**機器學習理論 HW1**

108064535 陳文遠

|  |
| --- |
| **Step 1. 分割 training data 與 test data 並分別存為 train.csv 和 test.csv** |

* 使用 random.shuffle() 函數將存放 wine 資料的串列打亂，再從每種 type 取出前 18 筆資料來當成 test data (test.csv)
* 其餘沒取到的部分則當成 training data (train.csv)

|  |
| --- |
| **Step 2. 讀取 train.csv 檔並計算其資料的 priori probability, mean, variance** |

* 已知 MAP 的數學式如下 :

其中，

* 上述數學式的 為事前機率 (priori prabability)，意思就是每種 type 紅酒出現的機率 :

第 i 類紅酒

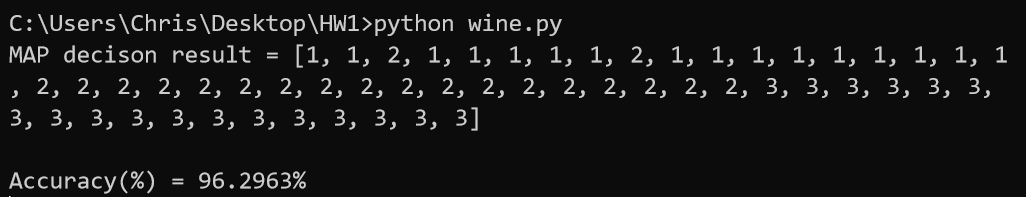
(例如 )

* 上述數學式的 是個 likelihood function，而題目告訴我們 13 種 feature，每個都是一個獨立的 Gaussian distribution，因此要將各個 Gaussian distribution 的平均值及變異數求出來，而我是直接使用 numpy.mean() 以及 numpy.var() 函數來分別求出其值
* 最後的 則是全機率 :

但在此題中我直接將 捨棄不算，因為 3 種 type 的 都一樣，故 MAP 省略為 :

|  |
| --- |
| **Step 3. 讀取 test.csv 檔並將 test data 以及 Step 2. 中求出的 priori probability, mean, variance 代入公式來求出 MAP** |

* 在 Step 2. 中我們已將 省略，並假設 mean 為 、variance 為 、x 為 test data 的資料，則 MAP 公式如下 :
* 特別注意的是，每筆 test data 都會算出 13 筆 posteriori probability，我的作法是將 13 筆乘起來，取最大的那個
* 利用 MAP 偵測出來的 再拿來與 test data 真實的 type 做比對即可得到 accuracy
* 由於 training data 與 test data 是隨機亂分的，因此每次執行根據抓取到的 training data 的好壞，都會得到不同的 accuracy，但基本上 accuracy 都會大於 90%，如下圖

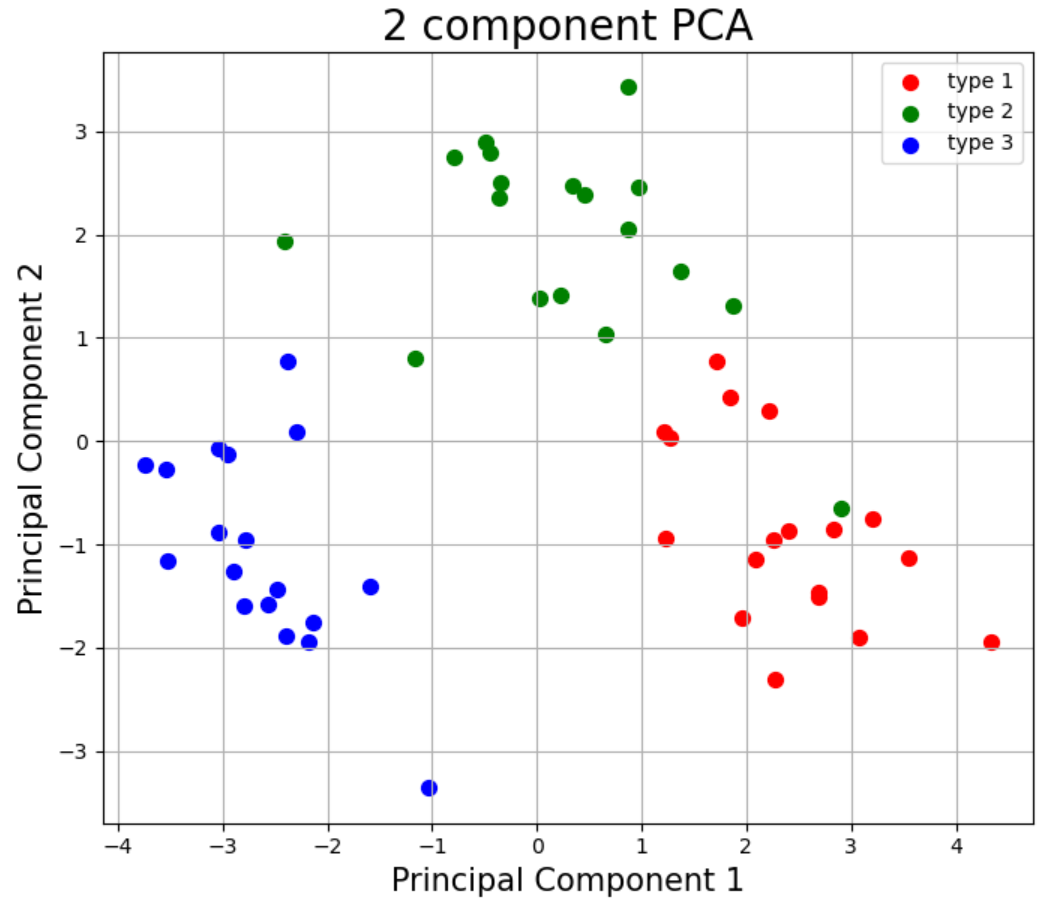


* 下表為測試 10 次的 accuracy

|  |  |
| --- | --- |
| 第 i 次 | 正確率 |
| 1 | 98.1481 % |
| 2 | 96.2963 % |
| 3 | 98.1481 % |
| 4 | 94.4444 % |
| 5 | 98.1481 % |
| 6 | 98.1481 % |
| 7 | 98.1481 % |
| 8 | 94.4444 % |
| 9 | 96.2963 % |
| 10 | 96.2963 % |

|  |
| --- |
| **Step 4. 使用 PCA 將 13 維的 feature 降成 2 維並畫圖** |

* 透過 PCA function 將 13 維的 feature 從中投射出令其特徵差異最大的兩維向量所對應的值，最後再將其畫成散點圖，如下圖



|  |
| --- |
| **Step 5. 討論** |

* 以下圖為例，可以看到使用黃色螢光筆標記的部分，那個 type 2 的紅酒因特徵太過相近於 type 1，因此在此例子中會誤判

