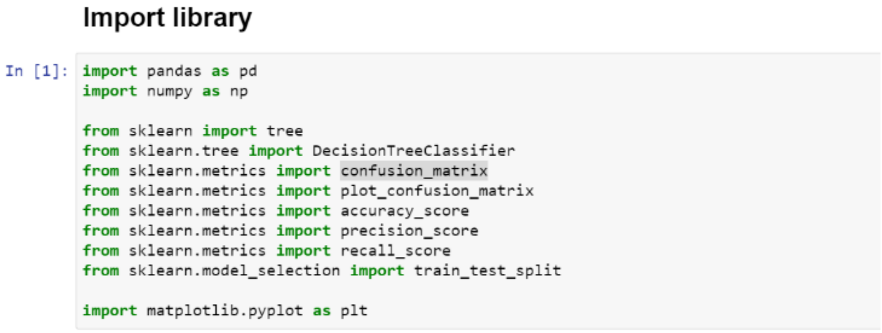
HW1_Decision Tree

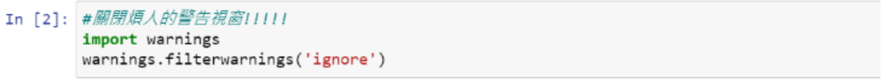
0813458 資財 簡辰穎

一. Import library



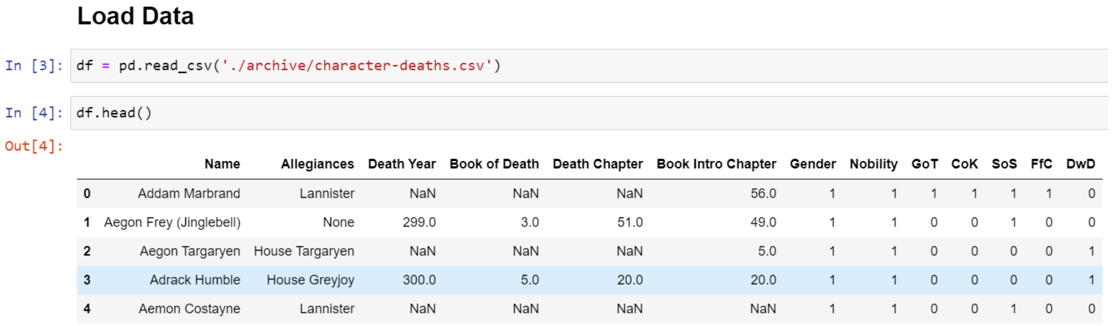
Import library

In [1]:
```
import pandas as pd
import numpy as np

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt
```



In [2]:
```
#關閉煩人的警告視窗!!!!!
import warnings
warnings.filterwarnings('ignore')
```

