


姓名:王佑恩 學號:108601205 系級:電機2A

上課成果

12382


YouEnW



0.75598

1

1s



Your First Entry!

Welcome to the leaderboard! Your score represents your submission's accuracy. For example, a score of 0.7 in this competition indicates you predicted Titanic survival correctly for 70% of people.

What next? You've got a few options:

- 👉 Learn skills that can improve your score in our [Intro to Machine Learning course by Dan Becker](#).
- 🔍 Check out [the discussion forum](#) to find lots of tutorials and insights from other competitors.
- 🏆 Find a new challenge by entering one of our [open, active competitions](#) or searching our [public datasets](#).

指標	分數
Precision	0.8279569892473119
Recall	0.7264150943396226
Accuracy	0.832089552238806
Kaggle	0.75598

在開始做其他實驗之前, 先對目前的生存預測做幾點分析:

1. 捨棄之資料:Pclass、Name、Ticket、Cabin
2. Age目前是選用各個性別之中位數來填補缺失資料。
3. Sex選用是否為男生來做分析(沒影響)
4. Embarked用統計最多的項目Embarked_S來彌補缺失資料。
5. 目前沒有用到Sibsp及Parch與生存率的關聯來做預測。
6. 目前使用Logistic regression model做預測。

註:做每個實驗前皆有把上個實驗的修改調為上課成果的狀態。

改善之實驗

實驗一：改變random_state


模型程式碼：

```
60 y = df['Survived']
61 # split to training data & testing data
62 from sklearn.model_selection import train_test_split
63 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=99)
64 # using Logistic regression model
```

說明：為隨意想到的測試，random state指的是該組隨機數的編號，在需要重複試驗的時候，保證得到一組一樣的隨機數。

結果分析：

YOUR RECENT SUBMISSION



for_submission_20220317.csv
Submitted by YouEnW · Submitted just now

Score: 0.76555

↓ Jump to your leaderboard position

指標	分數
Precision	0.651685393258427
Recall	0.6170212765957447
Accuracy	0.75
Kaggle	0.76555

碰巧測出的結果比原本更好，推測可能的原因是更動random state使得訓練的資料數受到更動，進而使得資料預測完整性提高。但其他三個指標的分數大為下降，顯示此實驗的模擬準確率其實不高，使得模擬參考價值下降。

實驗二：drop Age

模型程式碼：

```
46 df.drop(['Age'], axis=1, inplace=True)
47
```

測試程式碼：

```
19 df_test.drop('Sex_female', axis=1, inplace=True)
20 df_test.drop('Pclass', axis=1, inplace=True)
21 df_test.drop('Age', axis=1, inplace=True)
```


說明：因為年齡原本有缺失資料，透過填補才使得資料完備。但我覺得這樣可能會影響到原本的預測結果，所以試試看先把年齡捨棄掉，觀察年齡對生存率預測的影響有多大，藉此決定是否要對年齡缺失資料的填補方式做修正。

結果分析：

指標	分數
Precision	0.8020833333333334
Recall	0.7264150943396226
Accuracy	0.8208955223880597
Kaggle	0.76555

10882


YouEnW



0.76555

5

1s



Your Best Entry!

Your most recent submission scored 0.76555, which is the same as your previous score. Keep trying!

觀察實驗結果，Kaggle分數提高，且Recall指標的分數沒有改變，但Precision及Accuracy的分數皆下降，代表把年齡捨棄掉會對預測準確率有一定程度的影響，所以不適合把年齡捨棄掉，接著我們換一種方式來填補年齡的缺失資料。

實驗三：把Age填補方式由中位數改成用眾數填補

模型程式碼：

```
48 # Age缺失值男生就用男生的眾數、女生就用女生的眾數來填補
49 df['Age'].fillna(df['Age'].mode()[0],inplace=True)
50 df.apply(lambda x: sum(x.isnull()),axis=0)
51 df.isnull().sum()
```

測試程式碼：


```
11 df_test['Age'].fillna(df_test['Age'].mode()[0],inplace=True)
```

說明：原本年齡的缺失資料是用各個性別的「中位數」來填補，但我覺得上郵輪旅遊，應該也可能是很多同齡朋友一起出遊，所以想試試用上課學到的「眾數（最高頻率值）」來填補。

