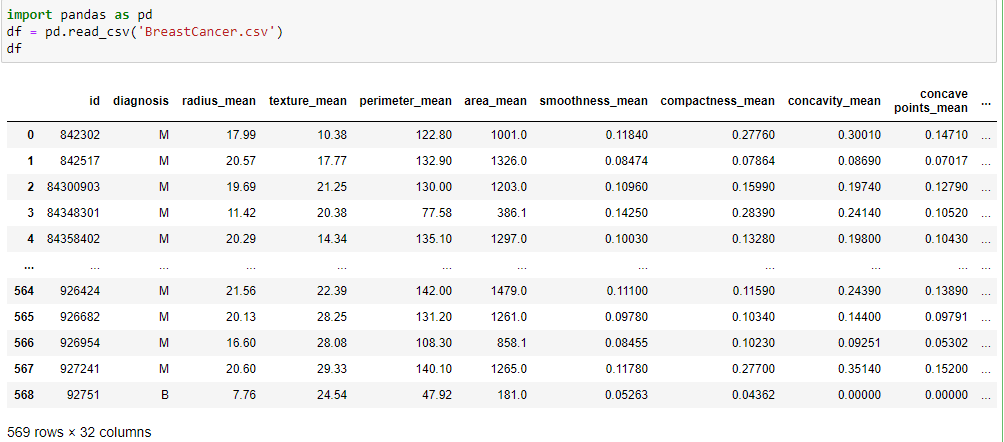
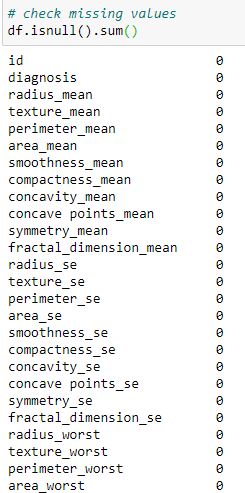
ECT HW5

**Python part:**

前置確認 :

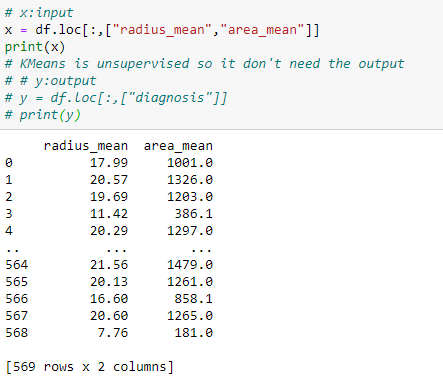


* 先導入資料集



* 確認資料完整性，若有missing values，可能要進行處理

**(a)**



* 因為KMeans是Unsupervised，所以只有input，根據題意把input切為radius\_mean, area\_mean兩部分。

**(b)**



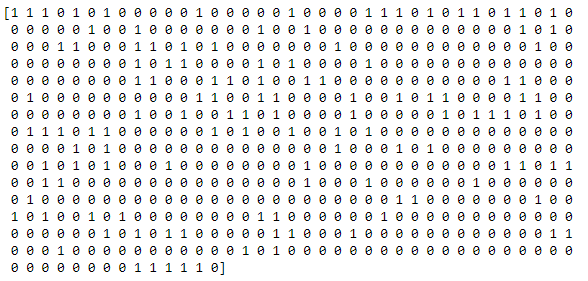
* 先把之後會用到的套件import進來



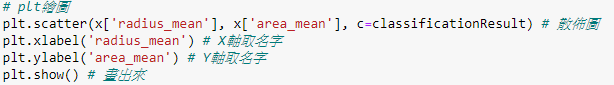
* 初始化Kmeans model時，設定參數n\_cluster = 2, 代表要分成兩群
* 其他像是迭代次數、初始亂數都有參數可以調整，在此沒有特別要求就使用默認參數。

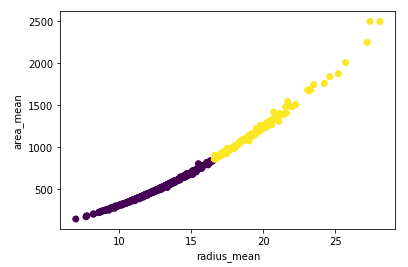
**(c)**



* 使用fit\_predict(x)對x進行分群，會先計算clustering的中心再把每個instance分給各個clustering
* 使用model.labels\_ 把分群結果保存下來
* 
* 可以看到分群完後分成0、1兩群

**(d)**

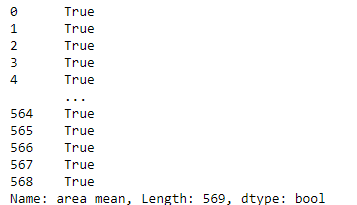




* 使用matplotlib.pyplot中的scatter來繪製散佈圖，前兩個參數為X軸Y軸、最後一個參數「c」代表顏色，我們把剛剛分群結果的「0、1」列表丟進去，讓他繪製不同的顏色。
* Xlabel、ylabel就只是單純地幫兩軸取名字
* Show()把圖畫出來，call過一次後之後不寫也可以畫出來

