Data Mining project 2

P76074240 資工碩一 蔡文傑

一、介紹

本篇報告在探討機器學習分類問題以及透過分類器訓練出的模型與真實資料之間的差異。

二、資料集產生與視覺化

在本篇報告中，我們設計了一個三個類別的分類問題，每個資料維三個維度共 1000 筆資料。將資料設定為只有三個特徵的原因是為了容易分析與視覺化，每一筆資料剛好可畫在一個三維空間之中以便觀察，也可將資料投影到特定平面來觀察某兩個特徵之間關係。在特徵設計過程中，我們故意將其中一個特徵設計成冗餘特徵(redundant feature)使之不太具有分類訊息，藉此觀察在 Decision Tree、Random Forest、Logistic Regression 與 SVC 使否能捕捉到這個訊息。

資料集是透過 sklearn make_classification 函式依據指定參數產生，產生指令為：

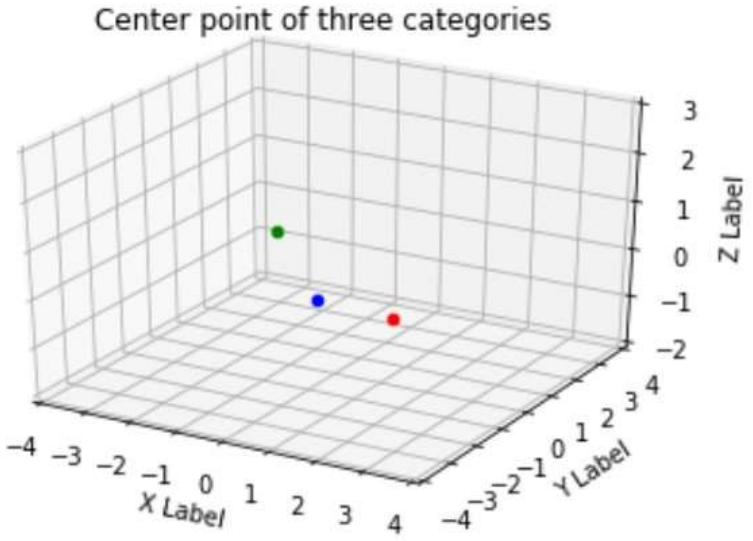
```
X, y = make_classification(n_samples=1000, n_features=3,
n_informative=2, n_redundant=1, n_classes=3, n_clusters_per_class=1,
random_state=2017)
```

參數說明：

1. n_samples=1000: 產生 1000 筆資料。
2. n_features=3: 每一筆資料有三個特徵。
3. n_informative=2: 設定其中兩個特徵為重要。
4. n_redundant=1: 設定其中一個特徵為多餘特徵，此特徵產生方式為透過其中一個重要特徵線性組合產生。
5. n_classes=3, n_clusters_per_class=1: 設定為三個類別。
6. random_state=2017: 亂數種子。

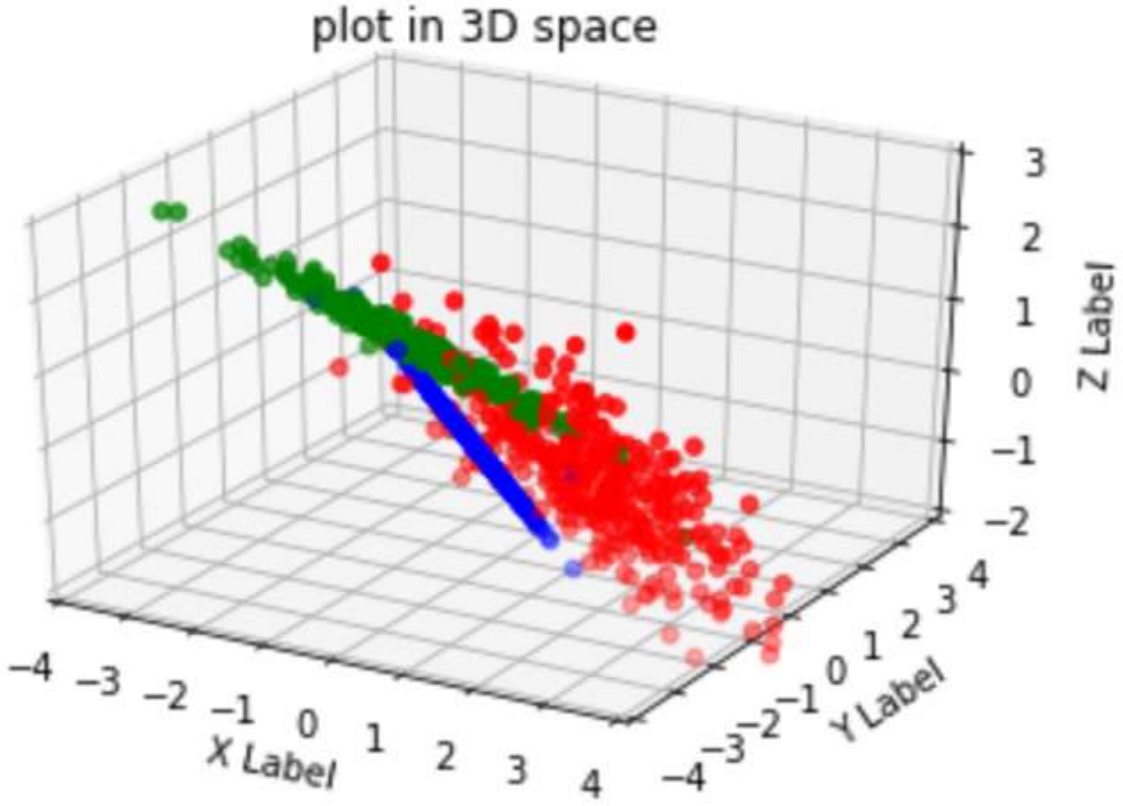
此資料集的‘absolutely right’規則為: 在三度空間中依據某個中心點分布的三個群體，如圖一所述的class0、class1、class2 center point，我們使用不同的顏色(紅、藍、綠)來區分不同類別:

```
class 0: center=[ 0.90779107  0.64984184 -0.96530905], ndata=334
class 1: center=[-0.98778854  1.04855655 -1.04175537], ndata=335
class 2: center=[-0.89324821 -0.62651245  0.93445001], ndata=331
```



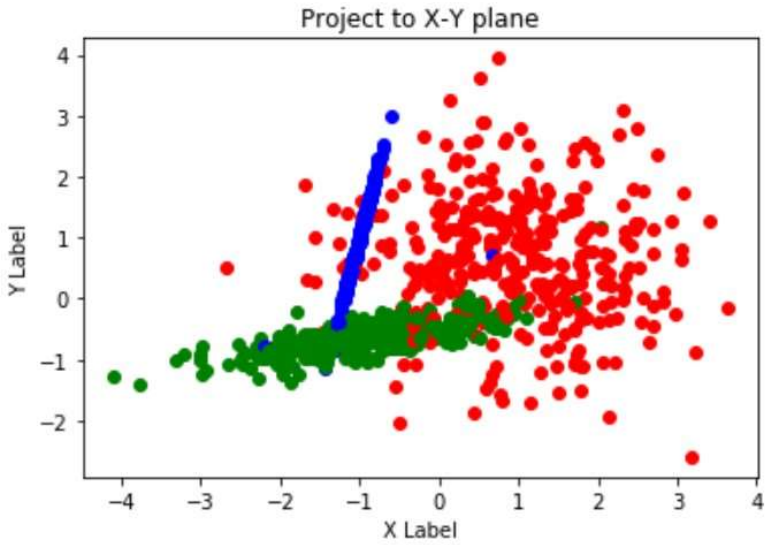
圖一

接下來資料集中每一筆資料都會分成這三個群體分布，如圖二：

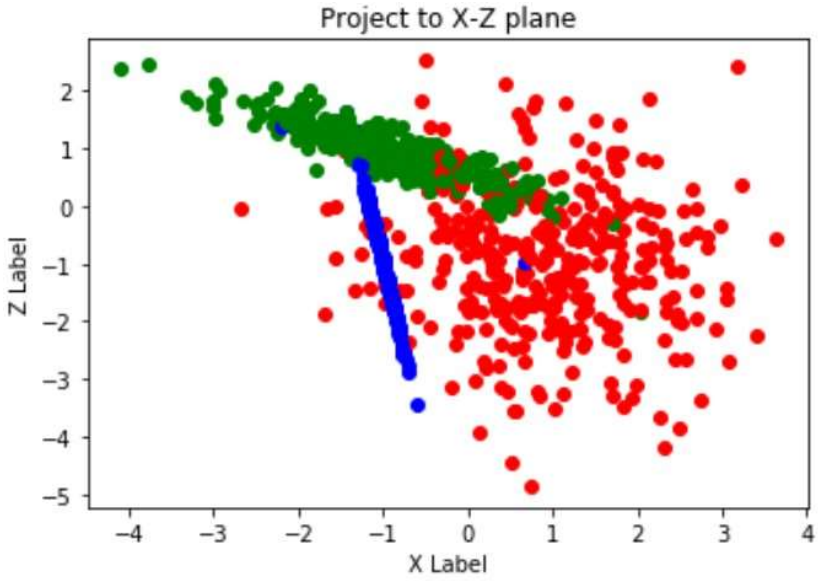


圖二

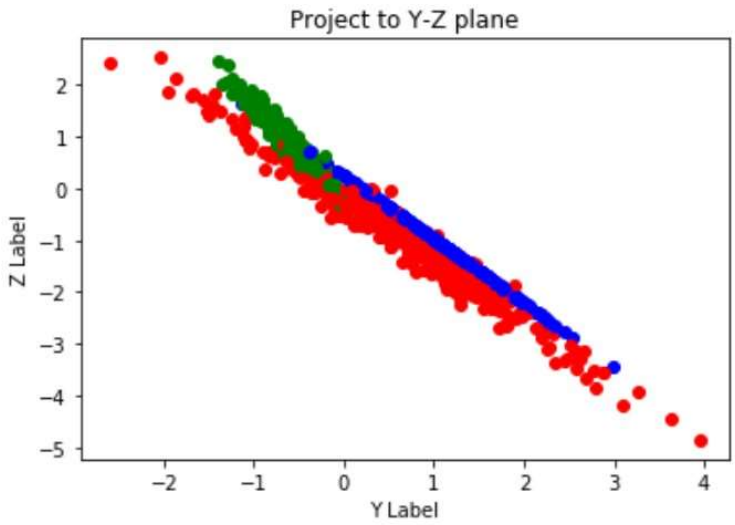
另外一點是，在前面所述的產生規則中我們指定其中一個特徵為冗餘特徵，此特徵會是另外兩個重要特徵的線性組合。因此我們嘗試分別將這些三維空間的點分別投影到不同平面上，觀察是否能找到此特性。結果如以下圖三、圖四與圖五，在圖五 Y-Z 投影可觀察這兩個特徵有線性組合關係。


Project to X-Y plane
Y Label
X Label
4
3
2
1
0
−1
−2
−4
−3
−2
−1
0
1
2
3
4

圖三


Project to X-Z plane
Z Label
X Label
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4

圖四



圖五

資料詳細產生過程請見 generate_data_and_visualize.ipynb。

三、實驗設計

在本篇報告中，我們使用了 Decision Tree、Random Forest、Logistic Regression 與 Support Vector Machine 這四種模型進行分析。分析時會分別使用兩種方式:

1. 將資料集分為 80%作為訓練且 20%作為測試。
2. 使用 10-Fold Cross Validation 進行測試。

我們會依據 accuracy、precision、recall、F1 score 與訓練時間進行分析與比較，我們會將訓練時間取經過自然指數函數處理以便觀察出差異。另外由於某些模型的特殊性，可做更進一步分析，在 Decision Tree 我們會將決策樹實際畫出來；在 Random Forest 可以依據不同的 random tree 找出每一個特徵的重要程度，此特性有助於驗證我們資料集中的冗餘特徵規則。

實驗過程請見 train_and_test.ipynb。

四、實驗結果

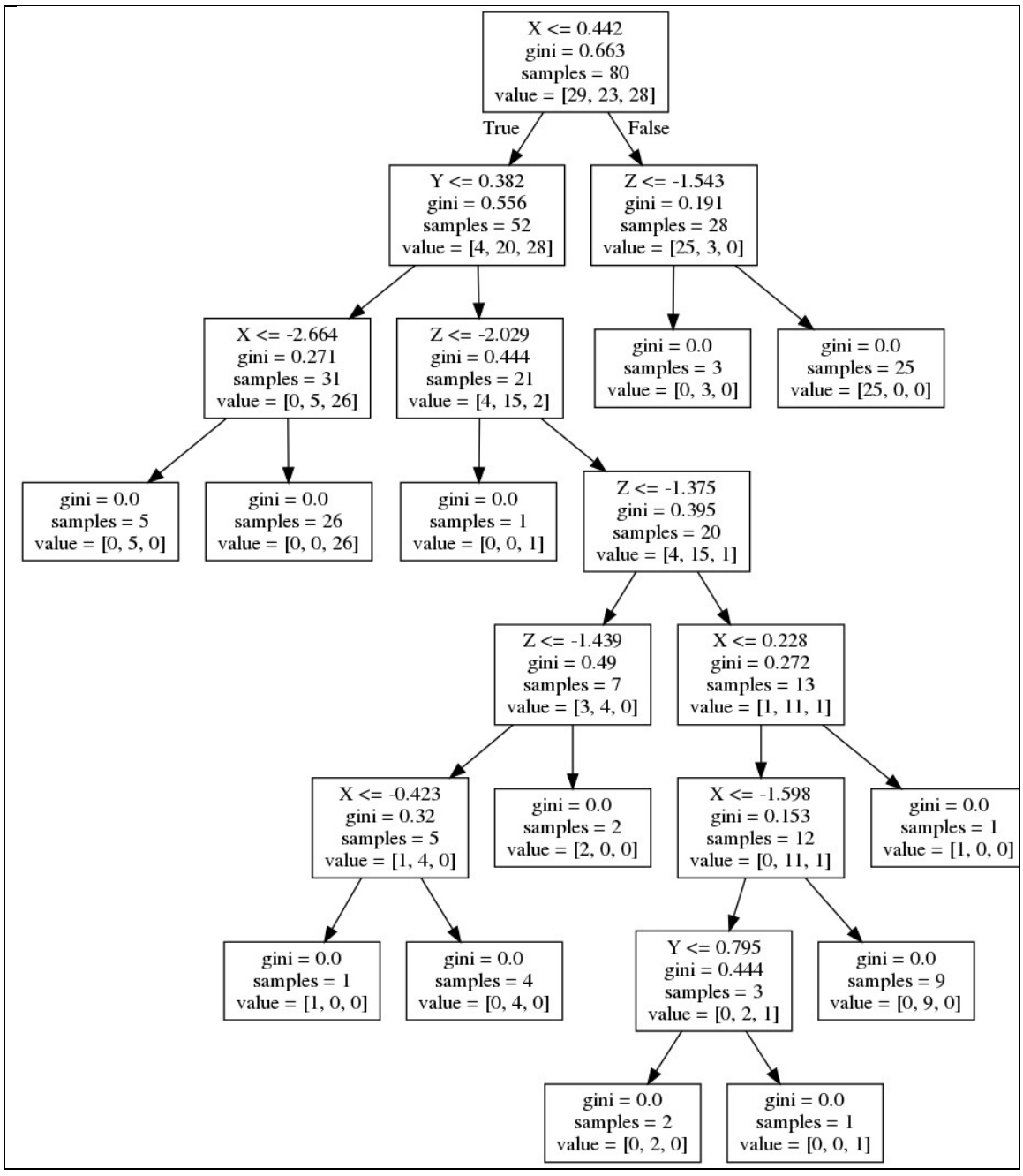
4.1 將資料集分為 80%作為訓練且 20%作為測試。

Model	Accuracy	Precision	Recall	F1-score	Exp of Train time
Decision Tree	0.9	0.9	0.9	0.9	1.0018536830203972
Random Forest	0.93	0.93	0.93	0.93	1.0094740463025924
Logistic Regression	0.875	0.88	0.88	0.88	1.0019784926226591
Support Vector Machine	0.925	0.93	0.93	0.93	1.0047746885802158

4.2 使用 10-Fold Cross Validation 進行測試。

Model	cv scores	mean	std
Decision Tree	[0.93137255 0.87128713 0.88118812 0.93069307 0.89 0.91919192 0.8989899 0.93939394 0.92929293 0.90909091]	0.910	0.022
Random Forest	[0.94117647 0.92079208 0.91089109 0.91089109 0.87 0.90909091 0.92929293 0.92929293 0.94949495 0.94949495]	0.922	0.023
Logistic Regression	[0.87254902 0.85148515 0.8019802 0.87128713 0.82 0.84848485 0.93939394 0.8989899 0.88888889 0.91919192]	0.871	0.040
Support Vector Machine	[0.95098039 0.89108911 0.92079208 0.92079208 0.92 0.88888889 0.95959596 0.92929293 0.93939394 0.93939394]	0.926	0.022

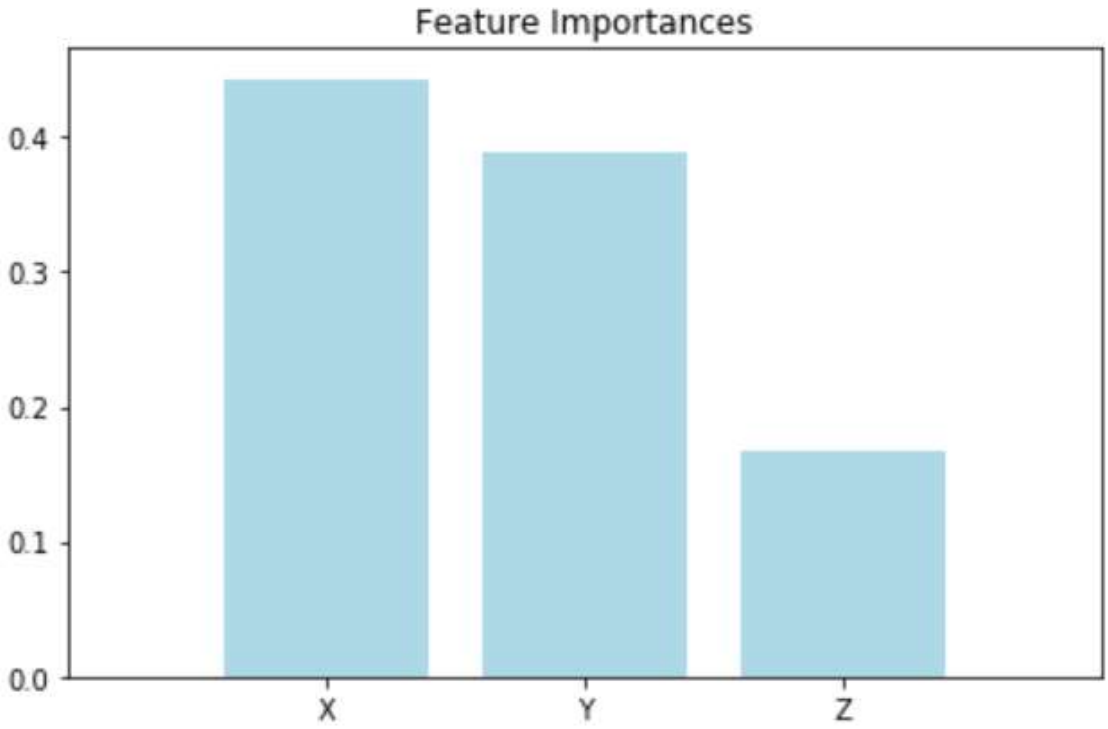
4.3 Decision Tree 決策樹視覺化



4.4 Random Forest 特徵重要性分析

```
X       0.44347027022648283
Y       0.38916443430996134
Z       0.1673652954635558

<function matplotlib.pyplot.show(*args, **kw)>
```



五、分析與結論

1. 在 4.1 將資料集分為 80%作為訓練且 20%作為測試實驗結果中，可以發現 Random Forest 方法不論在 accuracy、precision、recall、F1 score 皆有最好的表現。但是 Decision Tree 有最快的訓練時間。

2. 在 4.2 使用 10-Fold Cross Validation 進行測試實驗中，可以發現 Support Vector Machine 變成是表現最好，但是其與第二名的 Random Forest 差距只有 0.4%些微領先。
3. 在 4.4 Random Forest 特徵重要性分析實驗中，可以發現第三個維度的特徵 Z 在 Random Forest 模型中重要性比較低，這符合我們資料集產生的規則。