Data Mining

Project 2

Q36071059 蔡榮漾

本次作業採用 Kaggle 上的鐵達尼 dataset(https://www.kaggle.com/c/titanic/data)中的 train.csv 與 test.csv，在 Kaggle 的 Jupyter & VS Code 的 Python3 環境上執行。

選擇使用此資料庫的原因為：

1. 乘客可被簡單分類為生還 or 罹難兩種狀態
2. 此資料庫有多種屬性可討論以及作為分類依據
3. 資料缺失數量在可接受範圍

一、檢視資料：

首先我先查看前 10 筆資料以了解本資料的格式以及屬性，並思考有那些屬性適合被拿來作為分類的依據。

除了生還與否的屬性外，其餘 11 個屬性代表意義如下：

PassengerId：乘客 ID

Pclass：艙等

Name：乘客姓名

Sex：性別

Age：年齡

SibSp：堂兄弟/妹數

Parch：父母&小孩數

Ticket：船票資訊

Fare ：票價

Cabin：客艙編號

Embarked： 登船港口

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
0	22.0	NaN	S	7.2500	Braund, Mr. Owen Harris	0	1	3	male	1	0.0	A/5 21171
1	38.0	C85	C	71.2833	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	2	1	female	1	1.0	PC 17599
2	26.0	NaN	S	7.9250	Heikkinen, Miss. Laina	0	3	3	female	0	1.0	STON/O2. 3101282
3	35.0	C123	S	53.1000	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	4	1	female	1	1.0	113803
4	35.0	NaN	S	8.0500	Allen, Mr. William Henry	0	5	3	male	0	0.0	373450
5	NaN	NaN	Q	8.4583	Moran, Mr. James	0	6	3	male	0	0.0	330877
6	54.0	E46	S	51.8625	McCarthy, Mr. Timothy J	0	7	1	male	0	0.0	17463
7	2.0	NaN	S	21.0750	Palsson, Master. Gosta Leonard	1	8	3	male	3	0.0	349909
8	27.0	NaN	S	11.1333	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	2	9	3	female	0	1.0	347742
9	14.0	NaN	C	30.0708	Nasser, Mrs. Nicholas (Adele Achem)	0	10	2	female	1	1.0	237736
10	4.0	G6	S	16.7000	Sandstrom, Miss. Marguerite Rut	1	11	3	female	1	1.0	PP 9549

接著我利用以下指令以查看資料的狀況。

```
print(df_train.head())
print(df_train.info())
```

得到結果：

```
   PassengerId  Survived  Pclass    ...        Fare Cabin  Embarked
0            1         0       3    ...      7.2500   NaN         S
1            2         1       1    ...     71.2833   C85         C
2            3         1       3    ...      7.9250   NaN         S
3            4         1       1    ...     53.1000  C123         S
4            5         0       3    ...      8.0500   NaN         S

[5 rows x 12 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
None
```

至此我們得知，Age、Cabin 與 Embarked 有資料缺失，其中又以 Cabin 最為嚴重；但常理思考下，客艙編號與艙等/票價兩種屬性應會有因果上關聯，故我傾向可以直接捨棄該屬性。

而 Age 部分，一開始我在思考是否以平均數或中位數填滿所有缺失值，但又怕會影響分析結果，故暫時擱置。

查看各資料之分布，或許會得到有用資訊。

```
df_train.describe()
```

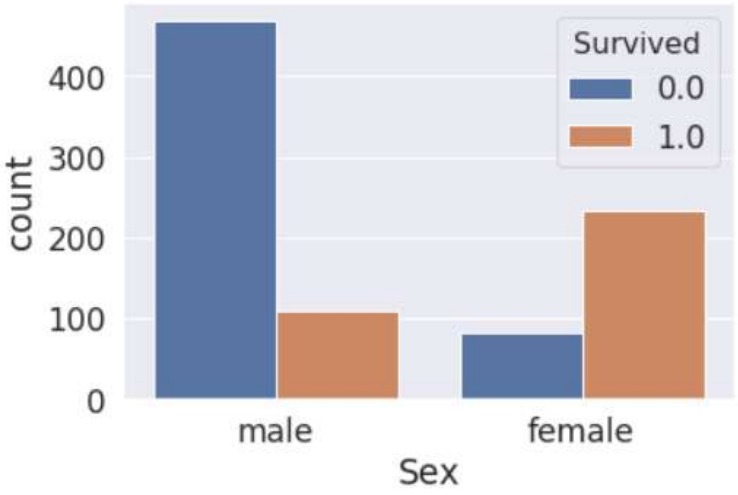
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

