**前言**

鐵達尼號的預測是許多機器學習中的範本，這次作業我參考的文章來自〝Kaggle競賽-鐵達尼號生存預測‑(前16%排名)〞這篇文章，而關於註解的部分我幾乎都打在python的程式碼裡面，此文章僅會有小部分的補充，如果有造成助教的不便，請多包涵！(PS：後面有附上上傳至kaggle的證明)

**欄位信息**

這次的feature總共有11個，以下作一些簡短的說明：

PassengerId：乘客ID

Pclass：乘客等級(1/2/3等艙位)

Name：姓名

Sex：性別

Age：年齡

SibSp：在船上同為兄弟姐妹或配偶的數目

Parch：在船上同為家族的父母及小孩的數目

Ticket：船票信息

Fare ：票價

Cabin：客艙

Embarked：登船港口

而訓練集又會比測試集多一項也就是我們要預測的〝生存與否〞。

**Base Model**

在我們開始測試各式各樣的特徵之前，我們必須先選一個模型，當然我們盡量選用對抗噪聲較強的模型(SVM，KNN，隨機森林)，基本上的考量是這樣的：

(1) 廣泛性

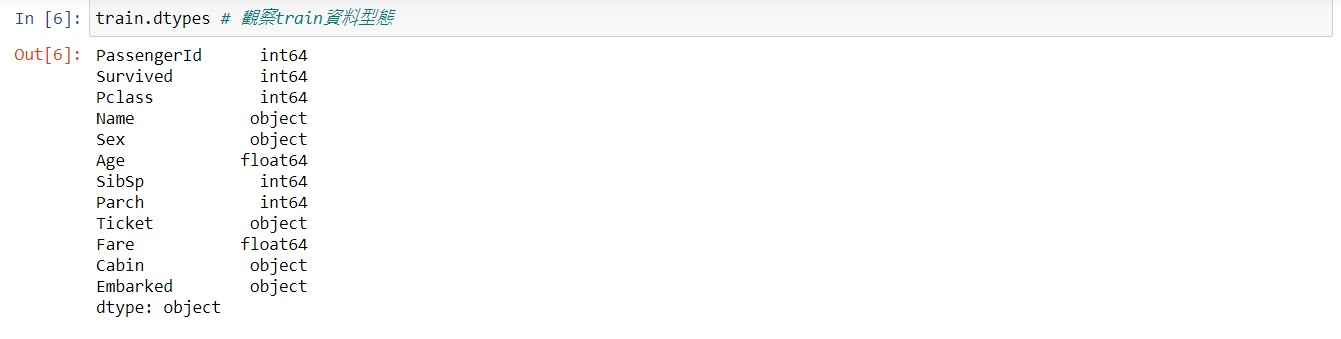
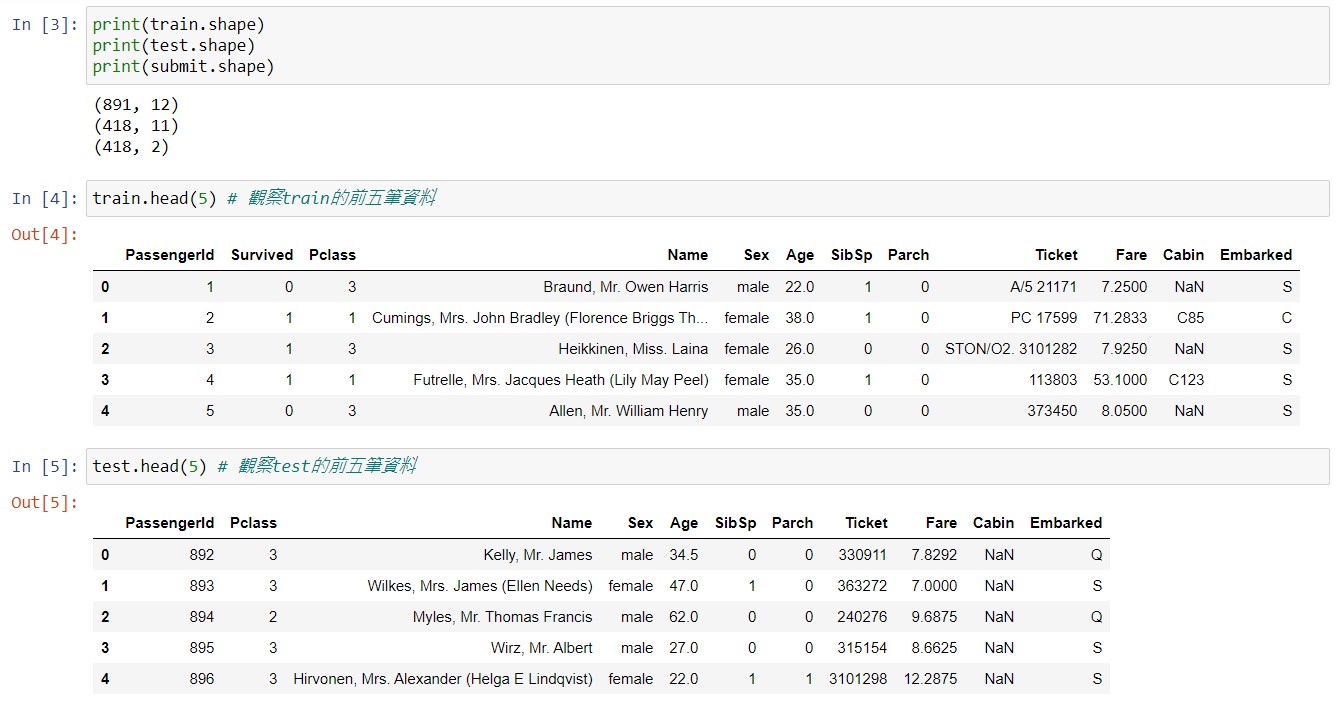
鐵達尼號的資料集非常小僅有891筆資料，因此SVM及KNN較為人詬病的問題"在較大的資料集非常沒有效率"也會因為資料集小獲得紓解，然而，隨機森林由於其平行化計算的特質在資料集小或是大時的運算效能都不錯。因此在廣泛性上隨機森林得一分。