二. 匯入資料集



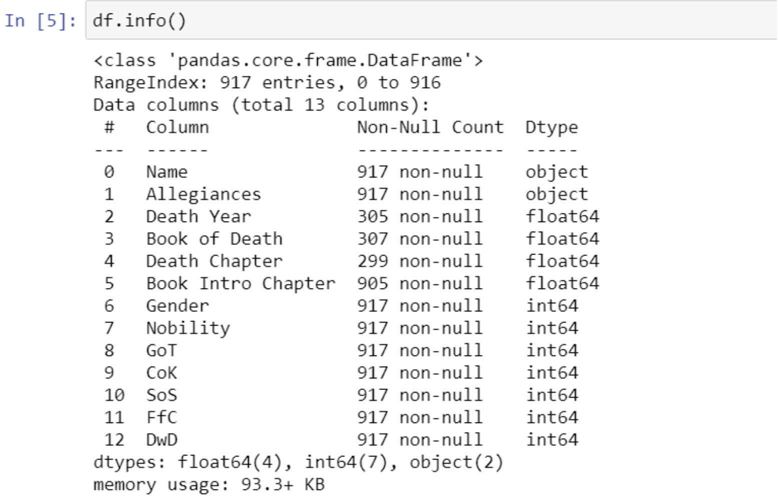
Load Data

In [3]:
```
df = pd.read_csv('./archive/character-deaths.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	Name	Allegiances	Death Year	Book of Death	Death Chapter	Book Intro Chapter	Gender	Nobility	GoT	CoK	SoS	FfC	DwD
0	Addam Marbrand	Lannister	NaN	NaN	NaN	56.0	1	1	1	1	1	1	0
1	Aegon Frey (Jinglebell)	None	299.0	3.0	51.0	49.0	1	1	0	0	1	0	0
2	Aegon Targaryen	House Targaryen	NaN	NaN	NaN	5.0	1	1	0	0	0	0	1
3	Adrack Humble	House Greyjoy	300.0	5.0	20.0	20.0	1	1	0	0	0	0	1
4	Aemon Costayne	Lannister	NaN	NaN	NaN	NaN	1	1	0	0	1	0	0



In [5]:
```
df.info()
```

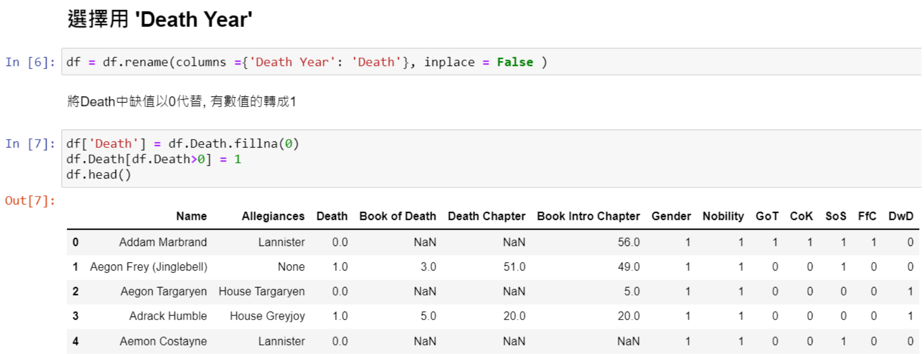
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 917 entries, 0 to 916
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                917 non-null    object 
 1   Allegiances         917 non-null    object 
 2   Death Year          305 non-null    float64
 3   Book of Death       307 non-null    float64
 4   Death Chapter       299 non-null    float64
 5   Book Intro Chapter  905 non-null    float64
 6   Gender              917 non-null    int64  
 7   Nobility            917 non-null    int64  
 8   GoT                 917 non-null    int64  
 9   CoK                 917 non-null    int64  
 10  SoS                 917 non-null    int64  
 11  FfC                 917 non-null    int64  
 12  DwD                 917 non-null    int64  
dtypes: float64(4), int64(7), object(2)
memory usage: 93.3+ KB
```

三. 資料前處理

- 選擇用 Death Year 當作預測目標
- Allegiances 轉成 dummy 特徵
- 刪除不會再用到多餘的欄位

一　處理缺失值

a.　用 Death Year　當作預測目標



選擇用 'Death Year'

```
In [6]: df = df.rename(columns ={'Death Year': 'Death'}, inplace = False )
```