結果分析：

指標	分數
Precision	0.8279569892473119
Recall	0.7264150943396226
Accuracy	0.832089552238806
Kaggle	0.75837

YOUR RECENT SUBMISSION



for_submission_20220317.csv

Submitted by YouEnW · Submitted a few seconds ago

Score: 0.75837

↓

Jump to your leaderboard position

Kaggle的分數只有略為提高，可能是此實驗對生存率預測的影響並不大。

實驗四：把Age用家庭的個別狀況取中位數來填補缺失值
模型程式碼：

```
54 # 創造新的變數：家庭人數
55 df['Family'] = df['SibSp'] + df['Parch'] + 1
56
57 Survival_Rate = df[['Family', 'Survived']].groupby(by=['Family']).agg(np.mean)*100
58 Survival_Rate.columns = ['Survival Rate(%)']
59 Survival_Rate.reset_index()
60 print(Survival_Rate)
61 # 將Family做級別區分
62 df['Family Class'] = np.nan
63 df.loc[ df.Family==0, 'Family Class' ] = 2
64 df.loc[ (df.Family>=1) & (df.Family<=3), 'Family Class' ] = 3
65 df.loc[ (df.Family>=4) & (df.Family<=6), 'Family Class' ] = 2
66 df.loc[ (df.Family>=7), 'Family Class' ] = 1
67 # 用Age個別情況的中位數來填補缺失值
68 table = df.pivot_table(values='Age', index='Family Class', columns='Sex', aggfunc=np.median)
69 def fage(x):
70     return table.loc[x['Family Class'], x['Sex']]
71 df['Age'].fillna(df.apply(fage, axis=1), inplace=True)
```

測試程式碼：

```
14 # 創造新的變數：家庭人數
15 df_test['Family'] = df_test['SibSp'] + df_test['Parch'] + 1
16
17 # 將Family做級別區分
18 df_test['Family Class'] = np.nan
19 df_test.loc[ df_test.Family==0, 'Family Class' ] = 2
20 df_test.loc[ (df_test.Family>=1) & (df_test.Family<=3), 'Family Class' ] = 3
21 df_test.loc[ (df_test.Family>=4) & (df_test.Family<=6), 'Family Class' ] = 2
22 df_test.loc[ (df_test.Family>=7), 'Family Class' ] = 1
23 # 用Age個別情況的中位數來填補缺失值
24 table = df_test.pivot_table(values='Age', index='Family Class', columns='Sex', aggfunc=np.median)
25 def fage(x):
26     return table.loc[x['Family Class'], x['Sex']]
27 df_test['Age'].fillna(df_test.apply(fage, axis=1), inplace=True)
```

說明：

我覺得填補Age缺失值應該還有更為適合的方式，所以想試著用上郵輪的乘客組成來分析可能的Age缺失值。

首先，參考網路的做法，把SibSp(有多少兄弟姊妹/配偶在船上)及Parch(有多少父母/小孩在船上)合成Family，並把Family分成三個級別建立Family Class。接著結合上課所教取得個別情況中位數的方法，把Family Class跟Sex依性別以及Family Class的三個級別分成六類，算出個別中位數使用該中位數來填補。

此方法的概念是在分析各個家庭的組成成員，年齡中位數通常會隨著家庭成員人數越多而越大，可依下表(table)看出此現象。並將Age缺失值填補的更加謹慎。

Interactive-1.interactive > table (3, 2)

	Family..	female	male
0	1	13.5	9
1	2	23	10
2	3	28	30

結果分析：

指標	分數
Precision	0.8260869565217391
Recall	0.7169811320754716
Accuracy	0.8283582089552238
Kaggle	0.75598

[for_submission_20220317.csv](#)

0.75598

2 minutes ago by YouEnW

把Age用家庭的個別狀況取中位數來填補缺失值

Kaggle分數以及其他三個指標相較於一開始沒有太大的變動，跟我預期的結果不太一樣，著實有點小失望，但經過多次實驗下來可以發現，修改Age的填補方式，能增加的預測準確率有限，所以需要再試試看修改別的模擬資料。