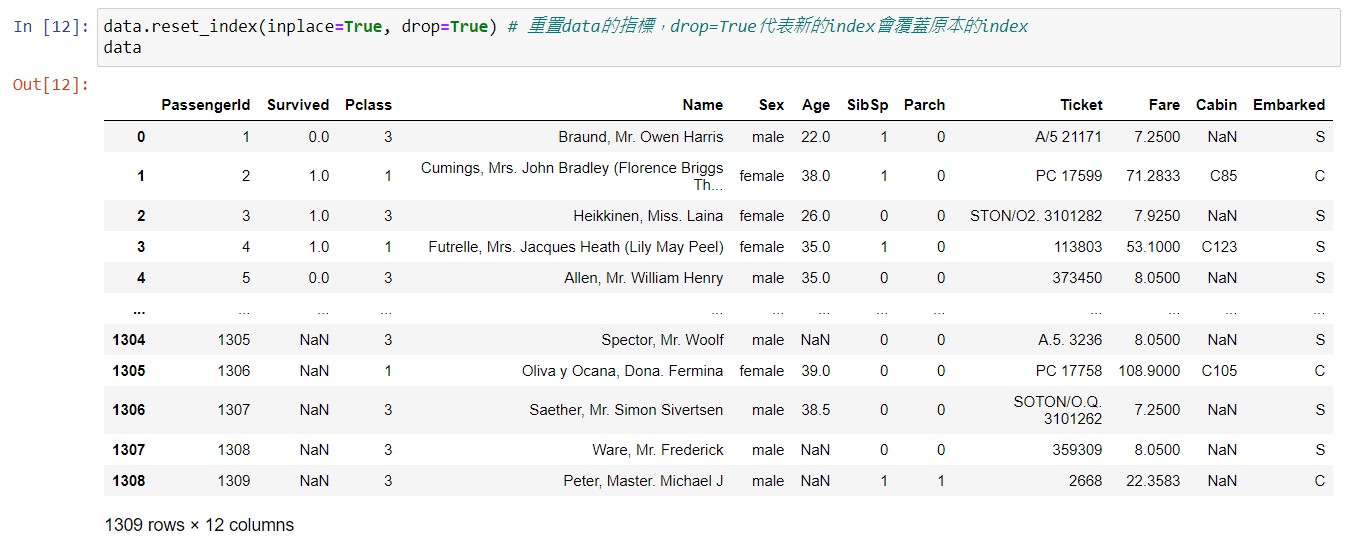
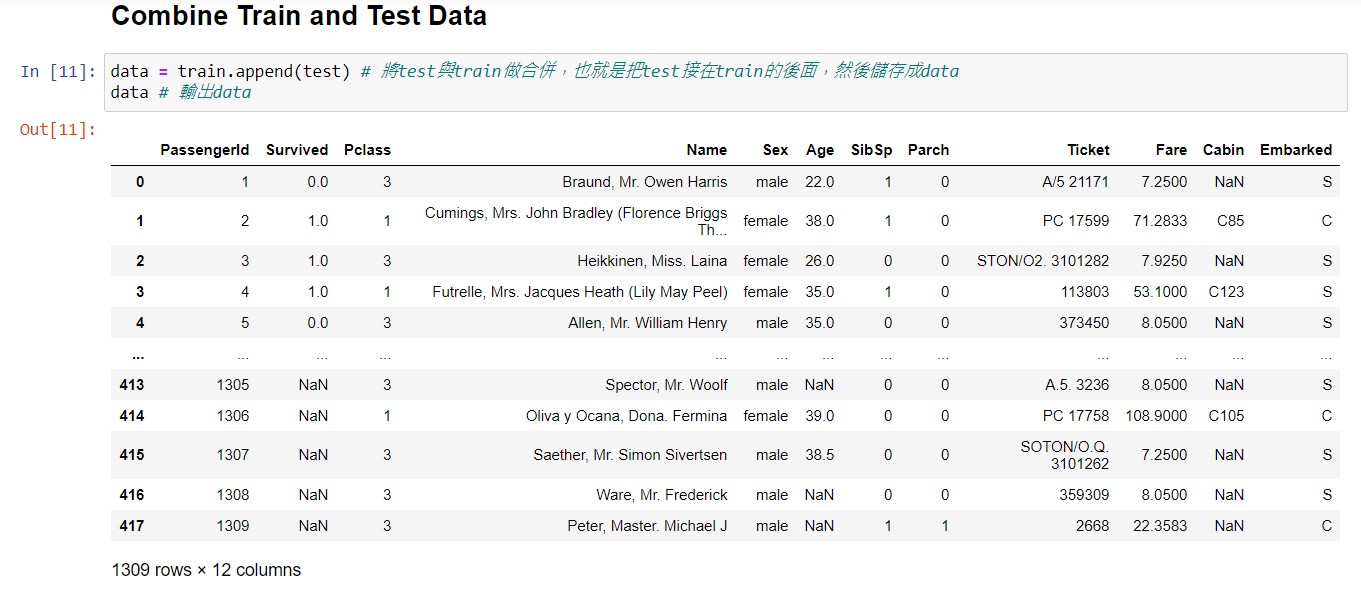
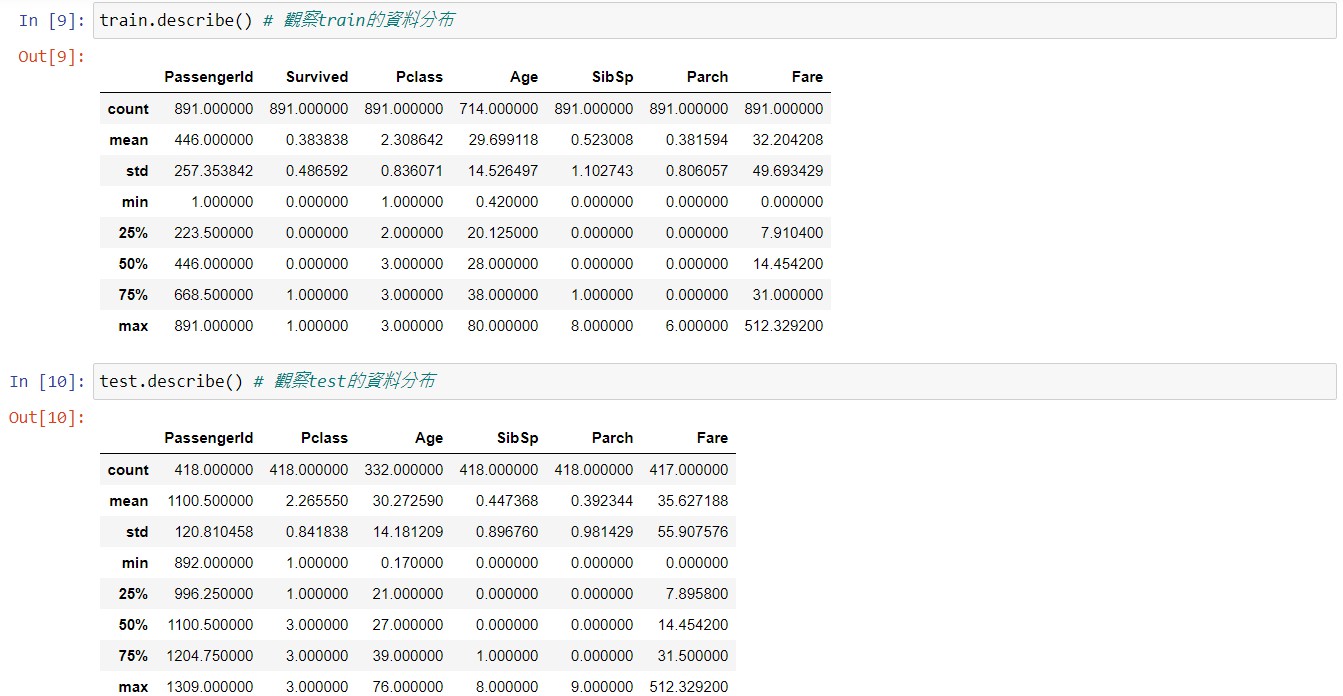
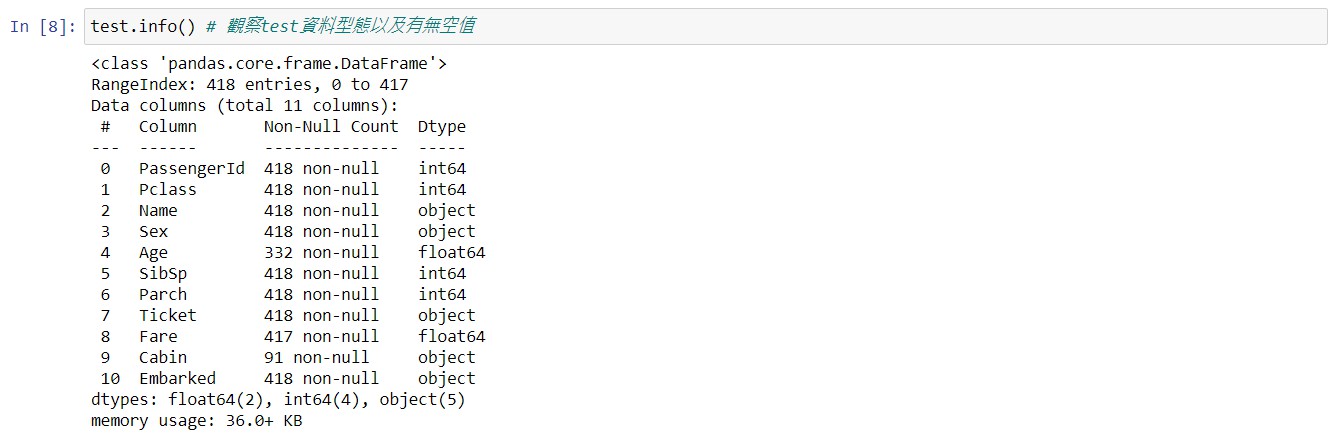
(2) 預處理