將Death中缺值以0代替, 有數值的轉成1

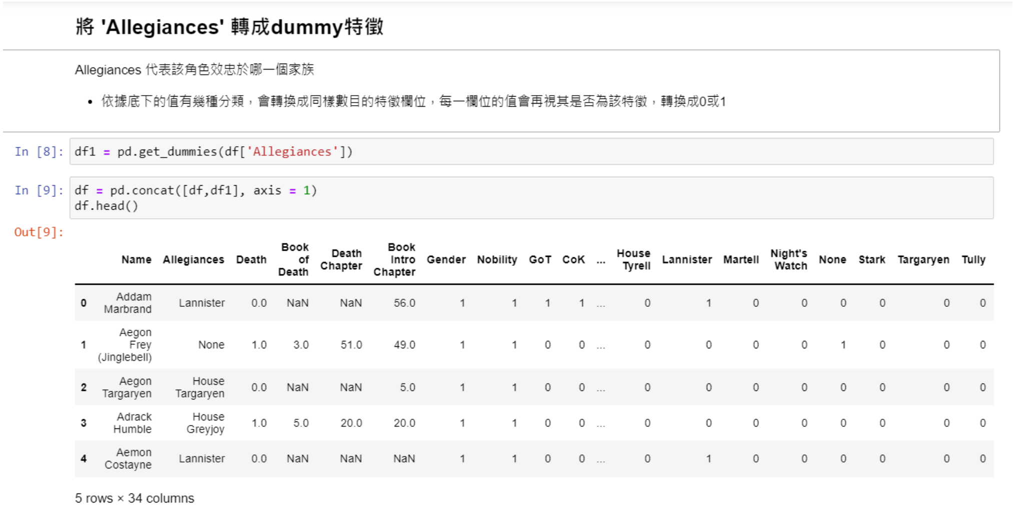
```
In [7]: df['Death'] = df.Death.fillna(0)
        df.Death[df.Death>0] = 1
        df.head()
```

Out[7]:

	Name	Allegiances	Death	Book of Death	Death Chapter	Book Intro Chapter	Gender	Nobility	GoT	CoK	SoS	FfC	DwD
0	Addam Marbrand	Lannister	0.0	NaN	NaN	56.0	1	1	1	1	1	1	0
1	Aegon Frey (Jinglebell)	None	1.0	3.0	51.0	49.0	1	1	0	0	1	0	0
2	Aegon Targaryen	House Targaryen	0.0	NaN	NaN	5.0	1	1	0	0	0	0	1
3	Adrack Humble	House Greyjoy	1.0	5.0	20.0	20.0	1	1	0	0	0	0	1
4	Aemon Costayne	Lannister	0.0	NaN	NaN	NaN	1	1	0	0	1	0	0

Death Year , Book of Death , Death Chapter 三者取一個，選擇用 Death Year 當預測目標，把空值補 0，有數值的轉成 1

b.　將 Allegiances 轉成 dummy 特徵



將 'Allegiances' 轉成dummy特徵

Allegiances 代表該角色效忠於哪一個家族

- 依據底下的值有幾種分類，會轉換成同樣數目的特徵欄位，每一欄位的值會再視其是否為該特徵，轉換成0或1

```
In [8]: df1 = pd.get_dummies(df['Allegiances'])
```

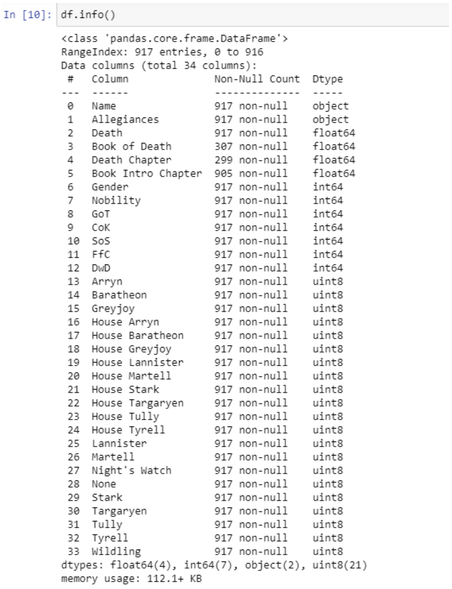
```
In [9]: df = pd.concat([df,df1], axis = 1)
        df.head()
```

Out[9]:

	Name	Allegiances	Death	Book of Death	Death Chapter	Book Intro Chapter	Gender	Nobility	GoT	CoK	...	House Tyrell	Lannister	Martell	Night's Watch	None	Stark	Targaryen	Tully
0	Addam Marbrand	Lannister	0.0	NaN	NaN	56.0	1	1	1	1	...	0	1	0	0	0	0	0	0
1	Aegon Frey (Jinglebell)	None	1.0	3.0	51.0	49.0	1	1	0	0	...	0	0	0	0	1	0	0	0
2	Aegon Targaryen	House Targaryen	0.0	NaN	NaN	5.0	1	1	0	0	...	0	0	0	0	0	0	0	0
3	Adrack Humble	House Greyjoy	1.0	5.0	20.0	20.0	1	1	0	0	...	0	0	0	0	0	0	0	0
4	Aemon Costayne	Lannister	0.0	NaN	NaN	NaN	1	1	0	0	...	0	1	0	0	0	0	0	0

5 rows × 34 columns

將 Allegiances 轉成 dummy 特徵(底下有幾種分類就會變成幾種特徵，值是 0 或 1，本來的資料集就會再增加約 20 種特徵)

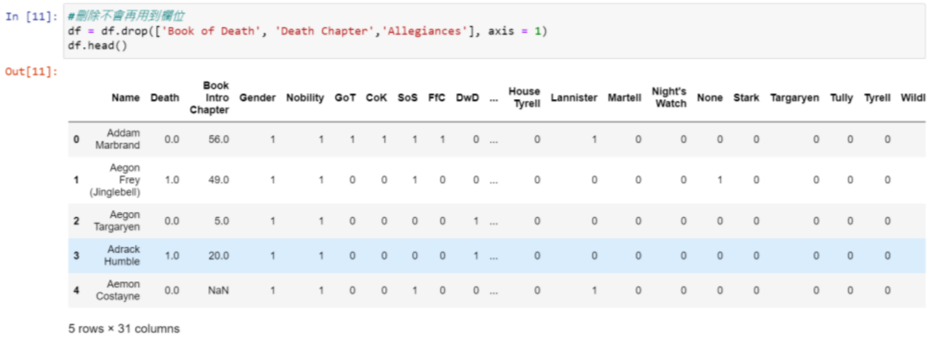


```
In [10]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 917 entries, 0 to 916
Data columns (total 34 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                917 non-null    object 
 1   Allegiances         917 non-null    object 
 2   Death               917 non-null    float64
 3   Book of Death       307 non-null    float64
 4   Death Chapter       299 non-null    float64
 5   Book Intro Chapter  905 non-null    float64
 6   Gender              917 non-null    int64  
 7   Nobility            917 non-null    int64  
 8   GoT                 917 non-null    int64  
 9   CoK                 917 non-null    int64  
 10  SoS                 917 non-null    int64  
 11  FfC                 917 non-null    int64  
 12  DwD                 917 non-null    int64  
 13  Arryn               917 non-null    uint8  
 14  Baratheon           917 non-null    uint8  
 15  Greyjoy             917 non-null    uint8  
 16  House Arryn         917 non-null    uint8  
 17  House Baratheon     917 non-null    uint8  
 18  House Greyjoy       917 non-null    uint8  
 19  House Lannister     917 non-null    uint8  
 20  House Martell       917 non-null    uint8  
 21  House Stark         917 non-null    uint8  
 22  House Targaryen     917 non-null    uint8  
 23  House Tully         917 non-null    uint8  
 24  House Tyrell        917 non-null    uint8  
 25  Lannister           917 non-null    uint8  
 26  Martell             917 non-null    uint8  
 27  Night's Watch       917 non-null    uint8  
 28  None                917 non-null    uint8  
 29  Stark               917 non-null    uint8  
 30  Targaryen           917 non-null    uint8  
 31  Tully               917 non-null    uint8  
 32  Tyrell              917 non-null    uint8  
 33  Wildling            917 non-null    uint8  
dtypes: float64(4), int64(7), object(2), uint8(21)
memory usage: 112.1+ KB
```

