

1. 把資料讀取進來(可用pandas套件)

```
data = pd.read_csv('D://character-deaths.csv')
```

```
data
```

```
In [1]: import pandas as pd
...: import numpy as np
...: from sklearn.model_selection import train_test_split
...: from sklearn import tree
...:
...: #讀取檔案
...: data = pd.read_csv('D://character-deaths.csv')

In [2]: data
Out[2]:
```

	Name	Allegiances	Death Year	...	SoS	FfC	DwD
0	Addam Marbrand	Lannister	NaN	...	1	1	0
1	Aegon Frey (Jinglebell)	None	299.0	...	1	0	0
2	Aegon Targaryen	House Targaryen	NaN	...	0	0	1
3	Adrack Humble	House Greyjoy	300.0	...	0	0	1
4	Aemon Costayne	Lannister	NaN	...	1	0	0
...
912	Zollo	None	NaN	...	1	0	0
913	Yurkhaz zo Yunzak	None	300.0	...	0	0	1
914	Yezzan Zo Qaggaz	None	300.0	...	0	0	1
915	Torwynd the Tame	Wildling	300.0	...	1	0	0
916	Talbert Serry	Tyrell	300.0	...	0	1	0

[917 rows x 13 columns]

2. 資料前處理

2.1 把空值以0替代

```
data = data.fillna(0)
```

```
data
```

data - DataFrame

Index	Name	Allegiances	Death Year	ook of Deat	eath Chapt	k Intro Cha	Gender	Nobility	GoT	CoK	So
0	Addam Marbrand	Lannister	0	0	0	56	1	1	1	1	1
1	Aegon Frey (Jinglebell)	None	299	3	51	49	1	1	0	0	1
2	Aegon Targaryen	House Targaryen	0	0	0	5	1	1	0	0	0
3	Adrack Humble	House Greyjoy	300	5	20	20	1	1	0	0	0
4	Aemon Costayne	Lannister	0	0	0	0	1	1	0	0	1
5	Aemon Estermont	Baratheon	0	0	0	0	1	1	0	1	1
6	Aemon Targaryen (son of Maekar I)	Night's Watch	300	4	35	21	1	1	1	0	1
7	Aenys Frey	None	300	5	0	59	0	1	1	1	1
8	Aeron Greyjoy	House Greyjoy	0	0	0	11	1	1	0	1	0
9	Aethan	Night's Watch	0	0	0	0	1	0	0	0	1
10	Aggar	House Greyjoy	299	2	56	50	1	0	0	1	0
11	Aggo	House Targaryen	0	0	0	54	1	0	1	1	1
12	Alan of Rosby	Night's Watch	300	5	4	18	1	1	0	1	1
13	Alayaya	None	0	0	0	15	0	0	0	1	0

2.2 Death Year , Book of Death , Death Chapter三者取一個，將有數值的轉成1

選擇 Death Chapter 欄位作為 Label ， 找到 Actual Data ， 並命名為Y

```
Y = data['Death Chapter']
Y = pd.DataFrame(np.where(Y>0,1,0))
Y.columns=['Actual Data']
Y
```

```
...: #將 Death Chapter 視為 Label
...: Y = data['Death Chapter']
...: Y = pd.DataFrame(np.where(Y>0,1,0))
...: Y.columns=['Actual Data']

In [6]: Y
Out[6]:
   Actual Data
0            0
1            1
2            0
3            1
4            0
..          ...
912           0
913           1
914           1
915           1
916           1

[917 rows x 1 columns]
```

2.3 將 Allegiances 轉成 dummy 特徵

將變數項移除，並將 Allegiances 轉換成 dummy 特徵

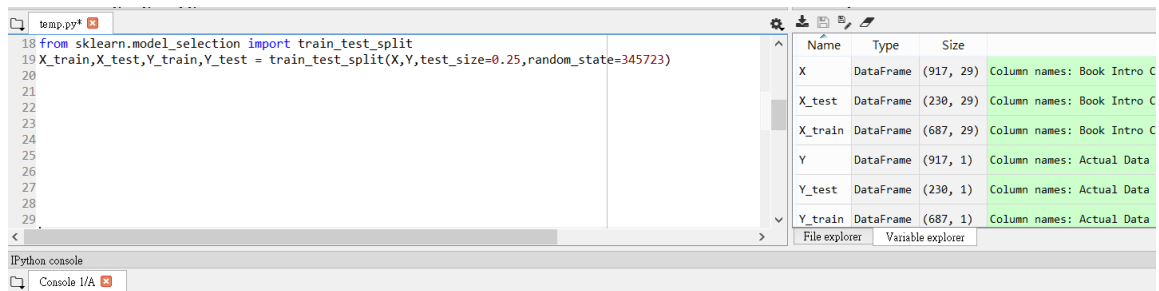
```
X = data.drop(['Name','Death Year','Book of Death','Death Chapter'],axis=1)
X = pd.get_dummies(X, columns=['Allegiances'])
print(chd['Allegiances'].value_counts())
```

```
...: X = pd.get_dummies(X, columns=['Allegiances'])
...: print(data['Allegiances'].value_counts())

None                253
Night's Watch       116
Lannister            81
Stark                73
Baratheon            56
Greyjoy              51
Wildling             40
House Stark          35
Martell              25
House Greyjoy        24
Arryn                23
Tully                22
House Lannister       21
House Targaryen       19
Targaryen            17
Tyrell               15
House Martell         12
House Tyrell          11
House Tully           8
House Baratheon       8
House Arryn           7
Name: Allegiances, dtype: int64
```

2.4 亂數拆成訓練集(75%)與測試集(25%)

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test =
train_test_split( X,Y,test_size=0.25,random_state=345723 )
```



```
In [27]: from sklearn.model_selection import train_test_split
...: X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25,random_state=345723)
```

3. 使用 scikit-learn 的 DecisionTreeClassifier 進行預測

決策樹預測準確度為0.76419

```
from sklearn import tree
clf = tree.DecisionTreeClassifier(max_depth=5)
clf = clf.fit(X_train,Y_train)
pred = clf.predict(X_test)
print('Accuracy of training set',clf.score(X_train,Y_train))
```

```
...: #運用 scikit-Learn 套件切分訓練集與測試集
...: X_train,X_test,Y_train,Y_test =
train_test_split(X,Y,test_size=0.25,random_state=345723)
...:
...: #呼叫 scikit-Learn 決策數分類器執行預測
...: clf = tree.DecisionTreeClassifier(max_depth=5)
...: clf = clf.fit(X_train,Y_train)
...: pred = clf.predict(X_test)
...: print('Accuracy of training set',clf.score(X_train,Y_train))
Accuracy of training set 0.7641921397379913
```

4. 做出Confusion Matrix，並計算Precision, Recall, Accuracy

Confusion Matrix 成功計算出

Accuracy : 0.70434 / Precision : 0.55421 / Recall : 0.59740

```
CF_M = pd.DataFrame( confusion_matrix(Y_test, pred),
                      index=['Actual Alive','Actual Death'],
                      columns=['Predicted Alive','Predicted Death'])

print('Testing Set')
print('Accuracy:',accuracy_score(Y_test, pred))
print('Precision:',precision_score(Y_test, pred))
print('Recall:',recall_score(Y_test, pred))
```