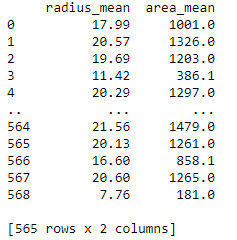
**(e)**



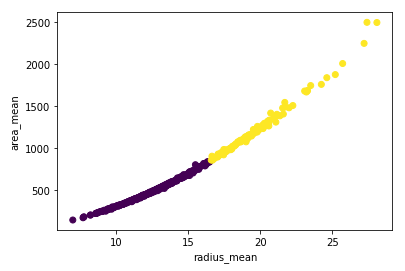


* 先試試看直接用條件切割，會發現values變成True、False，因此要做修正



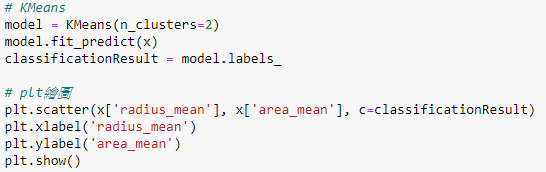


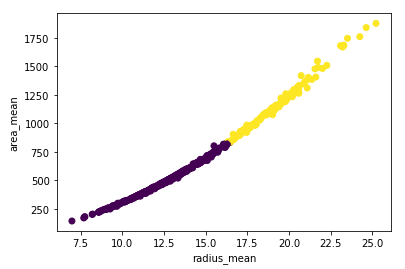
* 用loc()函數把結果為True的保留下來，False的去除
* 可發現處理前有569筆資料，處理後有565筆資料，刪除了4筆



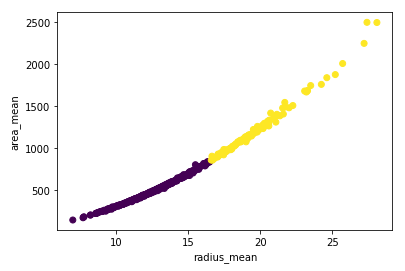
* 而原始圖形中的確有4筆資料在大於2000，的確相符。

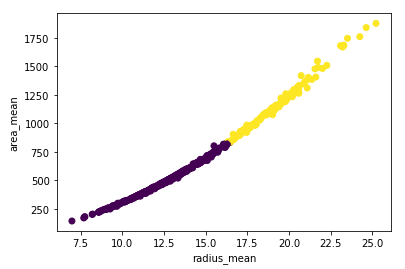
**(f)**





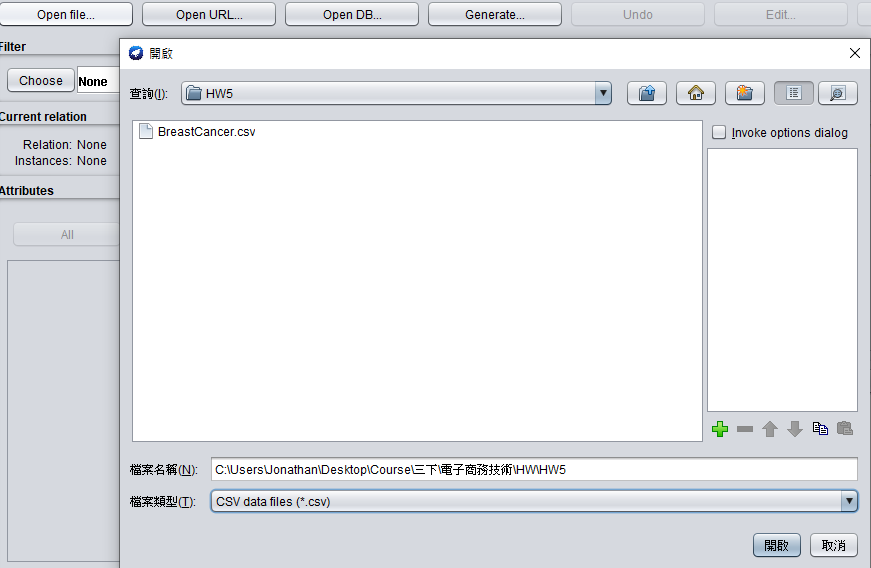
* 重複之前的動作，先用KMeans處理過後，再把結果畫出來。可以注意圖中Y軸已經沒有2000以上的數字了





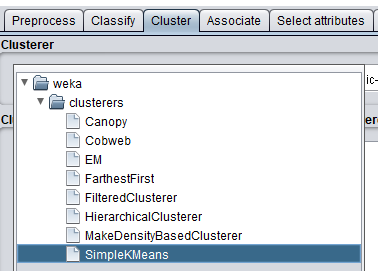
* 比較處理前、處理後的結果，可以看出把2000以上的偏差值處理掉後，clustering的結果更為平均，基本上就是一半一半了。因此可以認為這一步驟是有一定程度的意義。