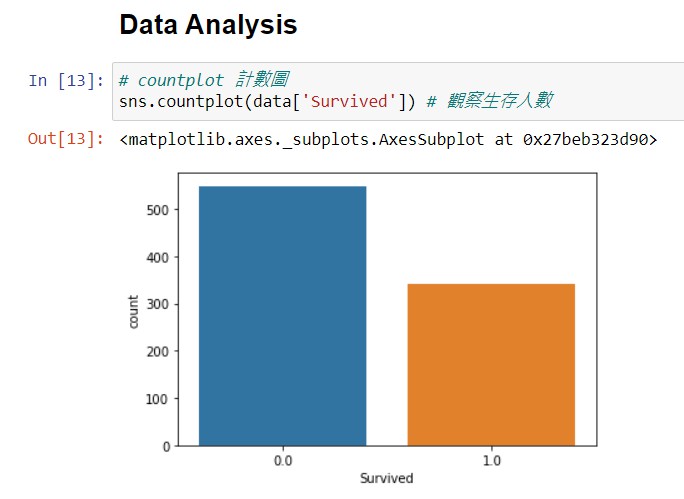
資料在餵進模型之前必須先經過預處理過程，SVM及KNN是以距離為基本來做超平面切分/鄰近投票，其中的手續比較多，也增加處理不好的風險，另一方面，隨機森林是以不純度函數來切分樣本，因此不需要歸一化或標準化，簡化了建模時的步驟。