後來發現能獲得額外資訊的屬性僅有 Pclass、Age、Fare 三者。

考量各屬性間的特性，與衡量我個人的能力，我能夠使用的屬性應有下列四者：Pclass 艙等、Sex 性別、Age 年齡、Fare 票價 ，而艙等與票價有所掛勾，故取艙等捨棄票價。

二、**屬性分析**：

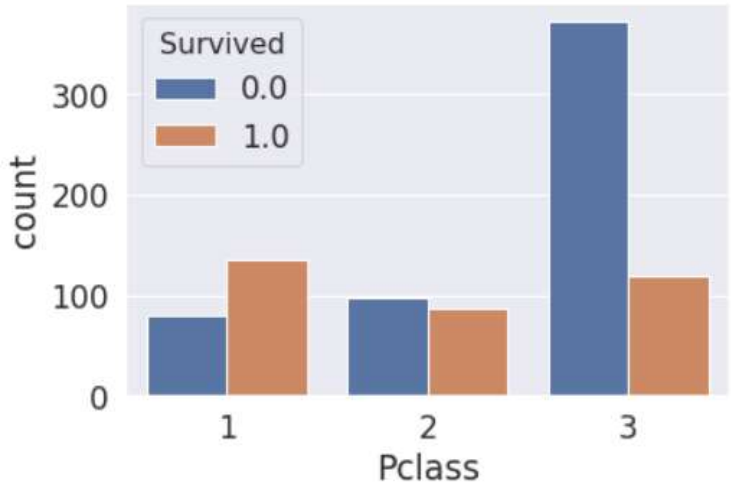
首先我先查看 Sex 屬性與生還與否間的關係，得到：



	Sex	Survived
0	female	0.742
1	male	0.189

顯然女性的生還率顯著的高於男性，推測就像電影所演，船難當下傾向讓老弱婦孺先行搭上救生艇，故資料顯示男性生還率較低也就能理解。

接著查看艙等：



	Pclass	Survived
0	1	0.630
1	2	0.473
2	3	0.242

得到生還率與艙等之間的關係：艙等越高生還率越高，我一開始是推測救生艇等求生工具在較高級的艙等會較為齊全。
但依據此圖得到的資訊更像是，因為低艙等搭乘人數較多，僧多粥少的狀況下死亡人數拉高，才會造成死亡率攀升，從圖中看來三種艙等的生還人數其實是差不多的。

三、**資料處理**：

首先由於性別屬性使用的表示方式是 male 與 female(字串)，無法直接丟進分類樹分析，先將其轉為 1/0 表示。

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
le.fit(df_train['Sex'])
df_train['Sex']=le.transform(df_train['Sex'])
print(df_train.Sex)
```

```
0      1
1      0
2      0
3      0
4      1
5      1
6      1
7      1
8      0
```

接著便是 Age 資料缺失部分的處理，這邊先嘗試用平均值去填滿空缺的部分，進行嘗試。

```
df_train.Age.fillna(df_train.Age.mean(),inplace=True)
```

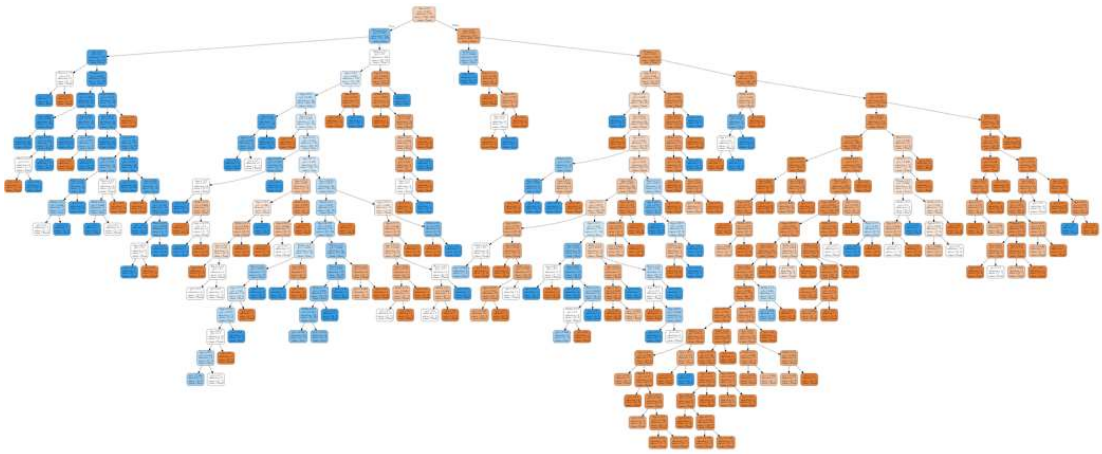
四、**決策樹分類**：

用選定的三個屬性進行決策樹的訓練，並用交叉驗證法計算此模型的準確率，得到約 80.14%的準確率，並將決策樹畫出來：。

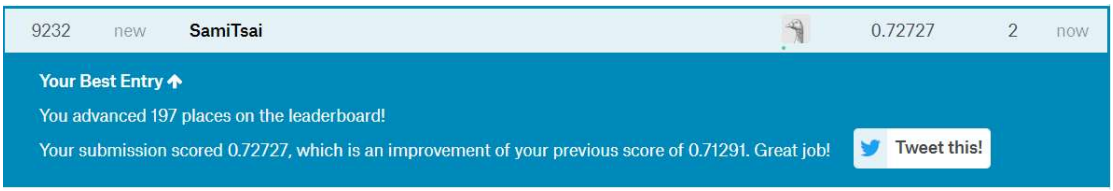
```
features=['Pclass','Sex','Age']
X=df_train[features]
y=df_train['Survived']
dt=tree.DecisionTreeClassifier()
score=cross_val_score(dt,X,y,cv=5,scoring='accuracy')
import numpy as np
print(np.mean(score))
```