實驗五: drop Fare

模型程式碼：

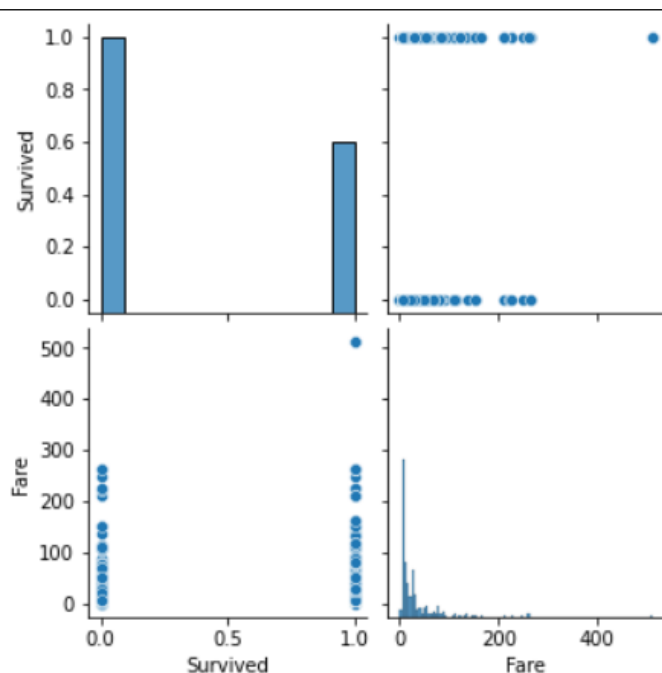
```
73
74 df.drop(['Fare'], axis=1, inplace=True)
```

測試程式碼：

```
19 df_test.drop('Sex_female',axis=1,inplace=True)
20 df_test.drop('Pclass',axis=1,inplace=True)
21 df_test.drop('Fare',axis=1,inplace=True)
```


說明:分析Fare跟生存率的關聯，觀察圖表可發現，兩者關聯十分參差不齊，只有一個票價500以上的確存活下來，其餘票價無法與生存率掛上十足的相關性，所以想實驗看看把票價捨棄會對生存率預測有什麼影響。


```
35 sns.pairplot(df[['Survived','Fare']], dropna=True)
```



結果分析：

指標	分數
Precision	0.8229166666666666
Recall	0.7452830188679245
Accuracy	0.835820895522388
Kaggle	0.76794

9286 YouEnW  0.76794 22 1s

 Your Best Entry!
Your most recent submission scored 0.76794, which is the same as your previous score. Keep trying!

把Fare捨棄掉的結果，可以看出各個指標的分數皆提高，雖然捨棄一個完整的資料有點可惜，但卻得出了目前最高的Kaggle分數。

實驗六：不要把Pclass drop掉

模型程式碼：

```
65 x = df.drop(['Survived'],axis=1)
66 y = df['Survived']
```

測試程式碼：

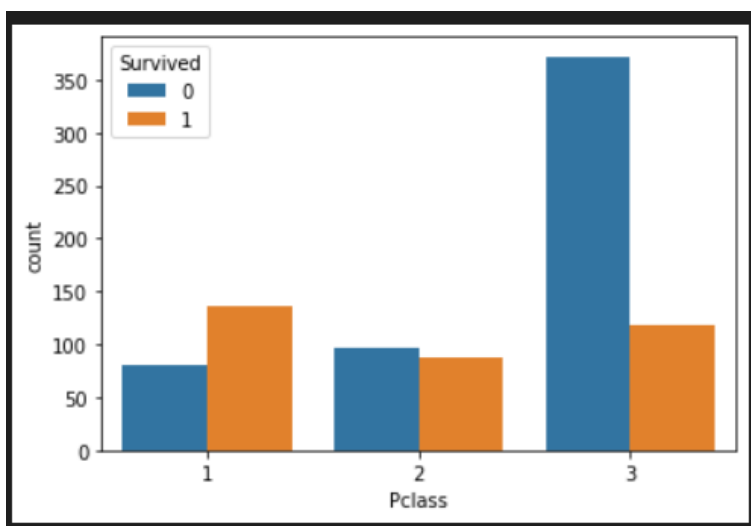
```
19 df_test.drop('Sex_female',axis=1,inplace=True)
20 # df_test.drop('Pclass',axis=1,inplace=True)
```