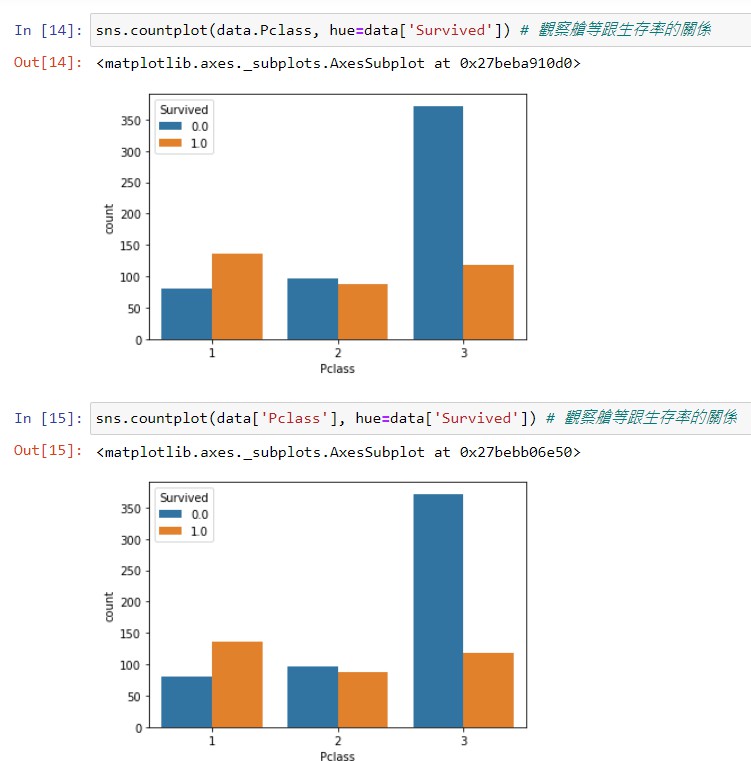
**Python程式碼截圖**

****

****

**使用info()函式觀察train以及test資料是否有空值，發現train的Age,Cabin,Embark有空值以及Test的Age, Fare, cabin有空值的情況，之後我們要想辦法來補這些空值。**

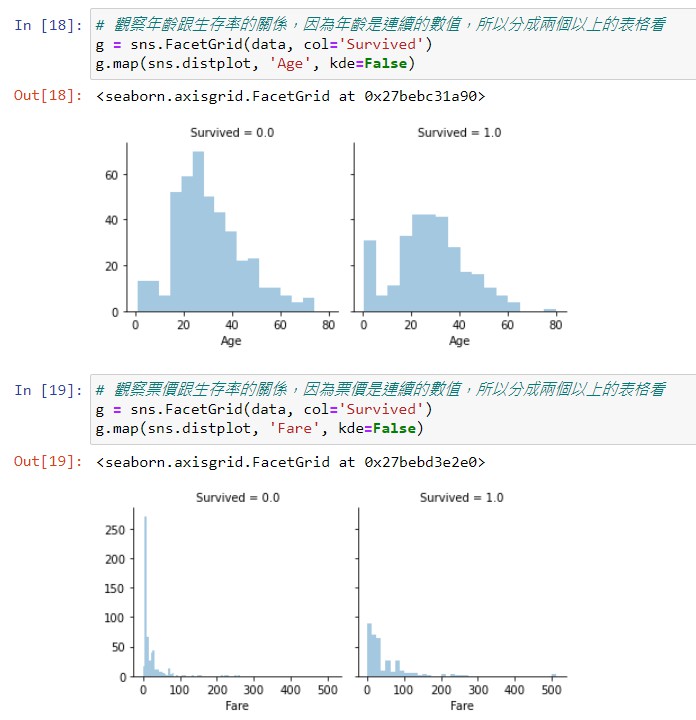
****

**發現大概死亡的比例是6成、生存的比例大概是4成**

**發現在1艙等的生存率最高、再來是2艙等、最後是3艙等的**

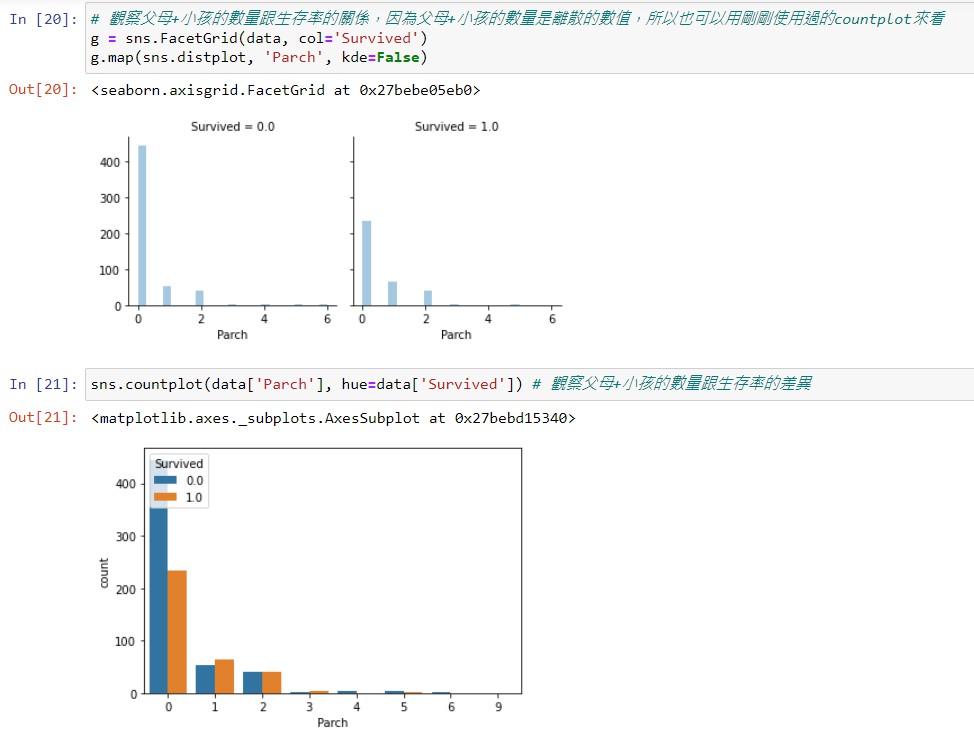
**發現女生生存率是男生的好幾倍**

**發現S港出發的都比較容易死亡**

****

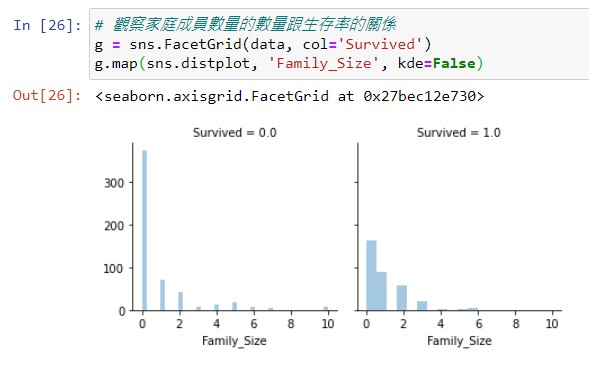
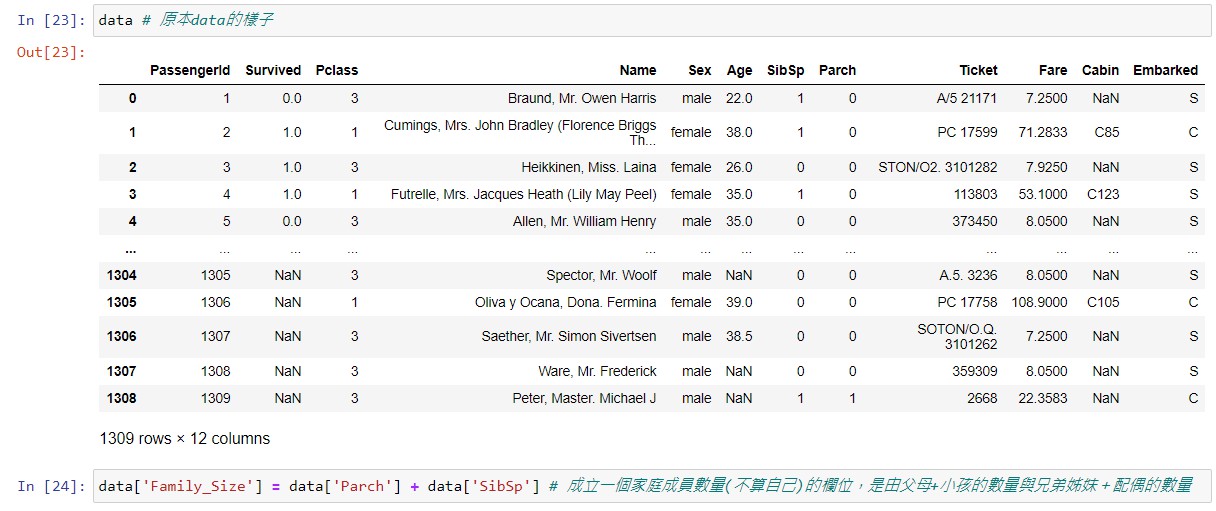
**發現年齡小的存活比例高出許多**