```
0.801416348114237
```

```
import graphviz
dt.fit(X,y)
from sklearn.tree import export_graphviz
f=export_graphviz(dt,feature_names=['Pclass','Sex','Age'],
    class_names = ['Dead', 'Alive'],
    out_file=None,filled=True,
    rounded=True,
    special_characters=True)
graph = graphviz.Source(f)
graph
```



Extra、提交 Kaggle 檢驗此決策樹分類準確度：



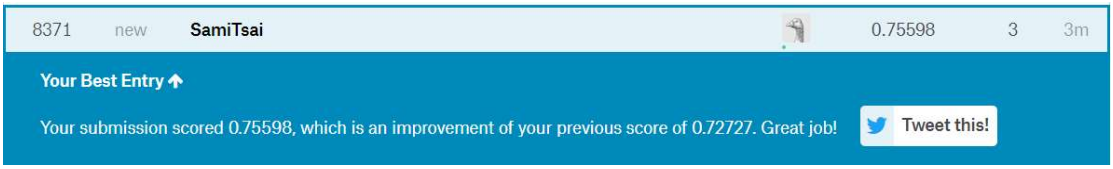
72.72%，與自行預測的 80%有很大的出入，排名更是淒慘的 9232/10144 XD 顯然還有很大進步空間；推測 Age 資料處理上使用平均數填充缺失資料的方式太為草率，以及或許可以設法充分利用其他未利用到的屬性。

Extra2、與隨機森林比較：

嘗試使用隨機森林進行分類

```
43    dt=RandomForestClassifier(random_state=2,n_estimators=250,min_samples_split=20,oob_score=True)
```

使用的特徵不變，一樣交給 Kggle 檢驗：

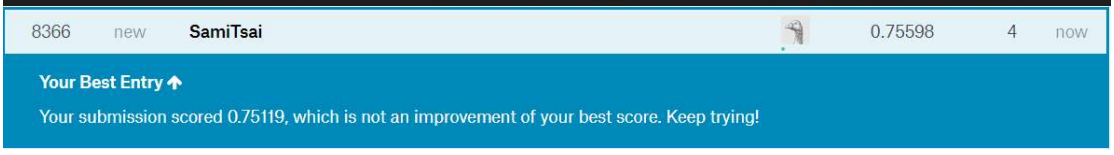


分數顯著上升至 75.59%。耳聞隨機森林較能抵抗 overfit，猜想或許除了 Age 處理手法過於粗糙外，先前決策樹的準確率偏低也有一部分是因為 overfit。

Extra3、討論 Age 屬性：

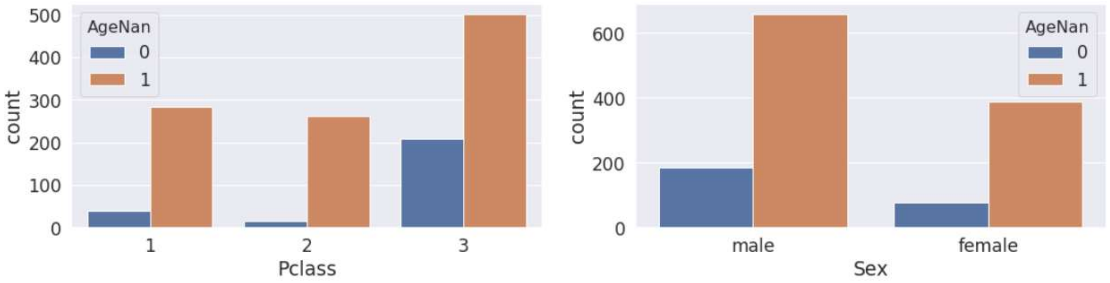
若單純使用中位數補上缺失值效果是否會比平均數好?

```
32    df_train.Age.fillna(df_train.Age.median(),inplace=True)
```



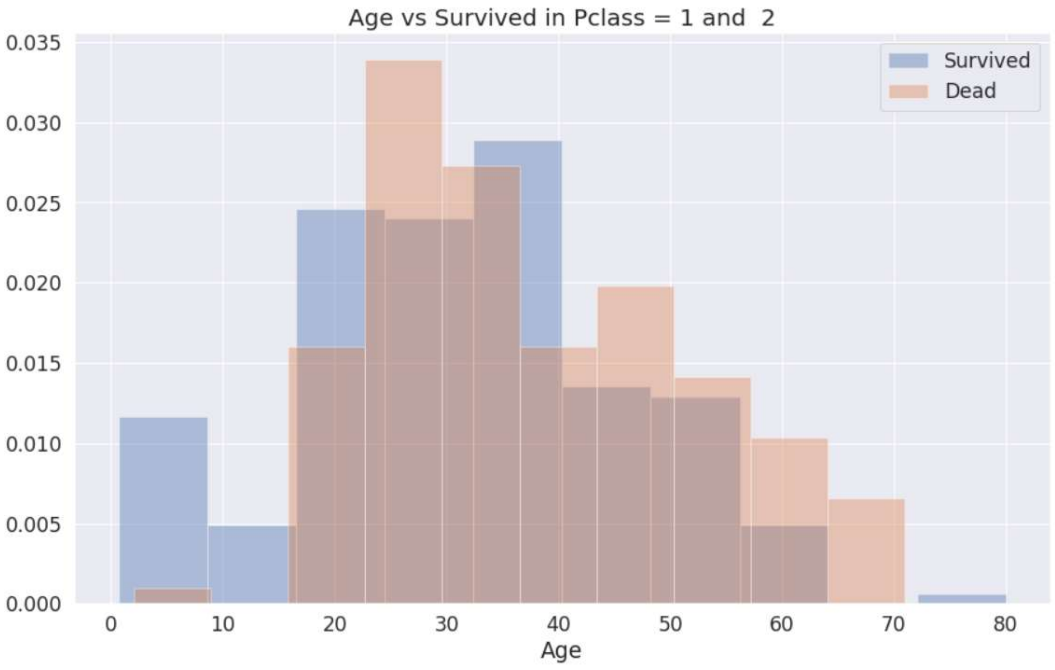
反而變得更差了，看來需要更進一步進行討論，先分別查看 Pclass 和 Sex 中 Age 屬性缺失的部分

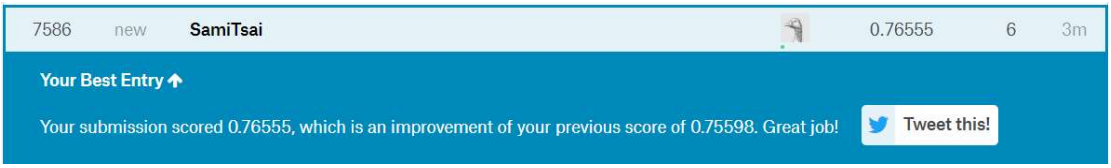
```
df_data['Has_Age'] = df_data['Age'].isnull().map(lambda x : 0 if x == True else 1)
fig, [ax1, ax2] = plt.subplots(1, 2)
fig.set_figwidth(18)
ax1 = sns.countplot(df_data['Pclass'],hue=df_data['Has_Age'],ax=ax1)
ax2 = sns.countplot(df_data['Sex'],hue=df_data['Has_Age'],ax=ax2)
pd.crosstab(df_data['Has_Age'],df_data['Sex'],margins=True).round(3)
```



從表中得知，在最低等客艙中的乘客，缺失 Age 屬性的比例明顯高出其他艙等多；而男性 Age 屬性缺失數量較女性多，但男女在缺失比例上差不多。也就是說，年齡這個屬性不太適合用來分類整體的生還率，否則在 3 等艙分類上會有較嚴重的問題。

若我們排除 3 等艙裡的資料，拿剩下的與年齡進行分析，得到一張生還與否的圖。從圖中可以觀察到，15 歲以下死亡人數趨近於零，而 15 歲以上則無法明顯歸納出一個規則；若我們額外創造一個屬性為”是否大於 15 歲”並取代 Age 進行分析，結果分數確實微幅的上升了。





最終排名停留在 7586/10165，雖然依然不甚理想，但回頭比起最初的 9232/10144 依然有了顯著的進步，若能利用其他屬性創造出合理的特徵，相信能夠再繼續進步。