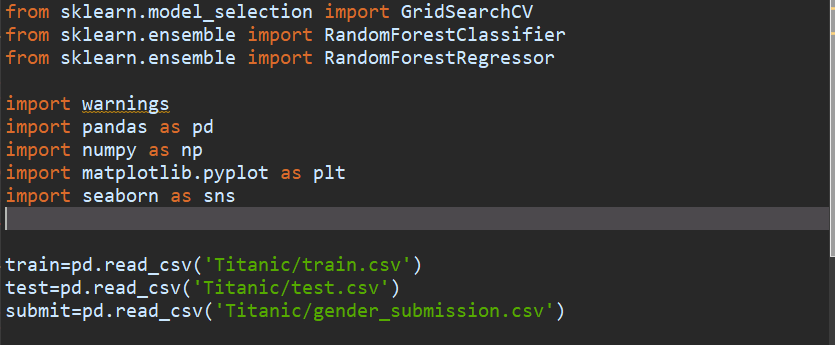
0853422 資管碩一 范姜鈞 HW0

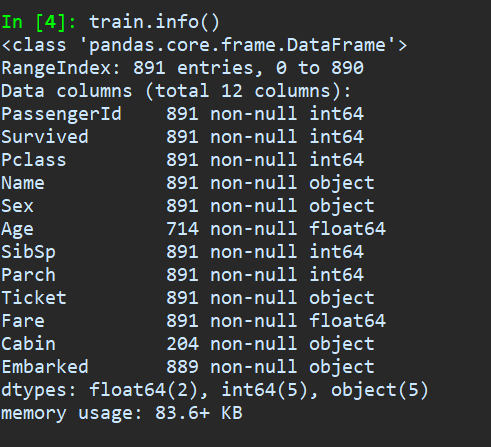
Titanic

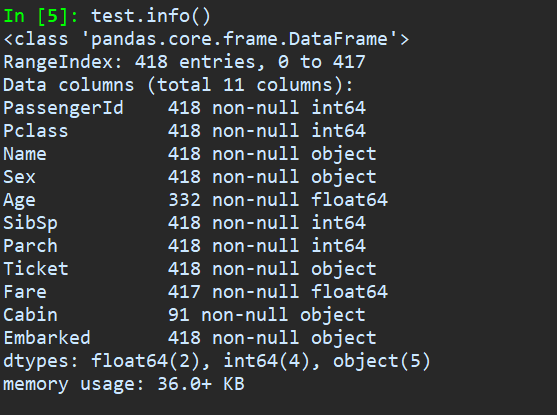
1. 讀取資料:

載入所需套件與資料



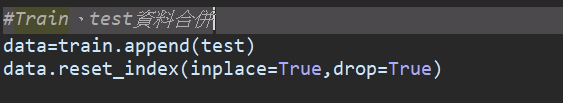
1. 資料前處裡:
2. 使用Info()函式觀察train及test資料中是否有空值