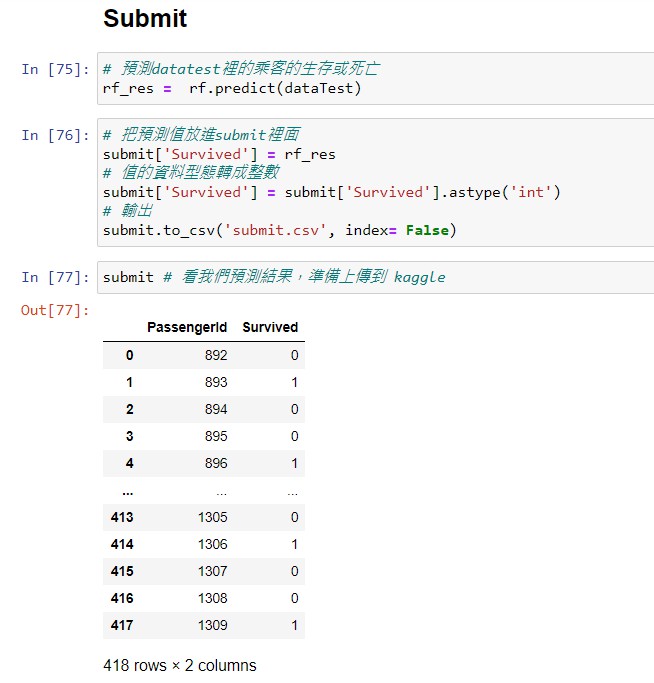
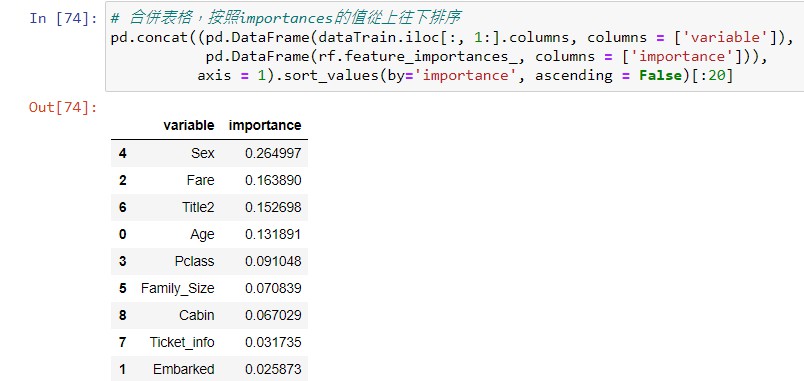
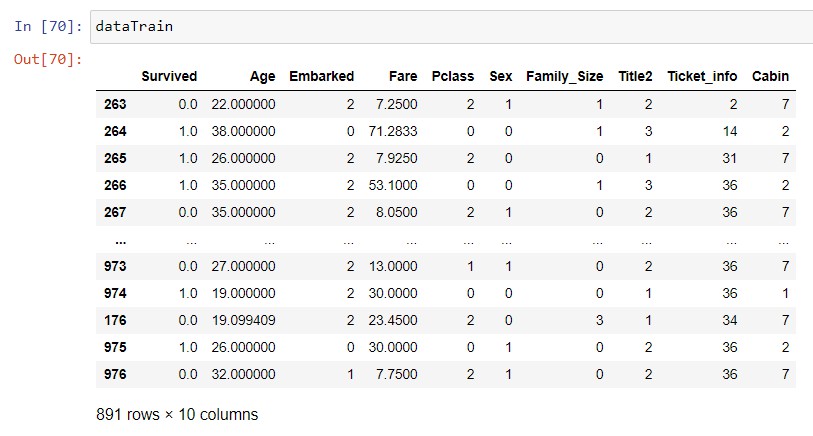
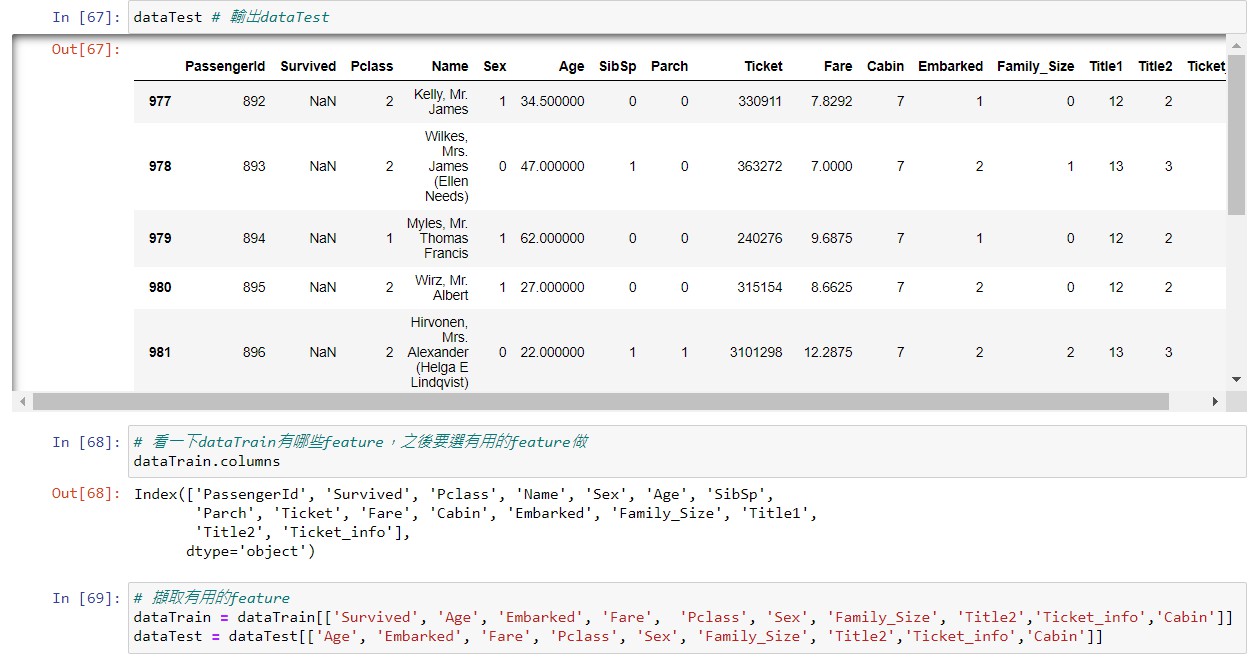
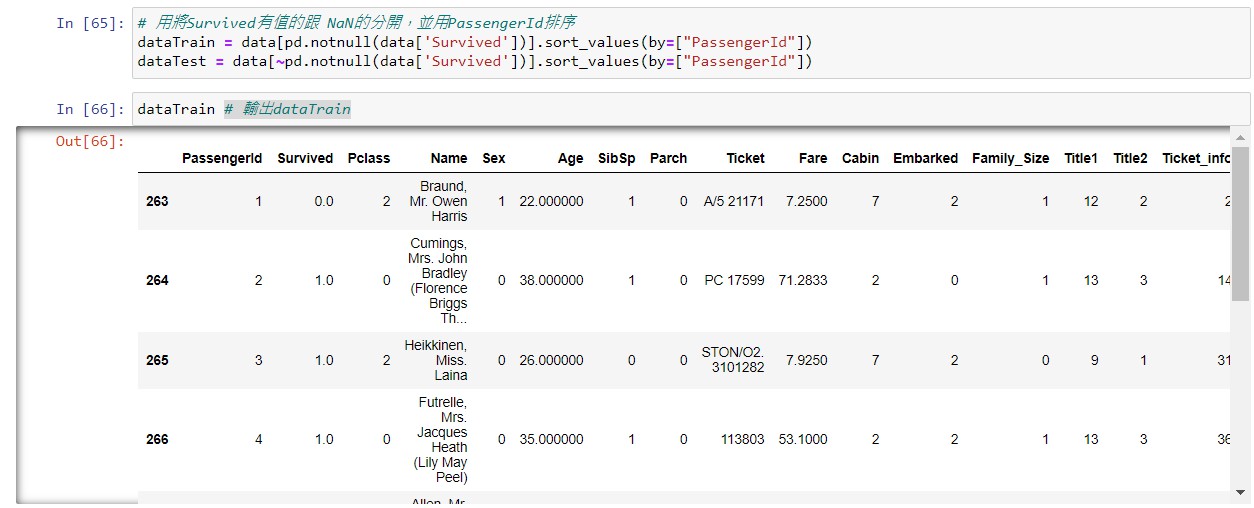
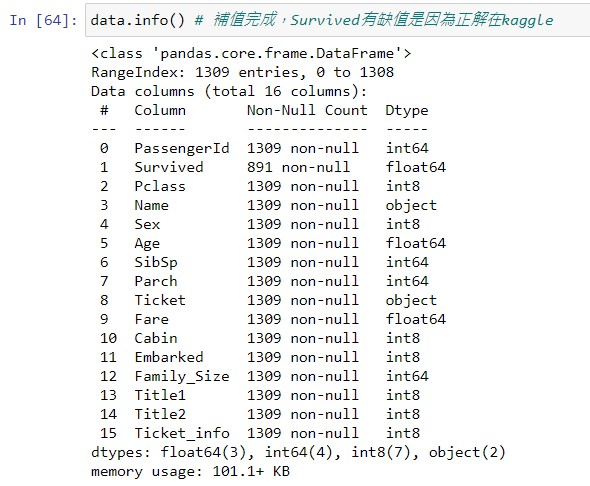
