* 如何執行程式

需在python3環境，執行不須下任何參數，直接於terminal輸入”python3 main.py”即可。執行完畢會將MAP detector的Accuracy rate與Type1 & Type2 & Type3的PCA之variance & variance ratio印在terminal，並且分別秀出二維與三維的PCA散布圖。

* MAP detector說明 (posterior = prior \* likelihood)

首先將Wine.csv內的每個data type各隨機拿出18組放進test.csv，剩下沒有被隨機拿出的資料放在train.csv。接著計算train.csv內每個type所占比例，求得三種type個自的prior probability。並再從train.csv內求得各個type內的13種feature的平均值與標準差。最後就是testing的環節，將test.csv的內容逐列取出，每一橫列就是一筆data，每筆data內有13個feature，分別將這13個feature套入normal distribution的probability density function中:(下面公式中的x就是input feature值，μ與σ分別為該type的該組feature之平均值與標準差)



上述做完後會得到13個f(x)，再將這13個f(x)相乘就可以得到此data對應某種type的likelihood，而該type的prior乘上likelihood即可得posterior probability，這邊我們得到三個posterior probability，將這三個posterior probability相加得marginal probability後，將這三個posterior probability分別除以marginal probability，就可以得到MAP detector識別某筆testing data是某類的機率值，最後我們取最高的機率所對應的type當作最後MAP detector的prediction。全部預測完畢所得到的結果如下: (有設亂樹種子使每次執行程式所產生的train.csv與test.csv都相同)