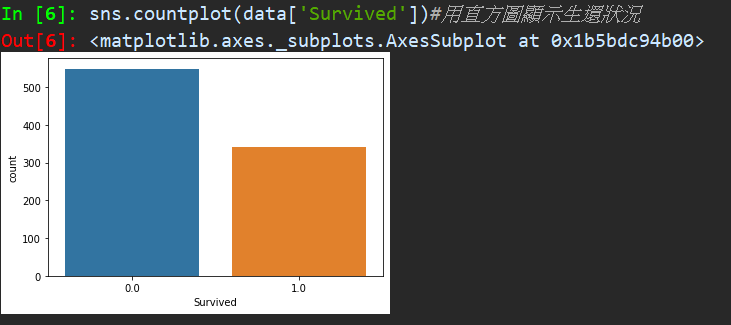


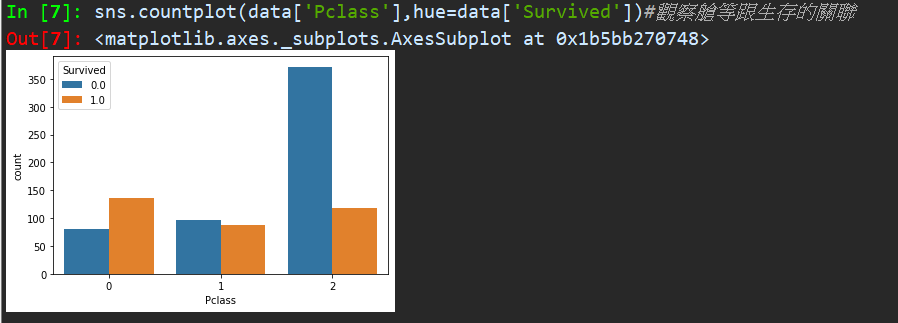
發現train中的Age、Cabin、Embarked欄位中有空值、test中的Age、Fare、Cabin欄位有空值，需要在後面做處裡

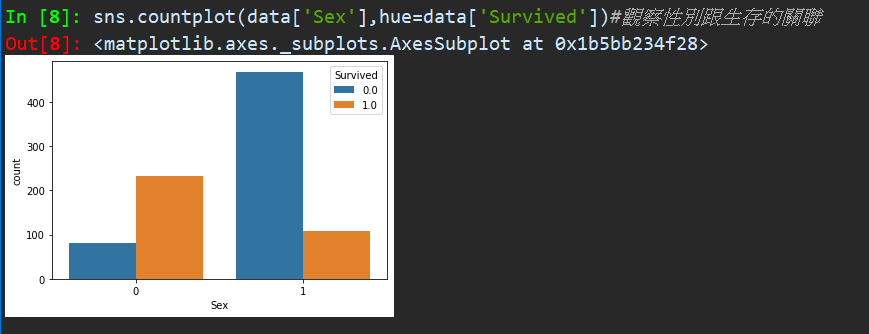
1. 由於要對整體資料做觀察，先做train與test資料的合併，並重新調整index

