- 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 $\sigma$ 分別為該 type 的該組 feature 之平均值與標準差)

  $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

  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)

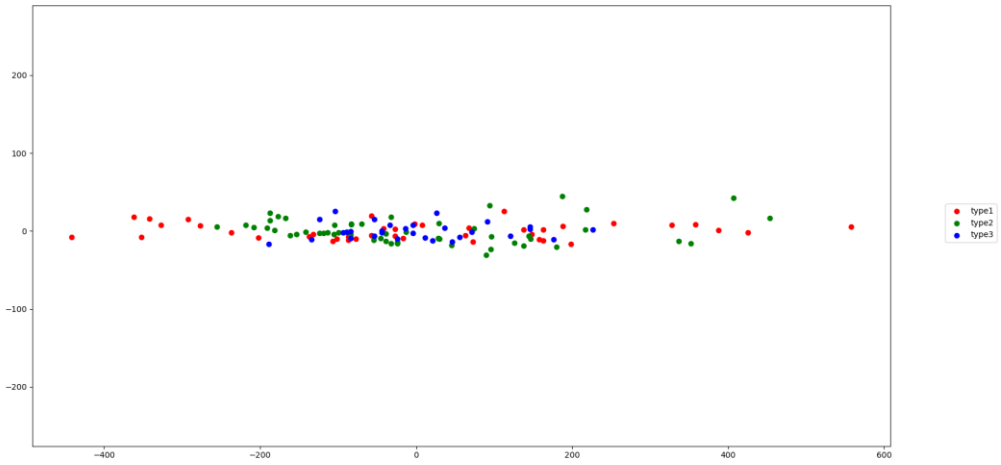
```
[Running] python -u "c:\Users\f6405\Desktop\ML_HW1\main.py"
Accuracy rate of MAP detector: 98.14814814814815 %
```

  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

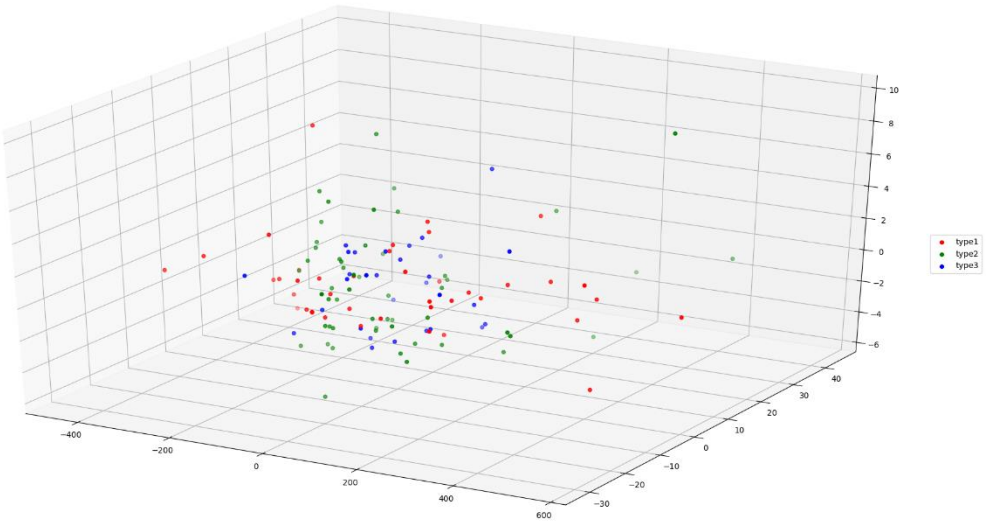
```
Type1 variance ratio of two principal components: [0.9979695  0.00190845]
Type1 variance of two principal components: [54627.75      104.46627]
Type2 variance ratio of two principal components: [0.9905221  0.00902551]
Type2 variance of two principal components: [27538.438     250.92674]
Type3 variance ratio of two principal components: [0.9874913  0.01144837]
Type3 variance of two principal components: [9753.721    113.07869]
```

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




```
Type1 variance ratio of three principal components: [9.9796951e-01 1.9084467e-03 8.6843327e-05]
Type1 variance of three principal components: [5.4627750e+04 1.0446627e+02 4.7537079e+00]
Type2 variance ratio of three principal components: [9.9052209e-01 9.0255113e-03 3.4437608e-04]
Type2 variance of three principal components: [2.7538438e+04 2.5092674e+02 9.5743237e+00]
Type3 variance ratio of three principal components: [9.8749131e-01 1.1448372e-02 5.2262913e-04]
Type3 variance of three principal components: [9.7537207e+03 1.1307869e+02 5.1621504e+00]
```

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。