Allegiances 轉成 dummy 特徵後所有欄位

(Book Intro Chapter 還有空值沒處理)

c. 刪除多餘欄位



```
In [11]: #刪除不會再用到欄位
         df = df.drop(['Book of Death', 'Death Chapter','Allegiances'], axis = 1)
         df.head()
```

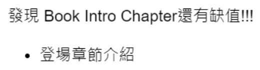
Out[11]:

	Name	Death	Book Intro Chapter	Gender	Nobility	GoT	CoK	SoS	FfC	DwD	...	House Tyrell	Lannister	Martell	Night's Watch	None	Stark	Targaryen	Tully	Tyrell	Wildl
0	Addam Marbrand	0.0	56.0	1	1	1	1	1	1	0	...	0	1	0	0	0	0	0	0	0	
1	Aegon Frey (Jinglebell)	1.0	49.0	1	1	0	0	1	0	0	...	0	0	0	0	1	0	0	0	0	
2	Aegon Targaryen	0.0	5.0	1	1	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	
3	Adrack Humble	1.0	20.0	1	1	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	
4	Aemon Costayne	0.0	NaN	1	1	0	0	1	0	0	...	0	1	0	0	0	0	0	0	0	

5 rows × 31 columns

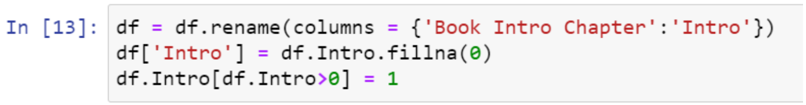
刪除已經處理過後，不會再用到的欄位（Book of Death, Death Chapter, Allegiances）

d. 處理缺失值



發現 Book Intro Chapter還有缺值!!!

- 登場章節介紹

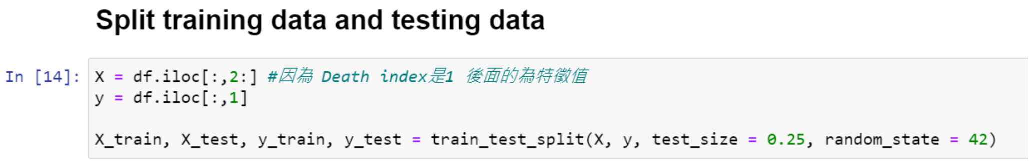


```
In [13]: df = df.rename(columns = {'Book Intro Chapter':'Intro'})
         df['Intro'] = df.Intro.fillna(0)
         df.Intro[df.Intro>0] = 1
```

資料集中 Book Intro Chapter 中還有許多空值，將空值補 0，有數值的轉成 1

四. 亂數拆成訓練集(75%)與測試集(25%)

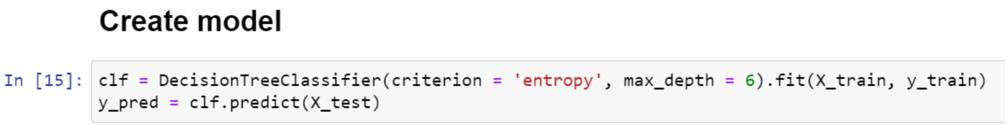
- random state 設 42



Split training data and testing data

```
In [14]: X = df.iloc[:,2:] #因為 Death index是1 後面的為特徵值
         y = df.iloc[:,1]

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 42)
```

五. 使用 Scikit-learn 的 DecisionTreeClassifier 進行預測



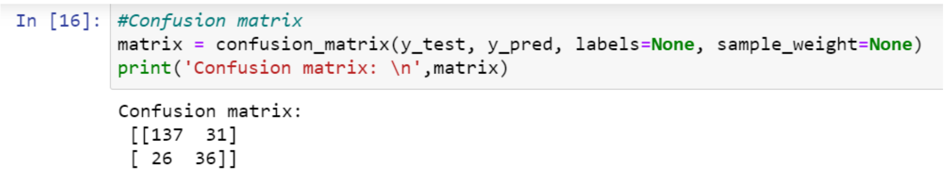
Create model

```
In [15]: clf = DecisionTreeClassifier(criterion = 'entropy', max_depth = 6).fit(X_train, y_train)
         y_pred = clf.predict(X_test)
```