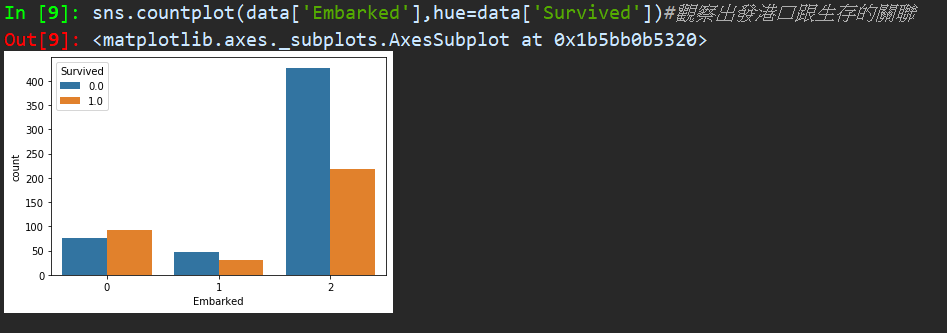


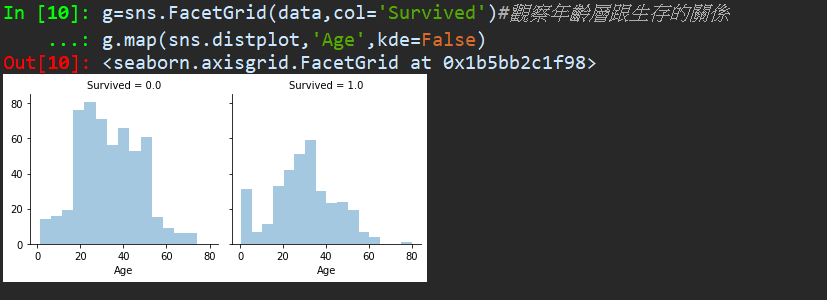
1. 資料分析

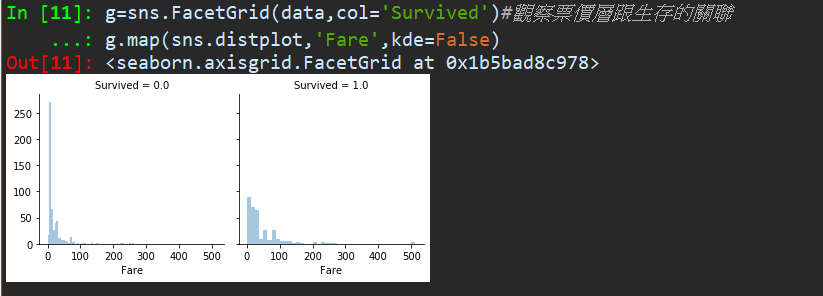


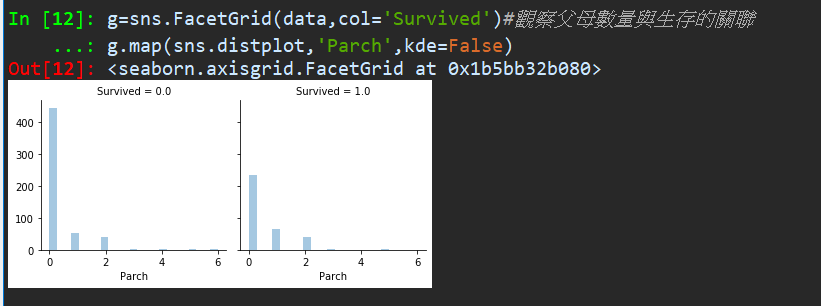


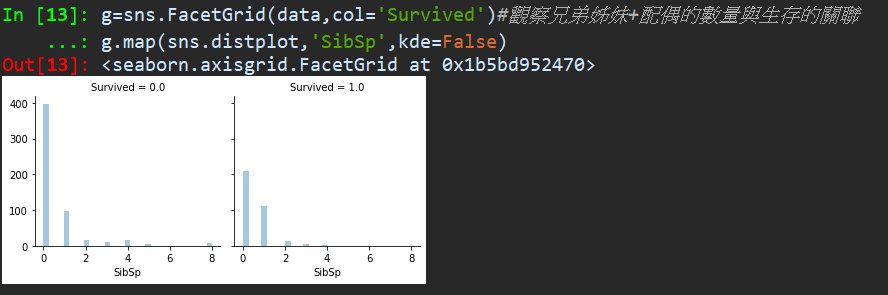


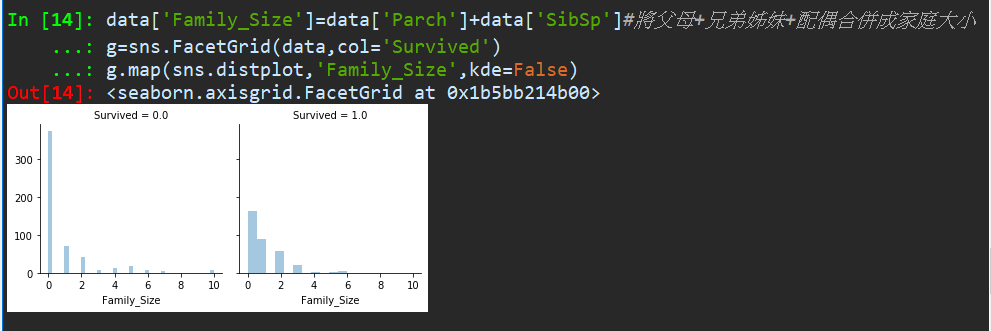




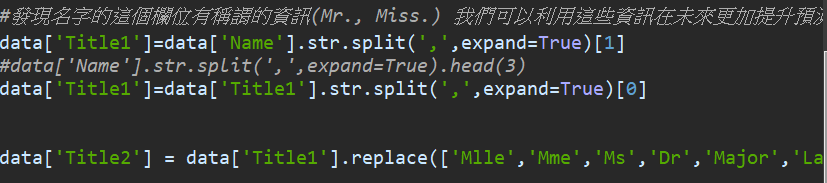








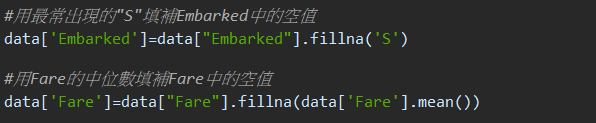
發現名字的這個欄位有稱謂的資訊(Mr., Miss.) 我們可以利用這些資訊在未來更加提升預測的準確度



