 找Weka -> filters -> unsupervised -> attribute



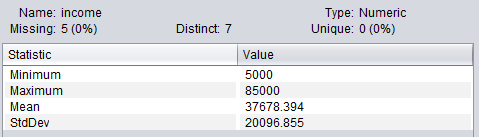
* 其中有一個ReplaceMissingValues

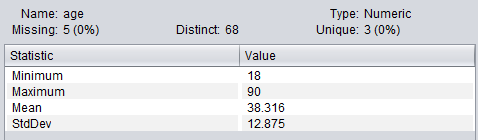


* 查看使用條件，可以處理Missing values和Numeric attribute，因此可用



* 根據使用說明，他會把numeric的missing value用mean填補上去

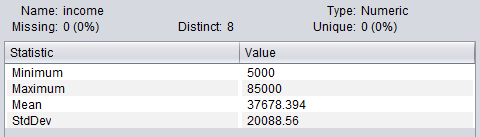


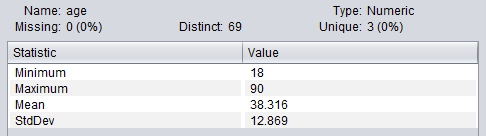


* 如上圖所示，有Missing Value的屬性分別為income、age。根據使用說明預期，用ReplaceMissingValues處理過後，mean並不會改變，因為只是加入一些為value = mean的數據。



* 點擊Apply使用ReplaceMissingValues處理數據



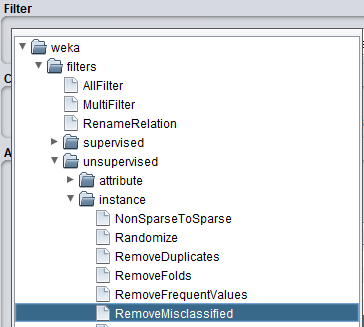


* 如圖所示，Missing的部分已變為0%，且Mean均沒有改變

因此題目所要求的 : 使用ReplaceMissingValue，「列出補上的值」為何？

* 可以合理推測
* 屬性Income填補值 = 37678.394
* 屬性Age填補值 = 38.316

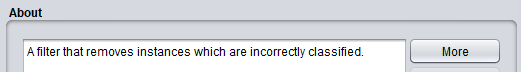
**(b)**



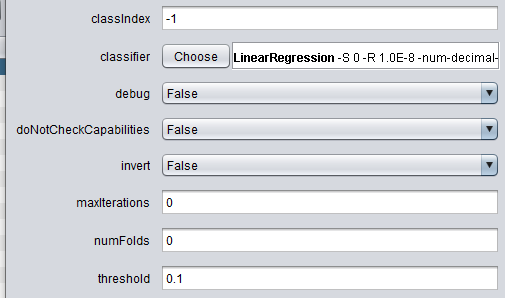
* 從Filter中依照此路徑找到RemoveMisclassified



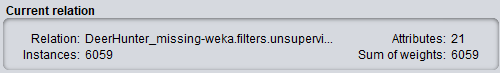
* 查看使用條件，可以處理Numeric的資料。



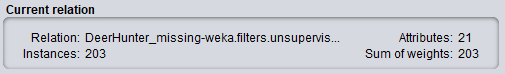
* 查看使用說明，發現他是使用分類器後把分類錯誤的部分當作outlier移除



* 依照題目要求，使用LinearRegression當作分類器，並且用默認設置的classIndex = -1，代表使用最後一個屬性當作class label，在此為「yes」屬性。Threshold代表對numeric數據進行處理時，所可以容忍的誤差值，在此也是使用默認的0.1

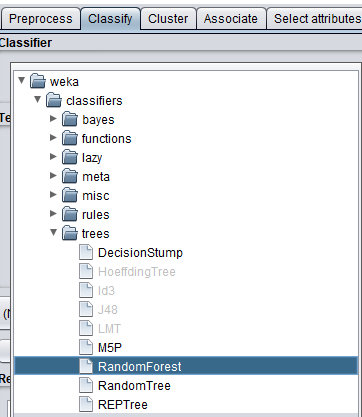


* 可以看到分類前有6059個instance

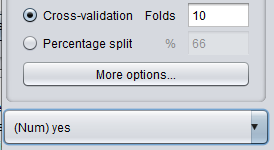


* 分類後只剩下203個instance在0.1的誤差之內。

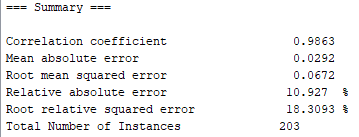
**(c)**



* 切換至「Classify」並找到題目要求的「RandomForest」

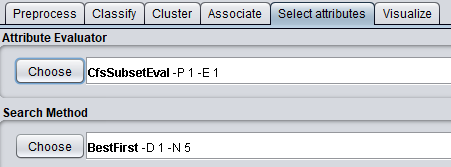


* 根據題目要求，使用10-Fold Cross-validation，class label = yse

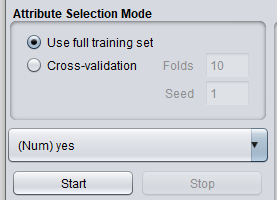


* 可以看到分類結果如上圖所示

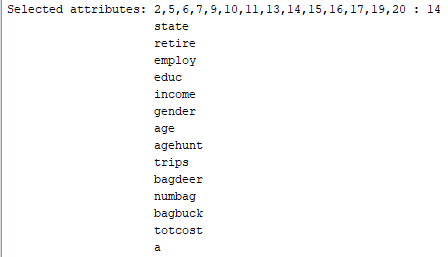
**(d)**



* 切換至Select Attributes，依照題目要求Attributes Evaluator選擇「CfsSubsetEval」、Search Method選擇「BestFirst」

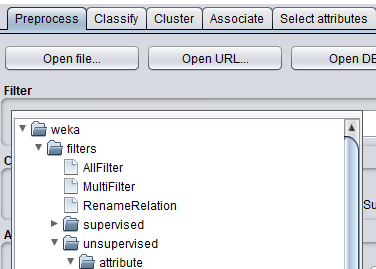


* 雖然 (c) 小題有要求使用Cross-Validation，但此題並未提及，因此直接使用默認的設定「Use full training set」+「class label = yes」。

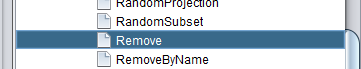


* 最後篩選出了14個attributes，少了1,3,4,8,12,18的attributes

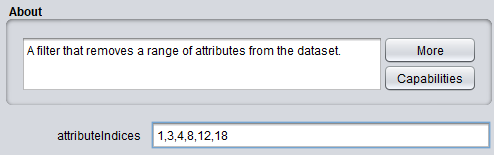
**(e)**



* 回到Preprocess的部分，找一個方法來處理被篩選掉的attributes



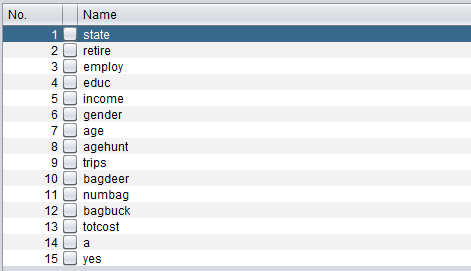
* 選擇其中的Remove



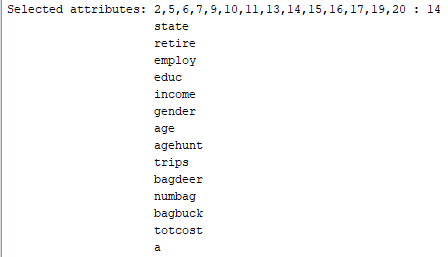
* 根據使用說明，他能幫我們從dataset中刪除不要的attribute，依照 (d) 小題的結果，我們要刪除1,3,4,8,12,18的共6個attributes，因此第一個參數部分就是這些Indices



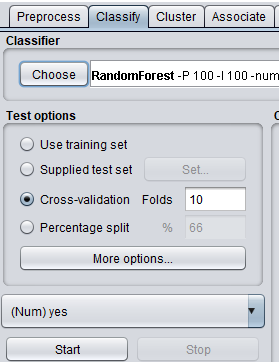
* 點擊Apply開始篩選



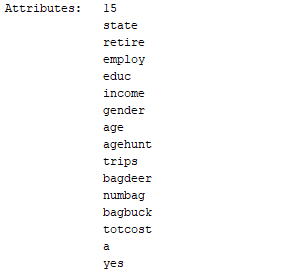
* 篩選結果如上圖所示



* 跟 (d) 小題的結果對照一致

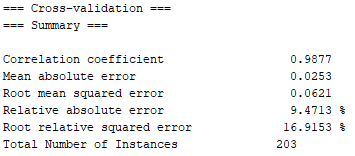


* 回到Classify並且不更改任何設定，直接點擊Start

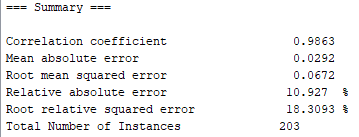


* 可看到現在只剩下15個屬性「14」個input attributes +「1」個output attribute

Attribute Selection之後:

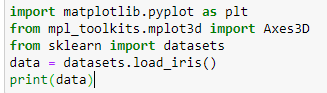
* 

Attribute Selection之前:

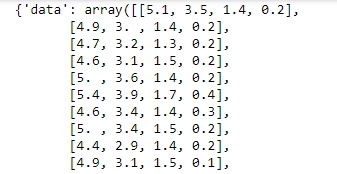
* 
* 可以看出進行Attribute Selection之後，Root mean squared error下降了，相關係數上升了，因此是一個有效的結果。

**Python Part:**

**(a)**

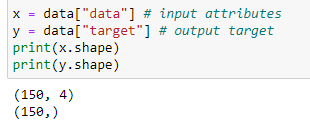


* 先載入IRIS資料集，並印出來看大致上的格式





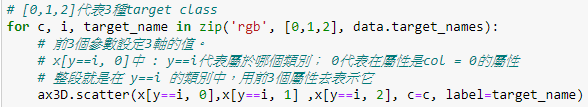
* 可以看出他是一個dictionary的形式，其中「data」代表Input attribute、「target」代表output attribute



* 因此把資料用這兩個key切分成Attribute、Target



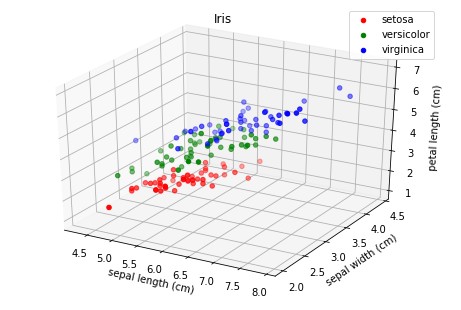
* 用Axes3D初始化3D模型



* 用zip把各個需要的參數包起來，用scatter把散佈圖畫出來。
* Scatter前3個參數分別為在3D空間中3軸的值
* c 代表顏色
* label代表此點之後在圖例上的名字
* 詳細內容可參考圖中的註解



* 做一些基本設定，把圖片該有的標示標清楚



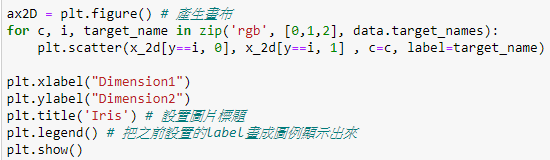
* 畫出來的3D分布圖如上圖所示
* 挑選的3個屬性 :「sepal length、sepal width、petal length」

**(b)**

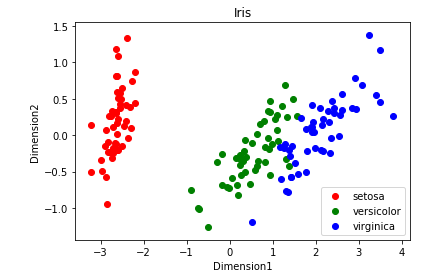


* 使用sklearn.decomposition中的PCA
* n\_components : 代表要降至多少維度
* fit\_transform : 降維並返回結果

**(c)**



* 這次純粹使用「plt」，因為是畫2D圖，方法跟前述相同，只是這次scatter的參數只剩下2軸的值

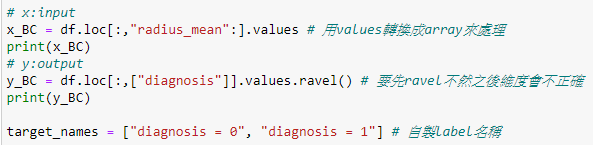


* 結果如圖所示

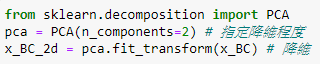
**(d)**



* 先導入資料集

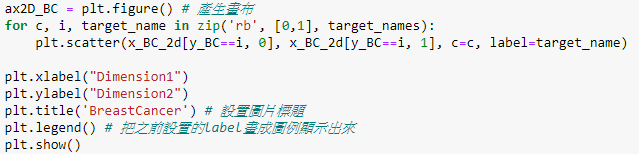


* 把資料集切分成input、output
* target\_names是之後畫圖時用來代表圖例中的label name
* 要使用ravel()展開y的部分，不然之後會報錯因為維度不正確

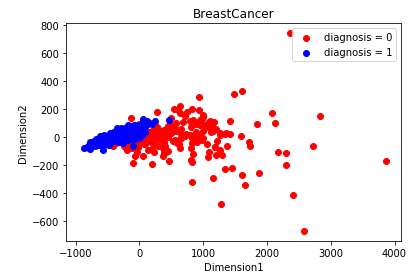


* 把input部分降維至2維，存放至x\_BC\_2d

**(e)**



* 方法跟前述相同，不加贅述



* 結果如圖所示