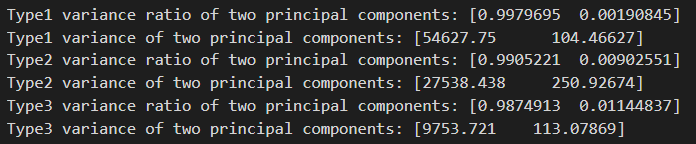


Accuracy rate的算法是將MAP detector預測正確的次數除以testing data總數，並乘上100得趴數。

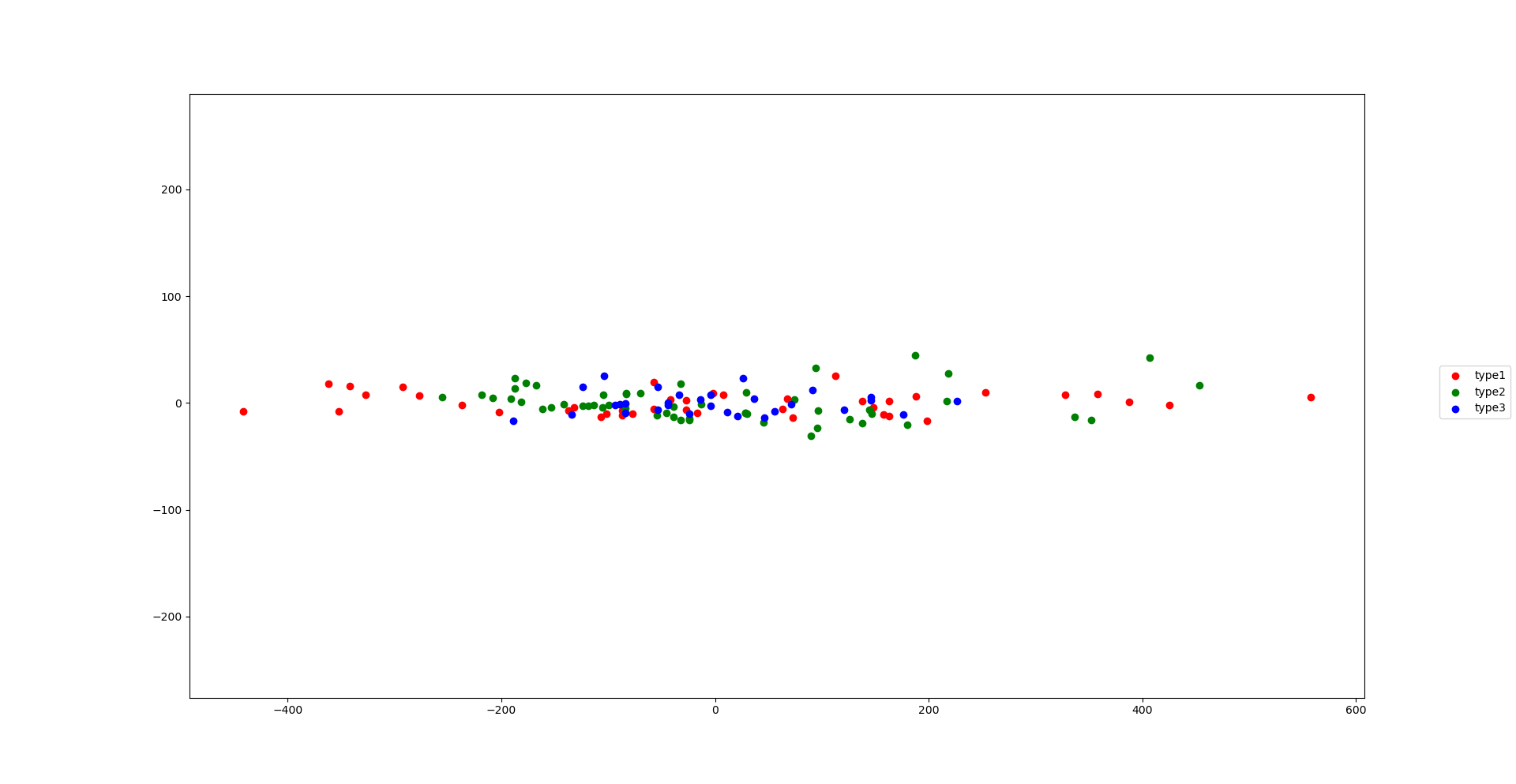
* PCA與視覺呈現

此部分透過sklearn.decomposition.PCA套件實作。我分別做了降成2維與降成3維。

* + 對testing data從原本的13維降至2維

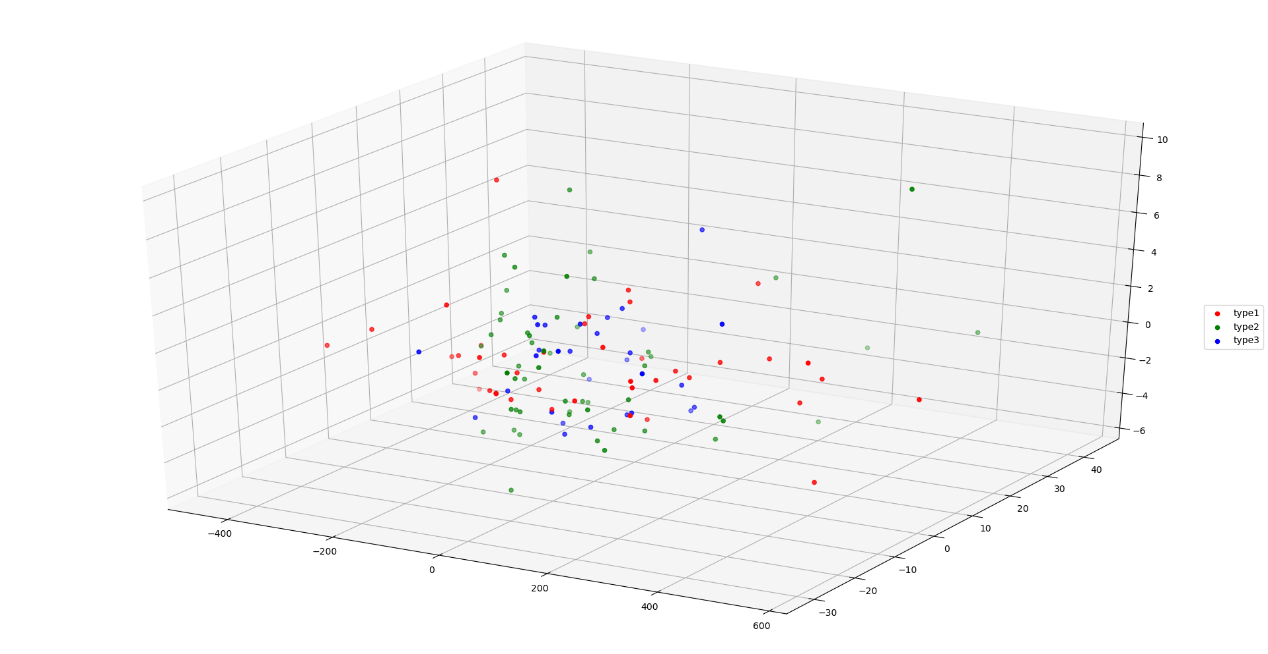


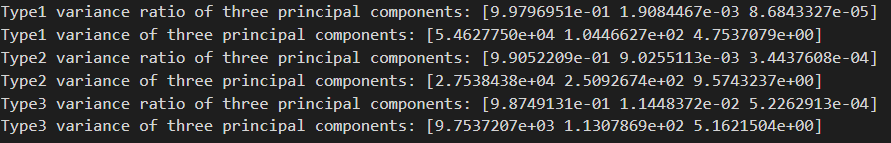
上圖可以發現ratio of first principal component，不論是在type1 or 2 or 3，投影後第一個特徵占了絕大多數的主成分比例。(值都落在0.98到0.99)，換句話說first principal component就已經幾乎做到PCA該有的分類表現了。



上圖是PCA降至2維的分類結果，並且我有將它原本的labeled type用不同顏色做區隔。可以發現構成的點有點接近於直線，這就驗證了我上述提到的，第一個特徵占了絕大多數的主成分比例，也就是第一個特徵已經讓資料的變異數(上圖橫軸看)非常大了，故第二個特徵能呈現出來的變異數(從上圖縱軸看)已經非常小。第一個特徵雖然能用非常不錯的變異程度作投影，但是可以發現其分類表現上不會太好，可想而知第二個特徵投影對分類效果也沒有明顯提升。

* + 對testing data從原本的13維降至3維





上圖可以發現用PCA降至3維的分類效果依然不佳，因為third principal component就只是找能夠讓資料的變異數第三大的投影，而前面兩個principal component已經幾乎包辦了所有主成分，可推得做third principal component意義不大。從上述實驗可說明此testing dataset不適合採用PCA來做分類，其效果遠輸給MAP detector。