把票號的資訊取出前面英文的部分，因為相同的英文代碼可能代表的是房間的位置，後面的號碼沒有意義所以省略，如果只有號碼的票號就用X來表示

data['Ticket\_info'] = data['Ticket'].apply(lambda x : x.replace(".","").replace("/","").strip().split(' ')[0] if not x.isdigit() else 'X')

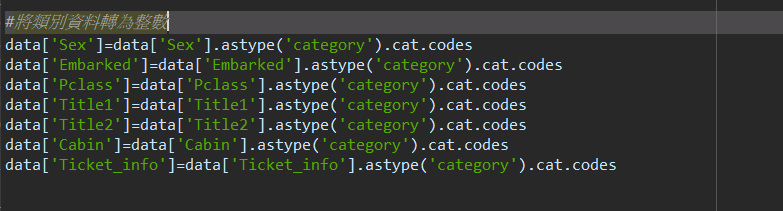
填補Embarked與Fare中的空值



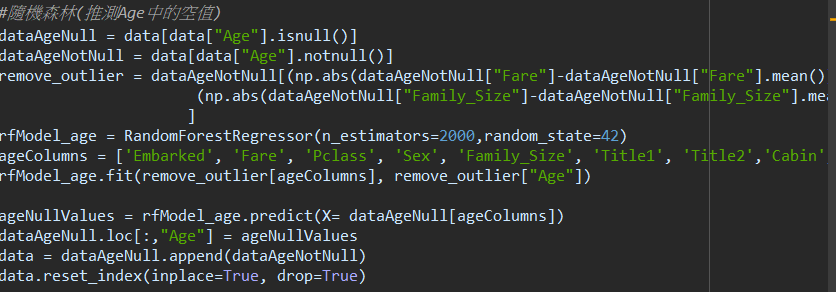
觀察Cabin的資料後，只取出最前面的英文字母，剩下的用NoCabin來表示



將類別資料轉為整數



利用隨機森林演算法推測Age中空的值



1. 跑模型預測結果

利用隨機森林演算法來預測存活率

dataTrain = data[pd.notnull(data['Survived'])].sort\_values(by=["PassengerId"])

dataTest = data[~pd.notnull(data['Survived'])].sort\_values(by=["PassengerId"])

dataTrain = dataTrain[['Survived', 'Age', 'Embarked', 'Fare', 'Pclass', 'Sex', 'Family\_Size', 'Title2','Ticket\_info','Cabin']]

dataTest = dataTest[['Age', 'Embarked', 'Fare', 'Pclass', 'Sex', 'Family\_Size', 'Title2','Ticket\_info','Cabin']]

rf = RandomForestClassifier(criterion='gini',

n\_estimators=1000,

min\_samples\_split=12,

min\_samples\_leaf=1,

oob\_score=True,

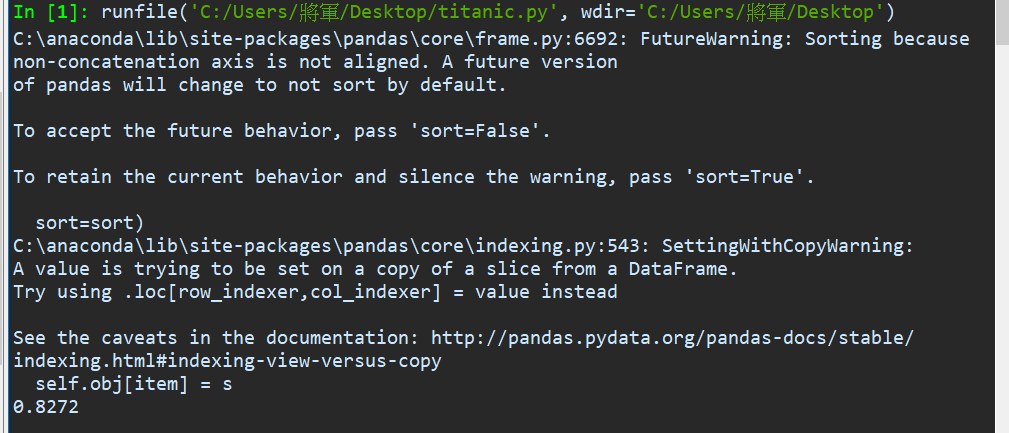
random\_state=1,

n\_jobs=-1)