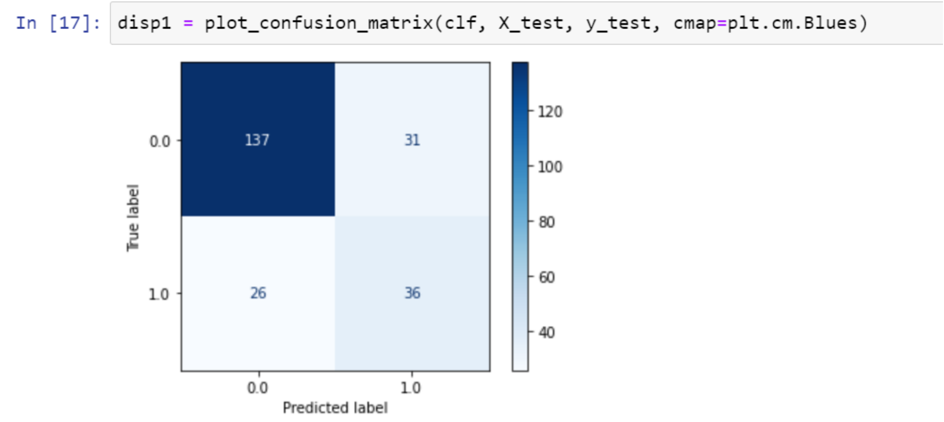
- 篩選條件使用 entropy(也可以用 gini，但跑出來的結果稍微比較差一點)
- 限制模型深度為 6

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

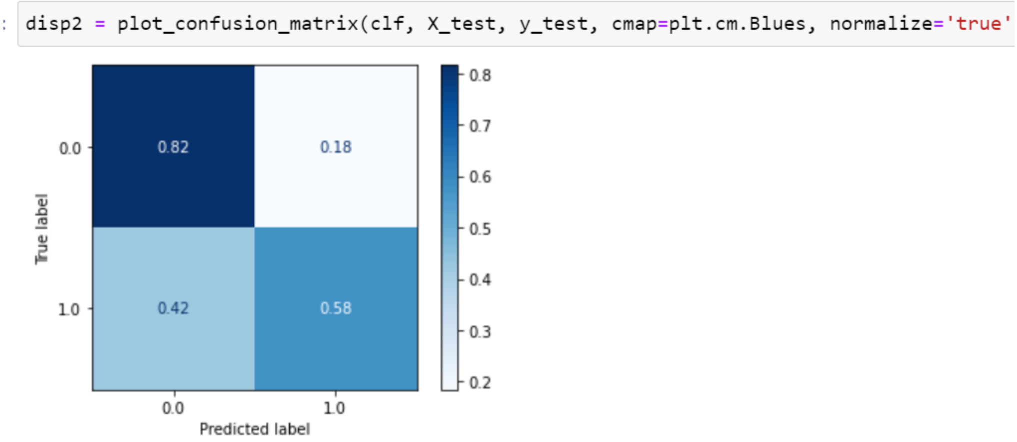
a. Confusion matrix



```
In [16]: #Confusion matrix
         matrix = confusion_matrix(y_test, y_pred, labels=None, sample_weight=None)
         print('Confusion matrix: \n',matrix)

         Confusion matrix:
          [[137  31]
          [ 26  36]]
```

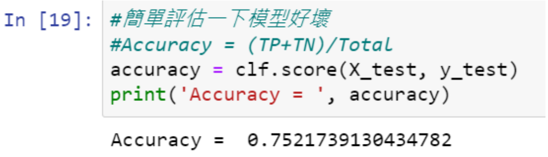


```
In [17]: disp1 = plot_confusion_matrix(clf, X_test, y_test, cmap=plt.cm.Blues)
```