說明：我認為艙等（設1、2、3為頭等艙、商務艙、經濟艙）還是會影響到存活率，畢竟高級郵輪的頭等艙一般都設在中高層的位置，會比較奢華，逃生設備一定相對完善；而經濟艙一般都設在底層的位置，逃生設備相對匱乏。

設各個艙別的生存率為： $(\text{該艙總生存人數})/(\text{該艙總人數})$

由下方圖表可看出，頭等艙的生存率為3個艙等中最高的，進而判斷我的假設成立。

```
33 sns.pairplot(df[['Survived','Pclass']], dropna=True)
34 sns.countplot(df['Pclass'], hue=df['Survived'])
```




結果分析：

指標	分數
Precision	0.7959183673469388
Recall	0.7358490566037735
Accuracy	0.8208955223880597
Kaggle	0.76794

9315


YouEnW



0.76794

13

1s



Your Best Entry!

Your most recent submission scored 0.76794, which is an improvement of your previous score of 0.76555. Great job!

[Tweet this](#)

Kaggle分數也達到我目前的最高分，明顯Precision的分數掉下來了，說明有較多預測為死亡但實際為存活的資料。但對於Accuracy指標還是有相當的分數，所以Pclass對存活率的影響還是有的，將其保留並做預測是一個不錯的選擇。

實驗七：把Age填補方式由中位數改成用眾數填補，保留Pclass，drop掉Fare
模型程式碼：

```
46 # 用Age各性別的眾數來填補缺失值
47 df['Age'].fillna(df['Age'].mode()[0],inplace=True)
48 df.apply(lambda x: sum(x.isnull()),axis=0)

76 # 把Survived, Fare丟掉
77 x = df.drop(['Survived','Fare'],axis=1)
78 y = df['Survived']
```

測試程式碼：

```
11 df_test['Age'].fillna(df_test['Age'].mode()[0],inplace=True)

19 df_test.drop('Sex_female',axis=1,inplace=True)
20 # df_test.drop('Pclass',axis=1,inplace=True)
21 df_test.drop('Fare',axis=1,inplace=True)
```

說明：實驗三、五與六合起來做預測，純粹是想把之前所做的實驗合起來試試看。

結果分析：

指標	分數
Precision	0.7676767676767676
Recall	0.7169811320754716
Accuracy	0.8022388059701493
Kaggle	0.77272

8221

YouEnW



0.77272

24

1m



Your Best Entry!

Your most recent submission scored 0.77272, which is an improvement of your previous score of 0.76794. Great job!

Tweet this

所得到的Kaggle分數又創新高，但其他三個指標分數都下降，顯示此次實驗的預測較為不準確，有較多的預測錯誤。

結論

經過以上實驗，可推斷出三件事情，首先是Age對Kaggle分數的影響度很低；其次，Kaggle分數跟另外三個指標分數並不成正相關；最後，所考慮的資料越多，預測的準確率不一定會增加。若能再學精一點，對其他數據做更有效的處理，應該能使Kaggle分數再提高。

還能再實驗的項目有以下幾項：

1. 將名字做分類，將其與家庭做關聯，進而分析此關聯與生存率的影響。
2. 改用不同的預測模型做預測。
3. Ticket與Cabin都可能影響到乘客的活動範圍，進而影響其生存率。

心得

雖然最後沒辦法把Kaggle的分數拉到0.8以上，但透過多次的實驗、debug、詢問老師和學長以及搜尋資料，還是學到了不少東西。因為對語法很不熟悉，所以整個打程式的過程花了很多時間理解上課資料以及自己設想，每次要操作設想好的實驗時，都會卡在程式碼的問題。雖說過程真的有點辛苦，但我學的蠻開心的，也真的是有學到東西。

參考資料

1. 鐵達尼生存預測
-https://aifreeblog.herokuapp.com/posts/64/Data_Analytics_in_Practice_Titanic/
2. Kaggle競賽-鐵達尼號生存預測
-<https://yulongtsai.medium.com/https-medium-com-yulongtsai-titanic-top3-8e64741cc11f>
3. 課堂講義P03、P04