Data Mining Hw0 Titanic 王廣和

載入訓練集資料

In [2]:
```
import pandas as pd
```

In [3]:
```
df_train = pd.read_csv(f'train.csv')
df_train.tail(5)
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [4]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Data的格式與含意

0 乘客ID

1 生/死

2 票種

3 姓名

4 性別

5 年齡

6 船上兄弟姊妹配偶

7 船上父母小孩

8 票ID

9 票價

10 座位編號

11 登船港口

用性別切割

男女做OHE 但只有兩個類別就做0,1

In [5]:
```
df_train['Sex'] = df_train['Sex'].map({'female' : 1,'male':0}).astype('int')
```

In [6]:
```
df_train[['PassengerId','Survived','Sex']].head(5)
```

Out[6]:

	PassengerId	Survived	Sex
0	1	0	0
1	2	1	1
2	3	1	1
3	4	1	1
4	5	0	0

In [7]:
```
female_dead = df_train[(df_train['Sex']==1) & (df_train['Survived']==0)].size
female_alive = df_train[(df_train['Sex']==1) & (df_train['Survived']==1)].size
female_alive/(female_alive+female_dead) #女性存活率
```

Out[7]:
```
0.7420382165605095
```

In [8]:
```
df_train[['Sex','Survived']].groupby(['Sex']).mean() #groupby用法
```

Out[8]:

	Survived
Sex	
0	0.188908
1	0.742038

用艙位切割

In [9]:
```
df_train[['Pclass','Survived']].groupby(['Pclass']).mean() #有錢人比較活得下來
```

Out[9]:

	Survived
Pclass	
1	0.629630
2	0.472826
3	0.242363

把訓練資料跟label分開

In [10]:
```
X = df_train.drop(labels=['Survived','PassengerId'],axis =1)
Y = df_train['Survived']
```

載入sklearn 的 randomforest

In [11]:
```
from sklearn.ensemble import RandomForestClassifier
```

使用兩個分割明顯的特性: sex & Pclass

In [12]:
```
label = ['Sex','Pclass']
Model = RandomForestClassifier(random_state = 311706009,      #隨機數seed
                               n_estimators=250,              #森林中有幾棵樹
                               min_samples_split=20,          #切割樣本數
                               oob_score=True)                #樹未使用的樣本(其他樹的樣本)作為驗證使用
Model.fit(X[label],Y)
print(Model.oob_score_)
```

```
0.7508417508417509
```

處理測試集資料

In [13]:
```
df_test = pd.read_csv(f'test.csv')
```

In [14]:
```
df_test.tail(5)
```

Out[14]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

In [15]:
```
df_test['Sex'] = df_test['Sex'].map({'female' : 1,'male':0}).astype('int')
```

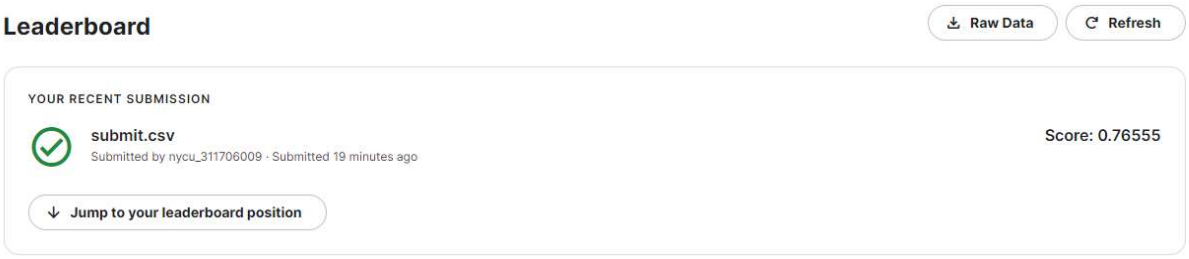
模型預測，測測看準確度

In [16]:
```
Model.predict(df_test[label])
```

Out[16]:
```
array([0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0],
      dtype=int64)
```

匯出成csv並上傳

In [18]:
```
submit = pd.read_csv('./gender_submission.csv')
rf_res =  Model.predict(df_test[label])
submit['Survived'] = rf_res
submit['Survived'] = submit['Survived'].astype(int)
submit.to_csv('submit.csv', index= False)
```



In [ ]:
```

```