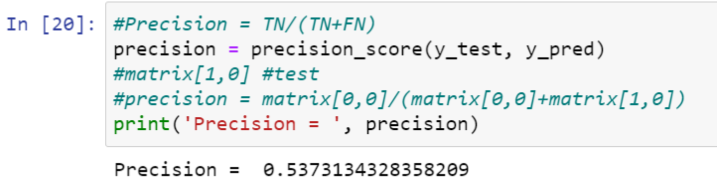
```
: disp2 = plot_confusion_matrix(clf, X_test, y_test, cmap=plt.cm.Blues, normalize='true'
```

b. Accuracy



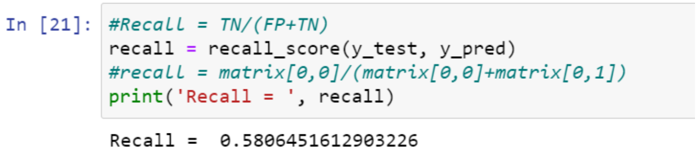
```
In [19]: #簡單評估一下模型好壞
         #Accuracy = (TP+TN)/Total
         accuracy = clf.score(X_test, y_test)
         print('Accuracy = ', accuracy)

         Accuracy =  0.7521739130434782
```

c. Precision



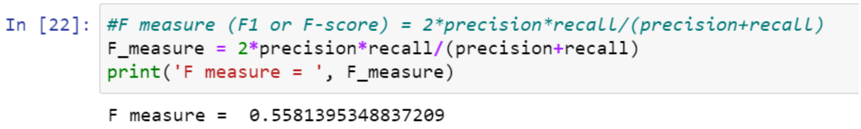
```
In [20]: #Precision = TN/(TN+FN)
         precision = precision_score(y_test, y_pred)
         #matrix[1,0] #test
         #precision = matrix[0,0]/(matrix[0,0]+matrix[1,0])
         print('Precision = ', precision)

         Precision =  0.5373134328358209
```

d. Recall



```
In [21]: #Recall = TN/(FP+TN)
         recall = recall_score(y_test, y_pred)
         #recall = matrix[0,0]/(matrix[0,0]+matrix[0,1])
         print('Recall = ', recall)

         Recall =  0.5806451612903226
```

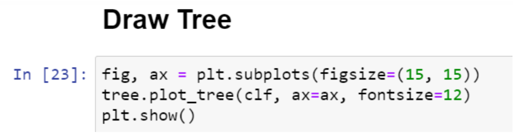
e. F-measure



```
In [22]: #F measure (F1 or F-score) = 2*precision*recall/(precision+recall)
         F_measure = 2*precision*recall/(precision+recall)
         print('F measure = ', F_measure)

         F measure =  0.5581395348837209
```