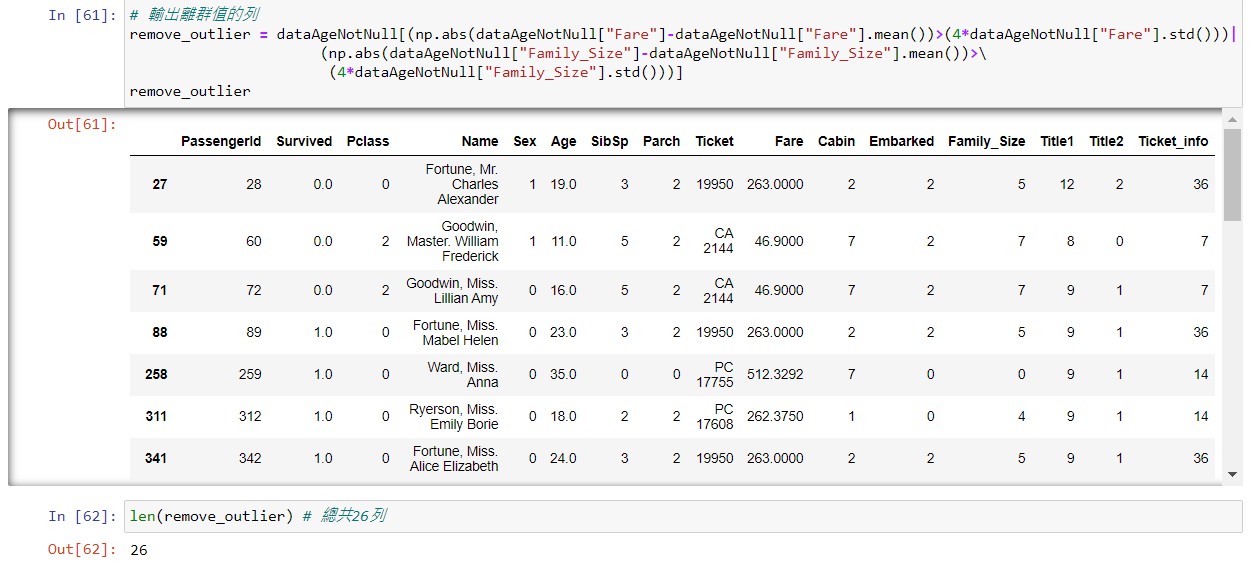
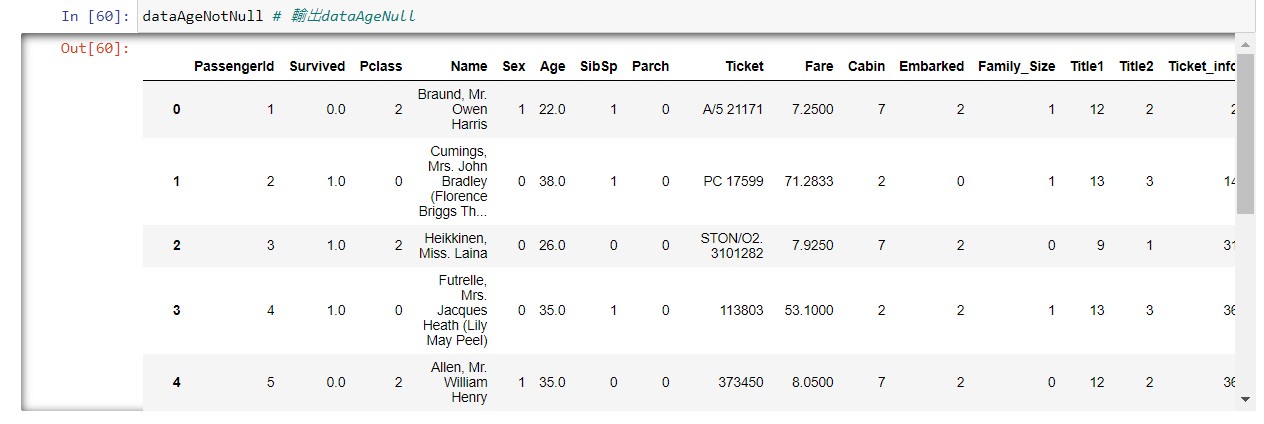
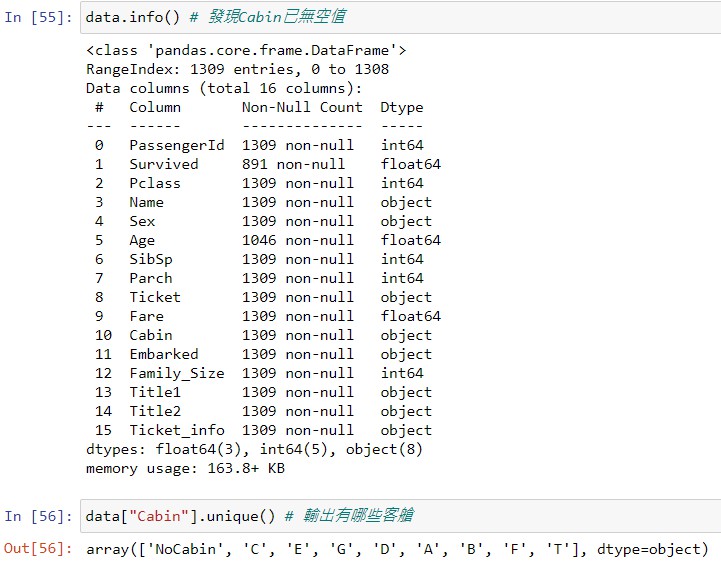
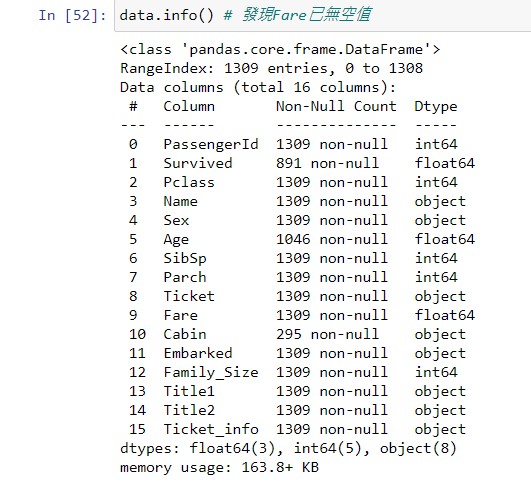
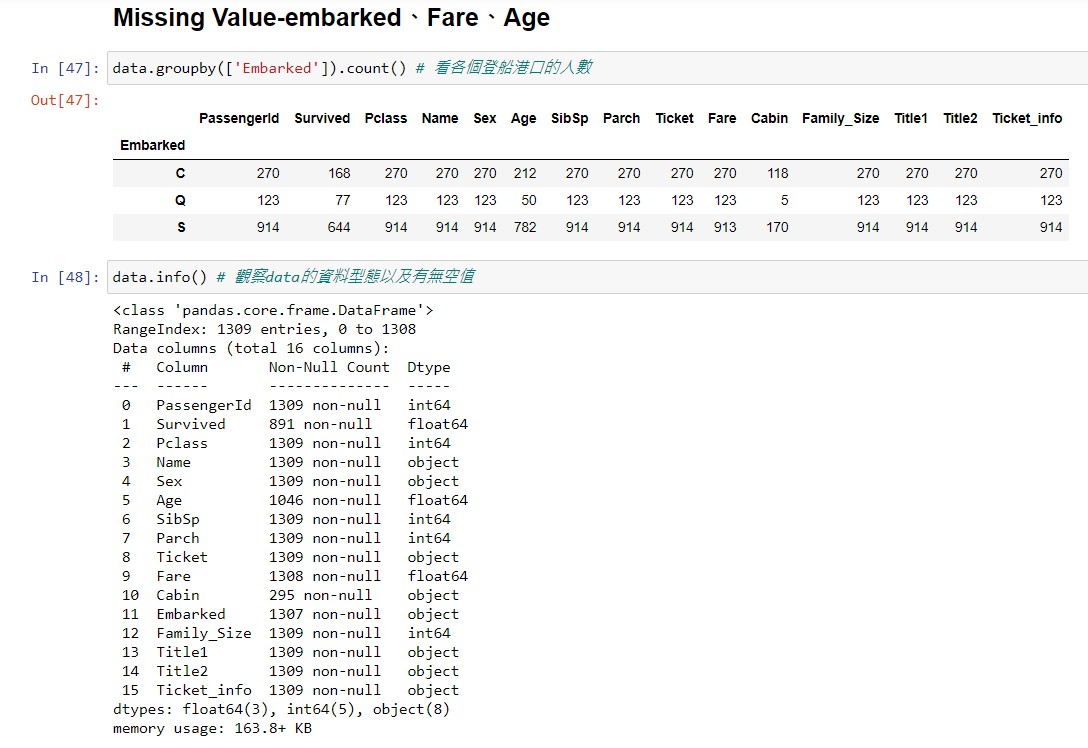