**Weka part:**



* 先用open file開啟檔案，注意因為這次是給csv，所以檔案類型部分要

改成CSV data files



* 因為KMeans是clusteing的方式，所以去Cluster部分尋找方法使用，依照題目要求選擇SimpleKMeans

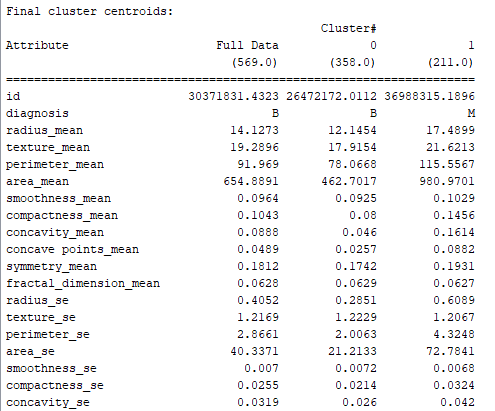


* 依照題意設定Use training set

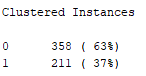


* 依照題意設定numClusters = 2

**(a)**

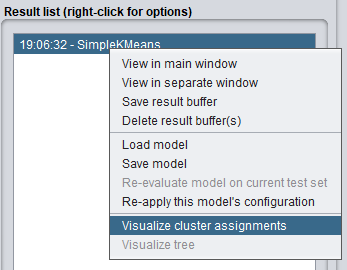


* 比較詳細的結果，可以看出cluster 0有358個instance，cluster 1有211個instance

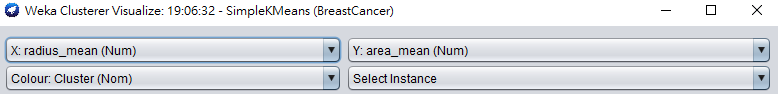


* 比較簡潔的結果，可以看出各自的比例。3
* Hint : 此結果並未把大於2000的instance刪除，因為我觀察到(b)小題要求的也是刪減前的結果。

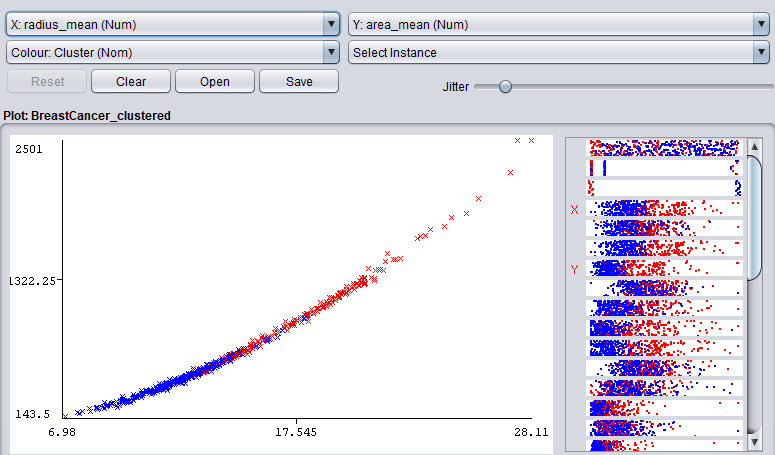
**(b)**



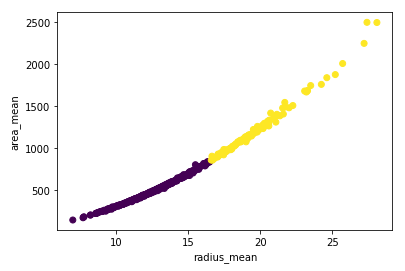
* 在Weka左下較的Result list中，選取此次的分析，用右鍵點取後選擇Visualize cluster assignments

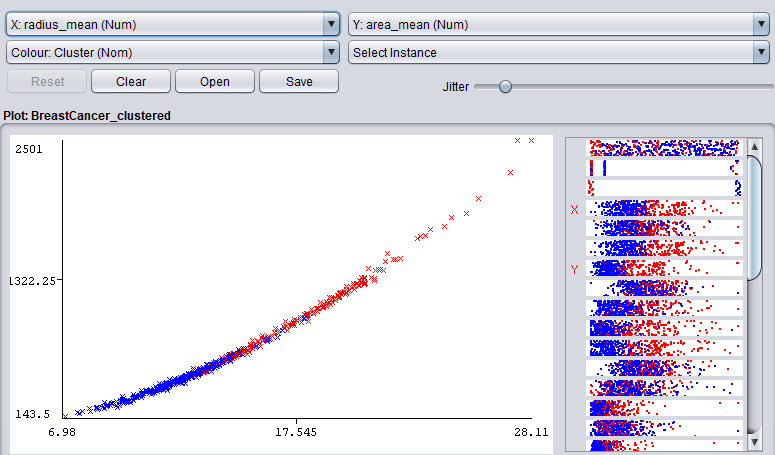


* 最上面兩個參數分別代表X、Y軸，依序調成radius\_mean, area\_mean



* 完成之後即可看到與之前在python分析時相似的結果





* 比較兩圖，的確非常相似。