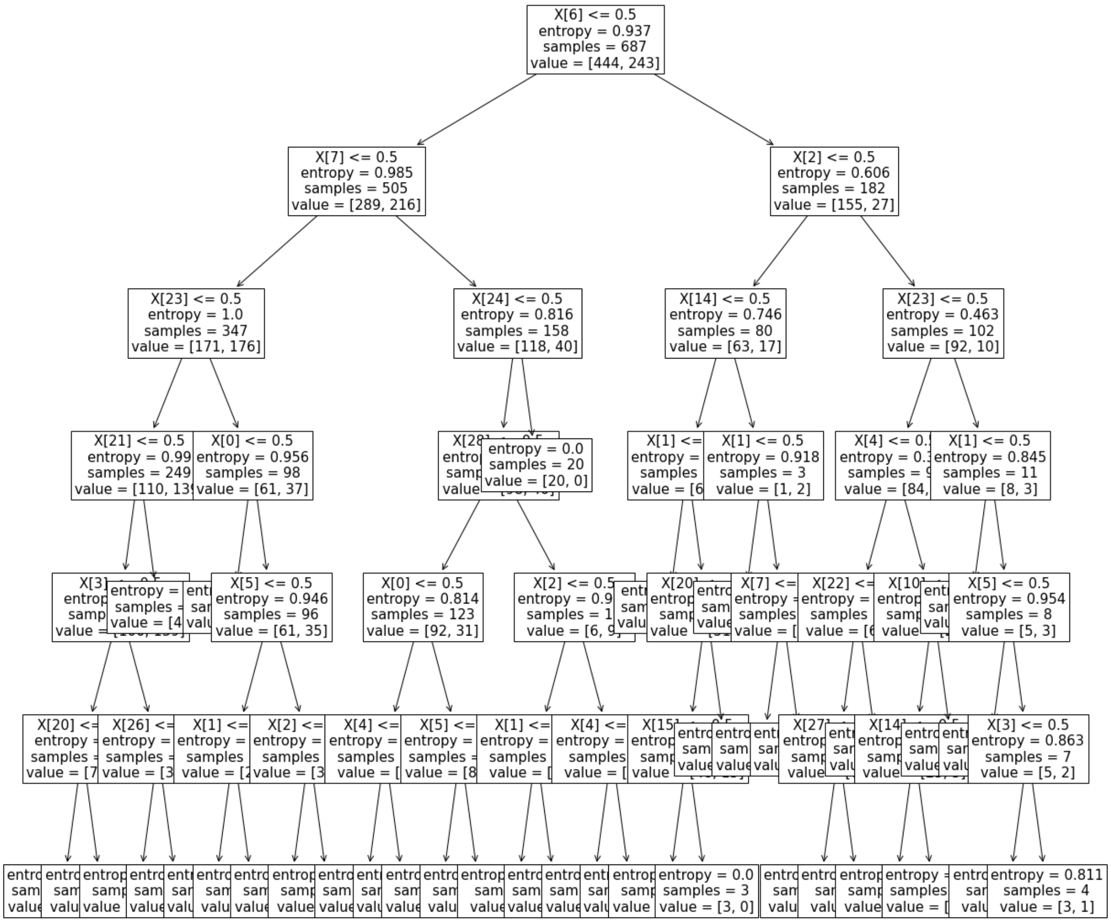
七、 產出決策樹的圖

- 用 matplotlib 裡面的 tree.plot_tree 函數畫出決策數



Draw Tree

```
In [23]: fig, ax = plt.subplots(figsize=(15, 15))
         tree.plot_tree(clf, ax=ax, fontsize=12)
         plt.show()
```



X[6] <= 0.5 entropy = 0.937 samples = 687 value = [444, 243]
X[7] <= 0.5 entropy = 0.985 samples = 505 value = [289, 216]
X[2] <= 0.5 entropy = 0.606 samples = 182 value = [155, 27]
X[23] <= 0.5 entropy = 1.0 samples = 347 value = [171, 176]
X[24] <= 0.5 entropy = 0.816 samples = 158 value = [118, 40]
X[14] <= 0.5 entropy = 0.746 samples = 80 value = [63, 17]
X[23] <= 0.5 entropy = 0.463 samples = 102 value = [92, 10]
X[0] <= 0.5 entropy = 0.956 samples = 98 value = [61, 37]
entropy = 0.0 samples = 20 value = [20, 0]
X[1] <= 0.5 entropy = 0.918 samples = 3 value = [1, 2]
X[1] <= 0.5 entropy = 0.845 samples = 11 value = [8, 3]
X[5] <= 0.5 entropy = 0.946 samples = 96 value = [61, 35]
X[0] <= 0.5 entropy = 0.814 samples = 123 value = [92, 31]
X[5] <= 0.5 entropy = 0.954 samples = 8 value = [5, 3]
X[3] <= 0.5 entropy = 0.863 samples = 7 value = [5, 2]
entropy = 0.0 samples = 3 value = [3, 0]
entropy = 0.811 samples = 4 value = [3, 1]