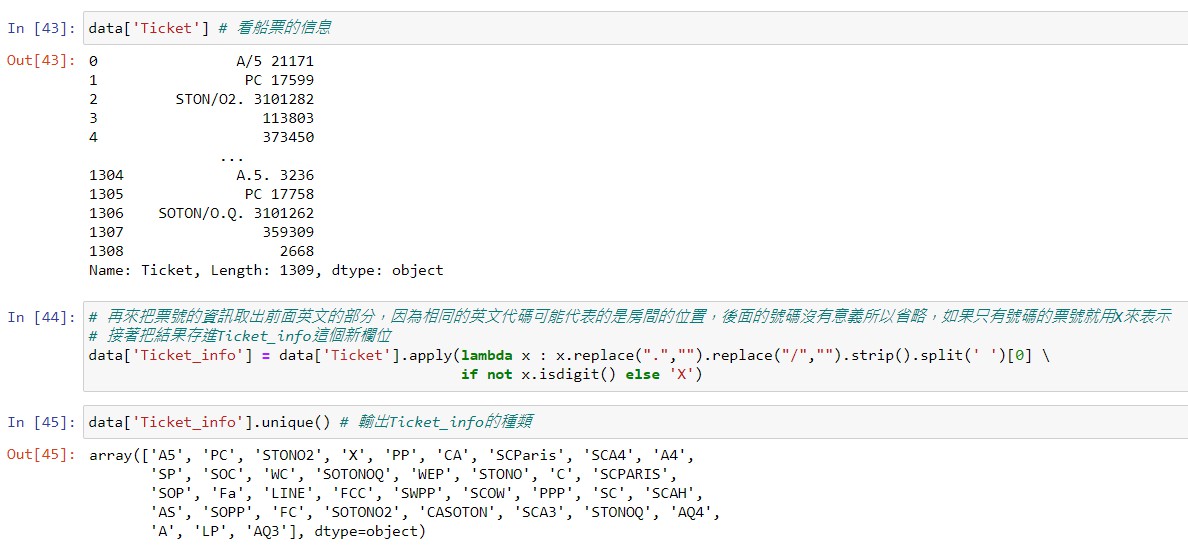
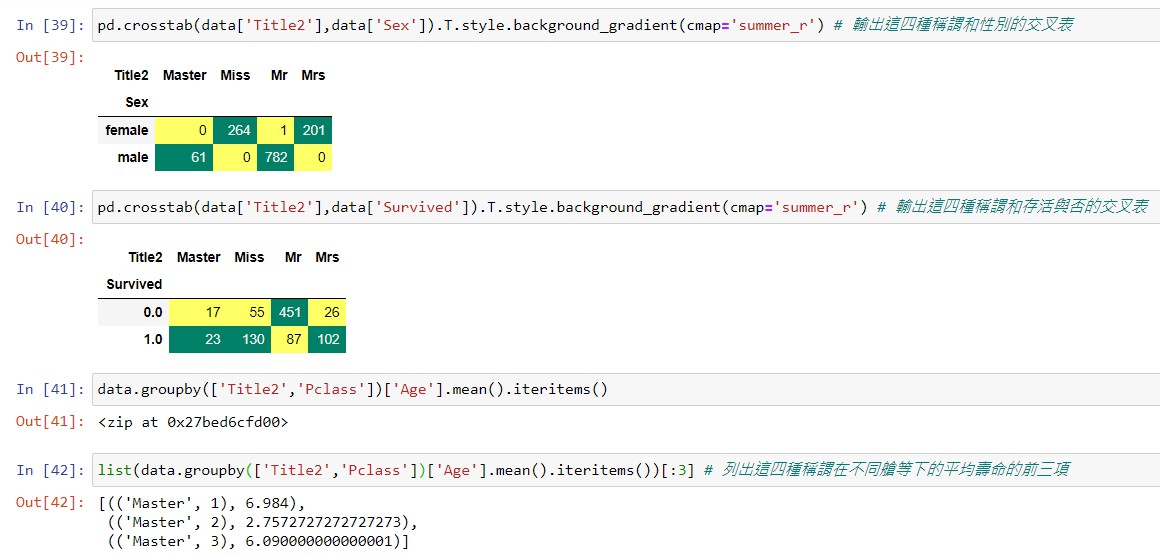
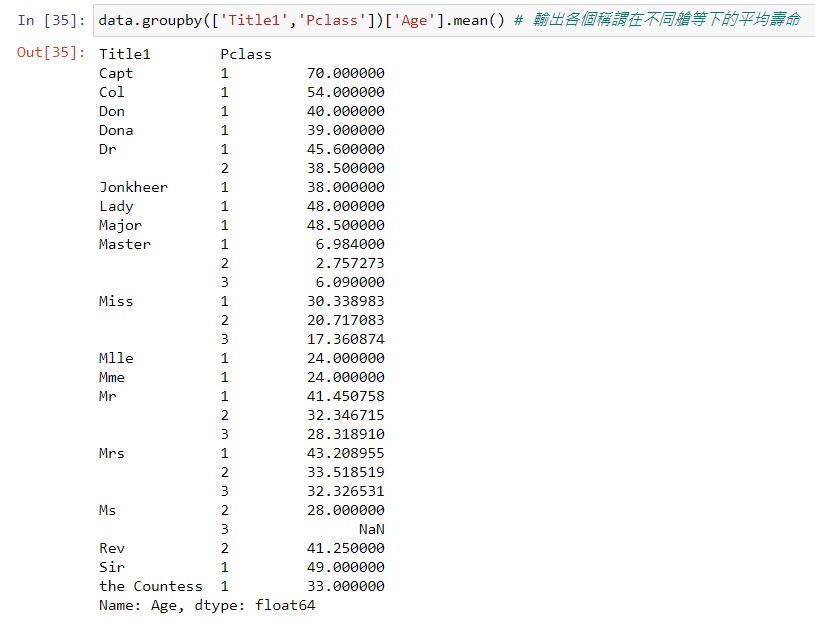
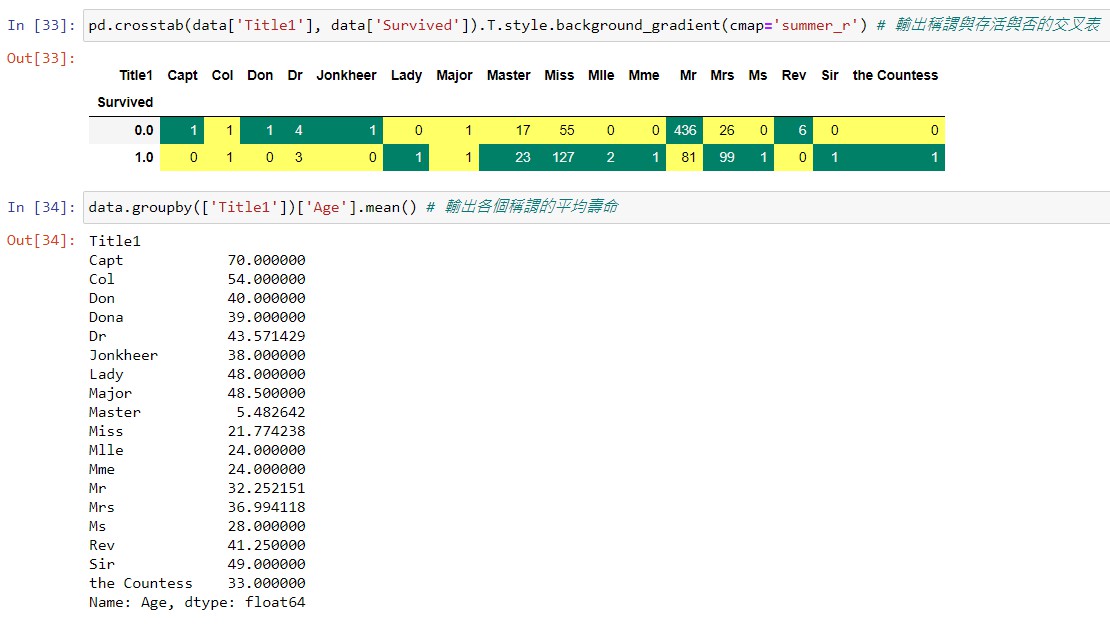
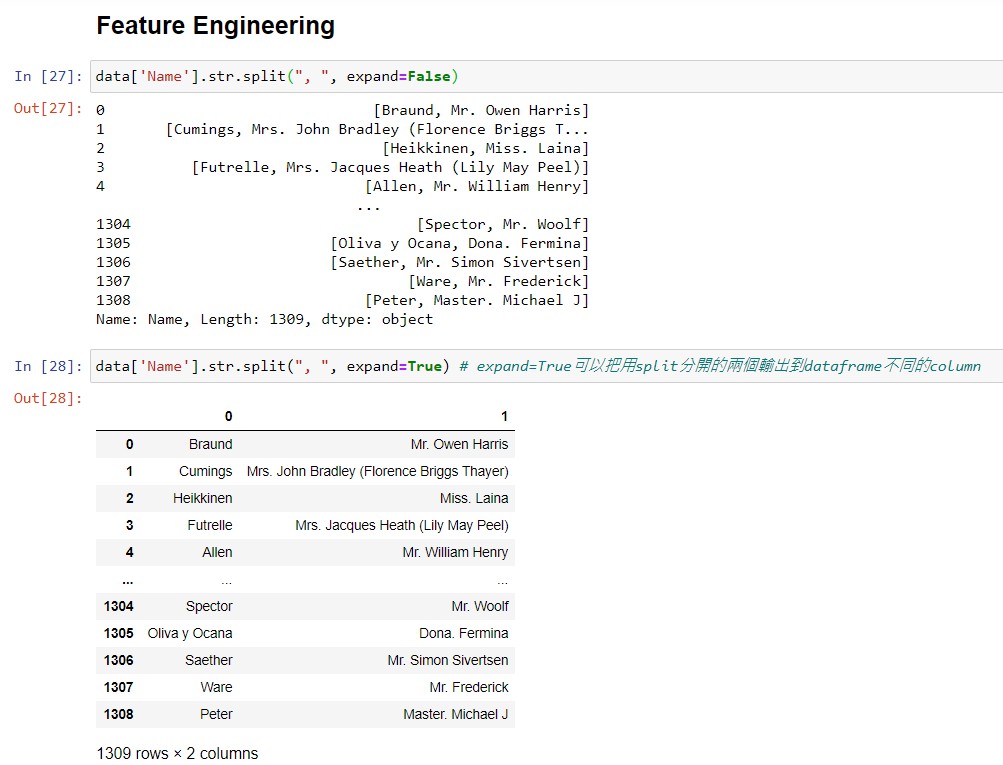
**發現票價低的乘客死亡率高出許多**

****

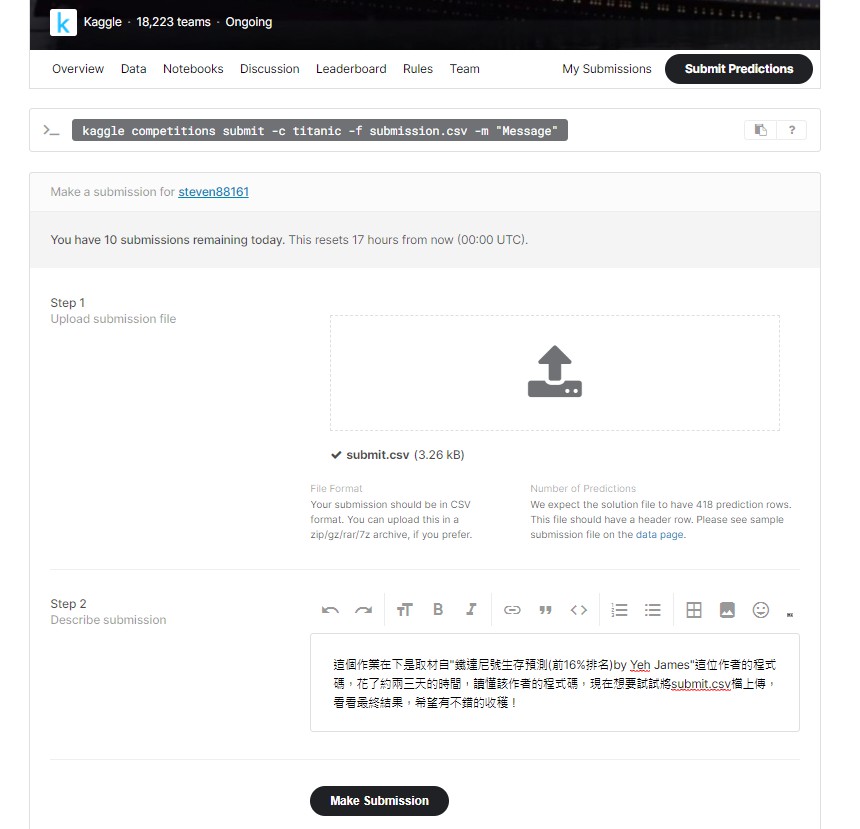
**發現沒有跟父母小孩一起來的生存率比起有跟父母小孩來的低**

****

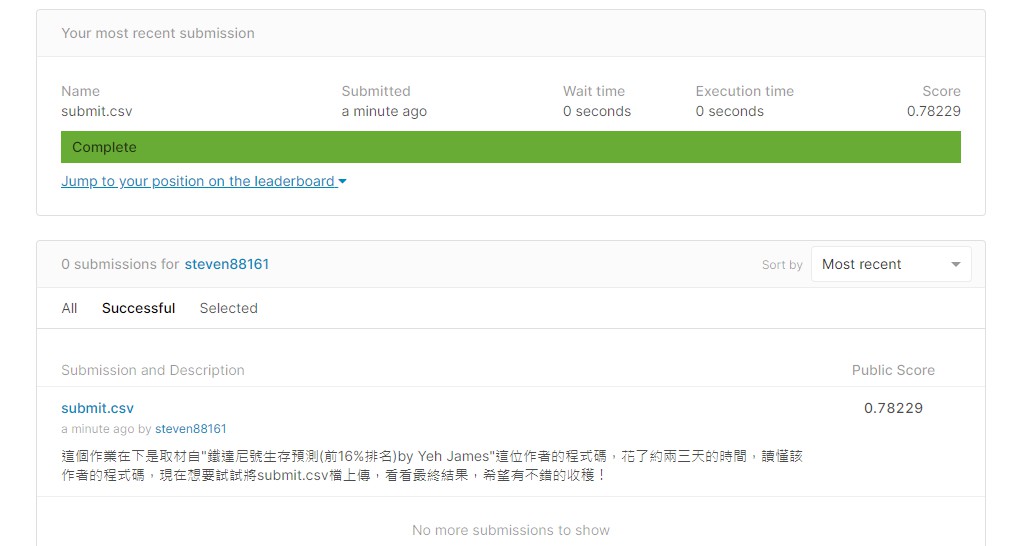
**發現沒有帶兄弟姊妹＋配偶一起來的生存率比起有跟兄弟姊妹＋配偶來的低。接下來要來處理之前提到一些特徵，像是姓名這個欄位的資料就不能直接拿來用，但如果直接丟掉是一種資訊的浪費，所以要作處理。**

****

**附上上傳至kaggle的證明**

****

**發現最後的準確率大概是78%左右**

****

**結語**

鐵達尼號的作業雖然不是很複雜，但還是著實的花了不少的心思，主要是有很多不熟悉的語法，所以不會的問題都有請教GOOGLE大神，感覺這也是寫CODE的定必經之路，最後期許這學期好好的花時間在這門課上，期末的時候可以不再懼怕ML！