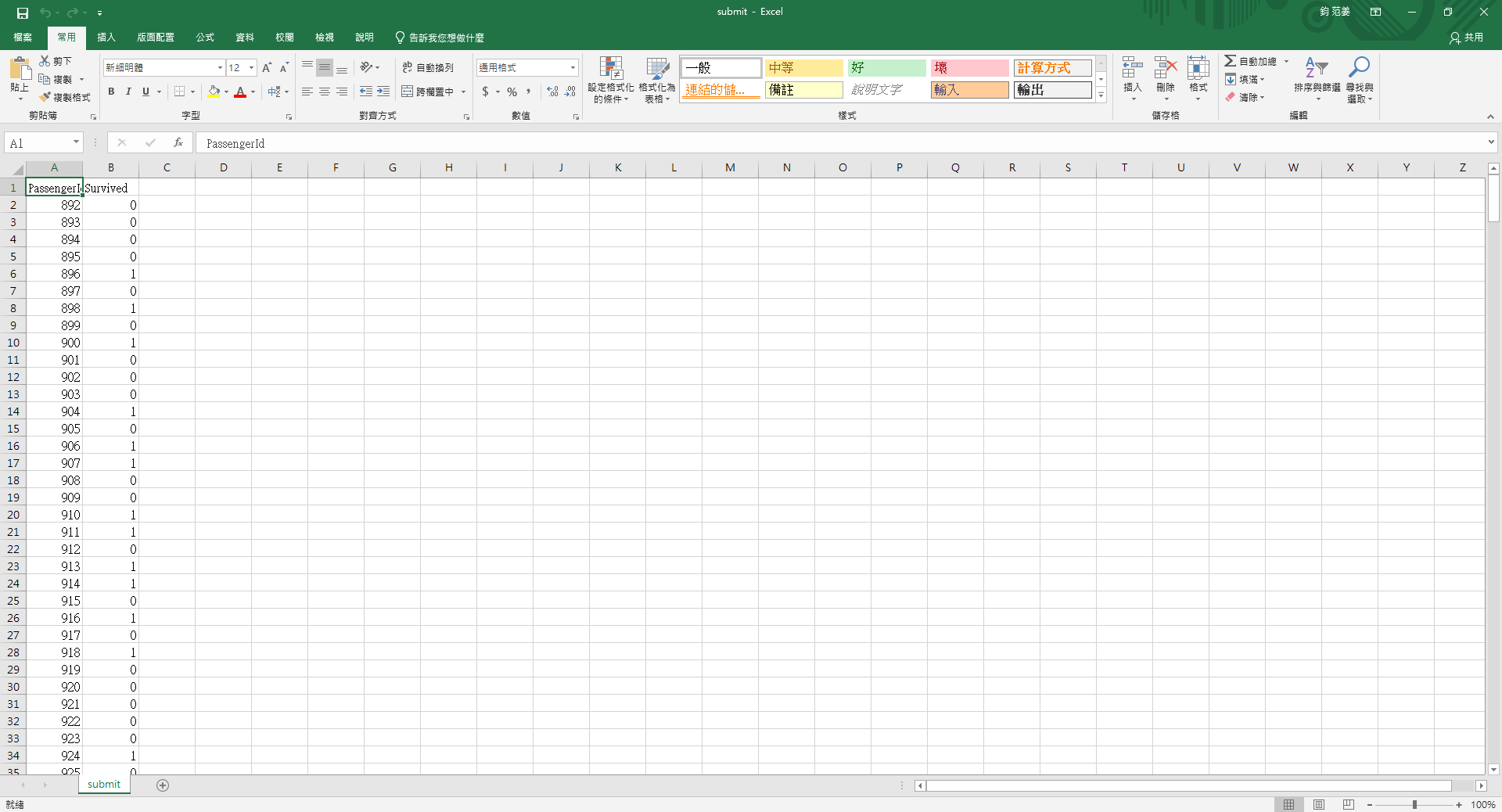
rf.fit(dataTrain.iloc[:, 1:], dataTrain.iloc[:, 0])

print("%.4f" % rf.oob\_score\_)

預測結果: 0.8272



預測結果的csv檔:



1. 上傳kaggle:

