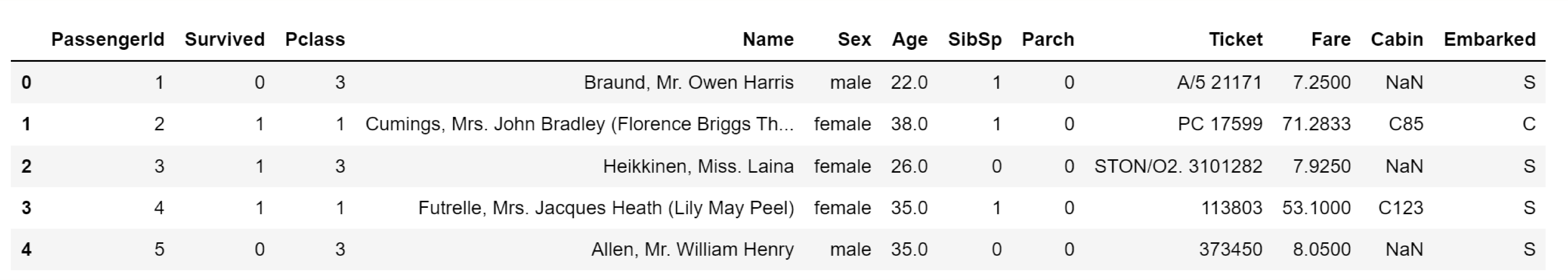
鐵達尼生存預測

407170460 關佳怡

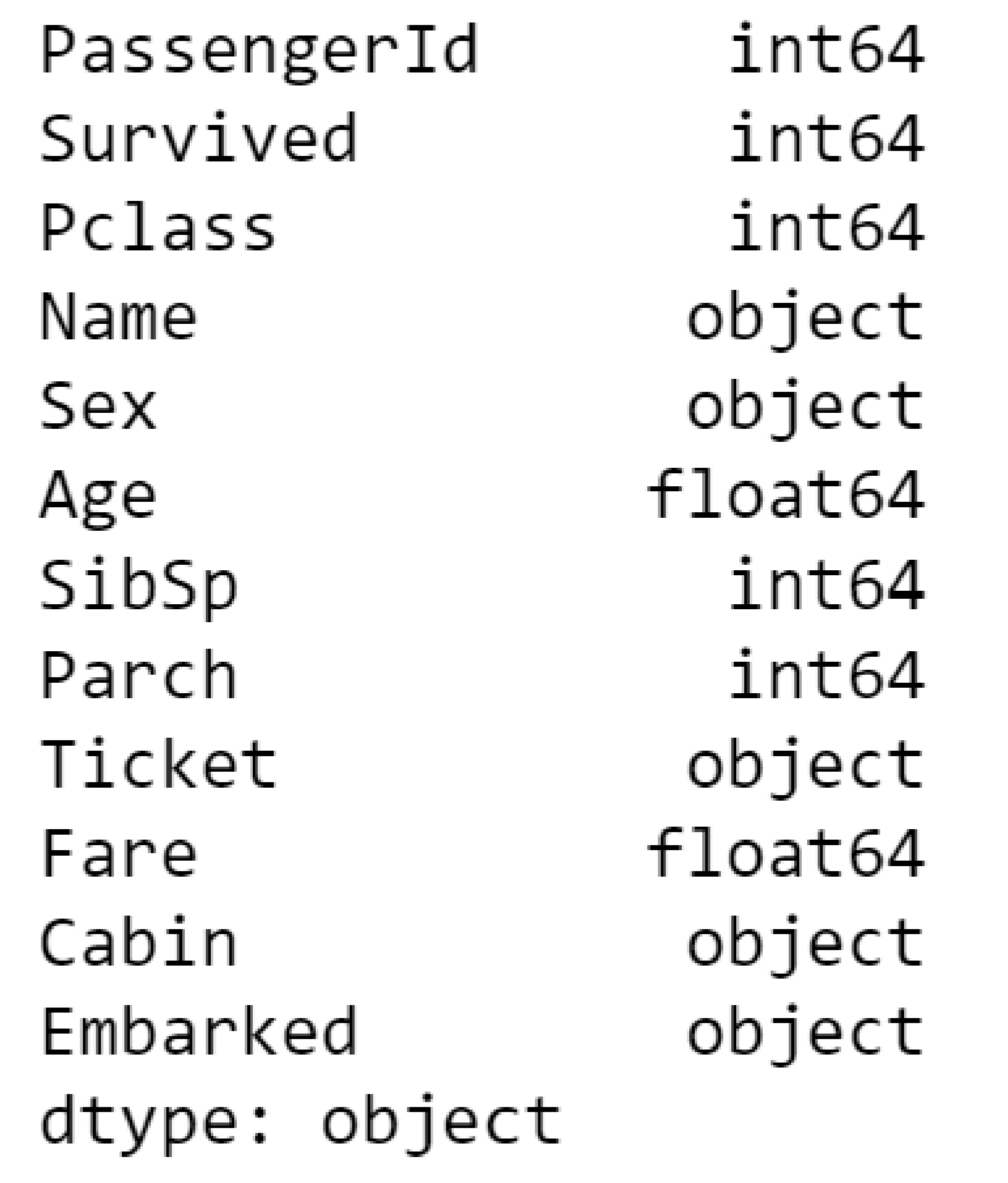
Traindata : 891筆
Testdata : 418筆



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

特徵介紹

旅客編號
倖存下來
票務艙等
名稱
性別
年齡
船上的兄弟姐妹配偶人
船上的父母子女人數
票號
乘客票價
艙
登船港口



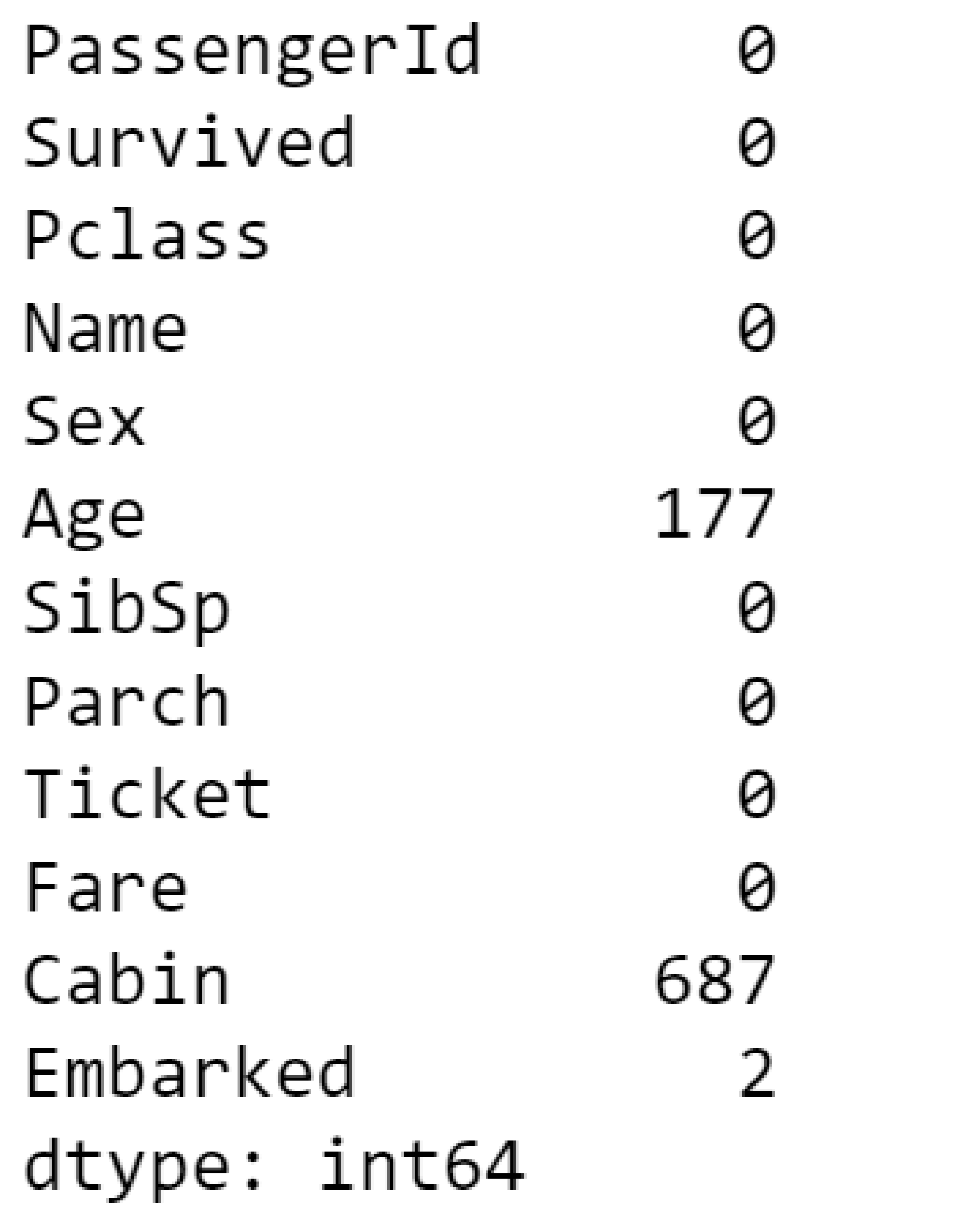
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

查看遺失值

train

有遺失值：

- Age(年齡) -177
- Embarked(登船港口)- 2
- Cabin(船艙) - 687



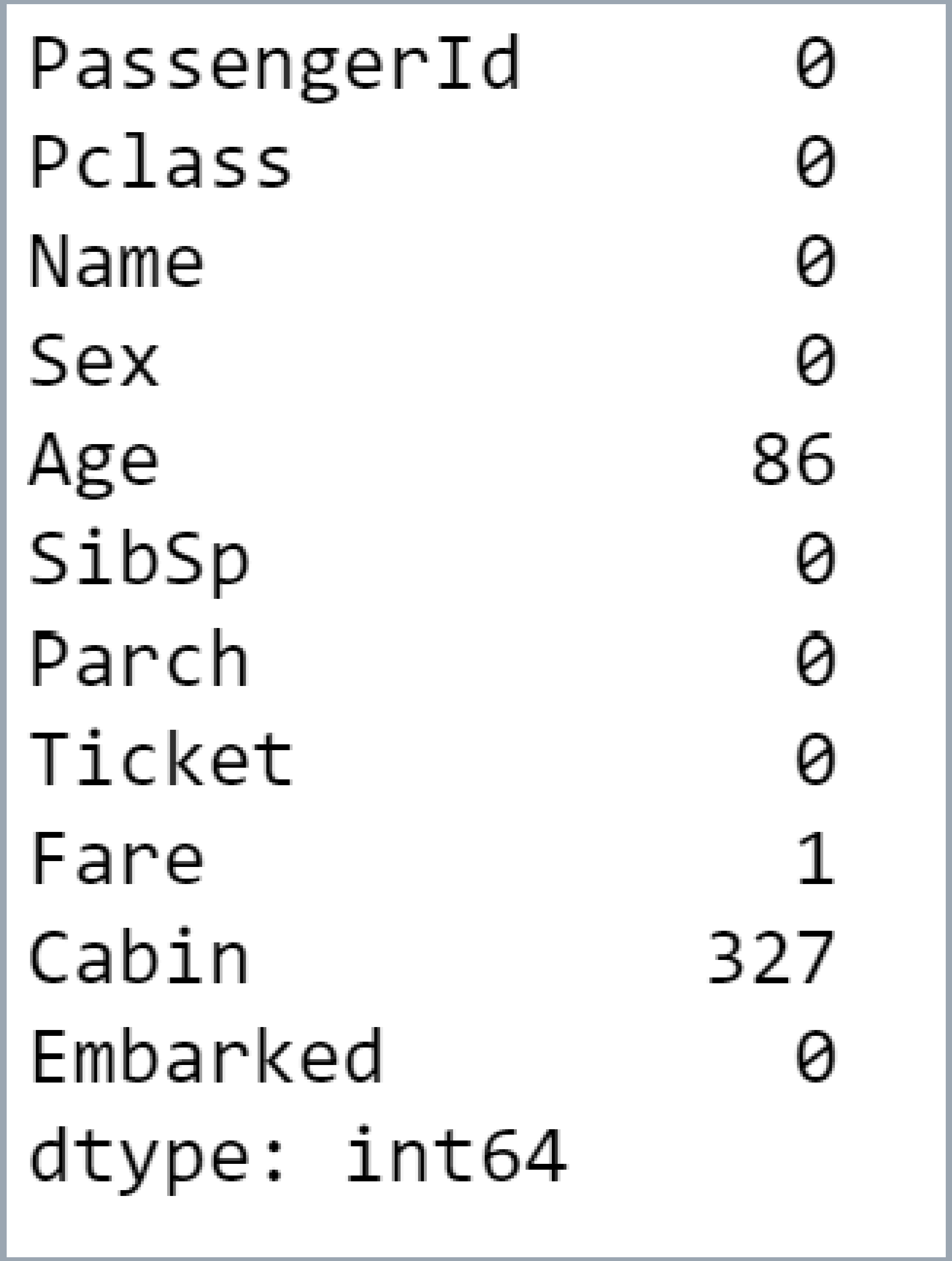
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

查看遺失值

test

有遺失值 :

- Age(年齡) -86
- Fare(登船港口)- 1
- Cabin(船艙) - 327



```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

特徵工程

06.

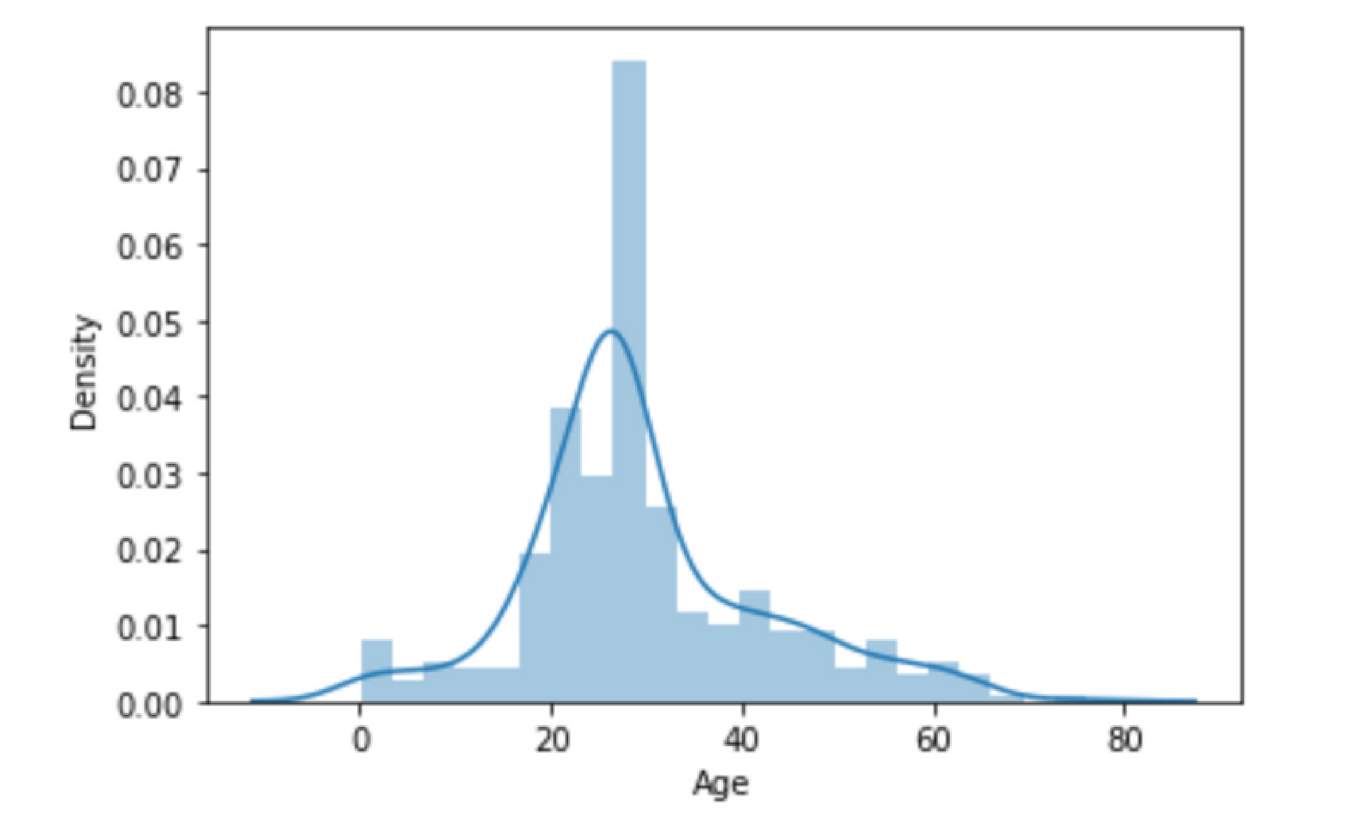
填補遺失值

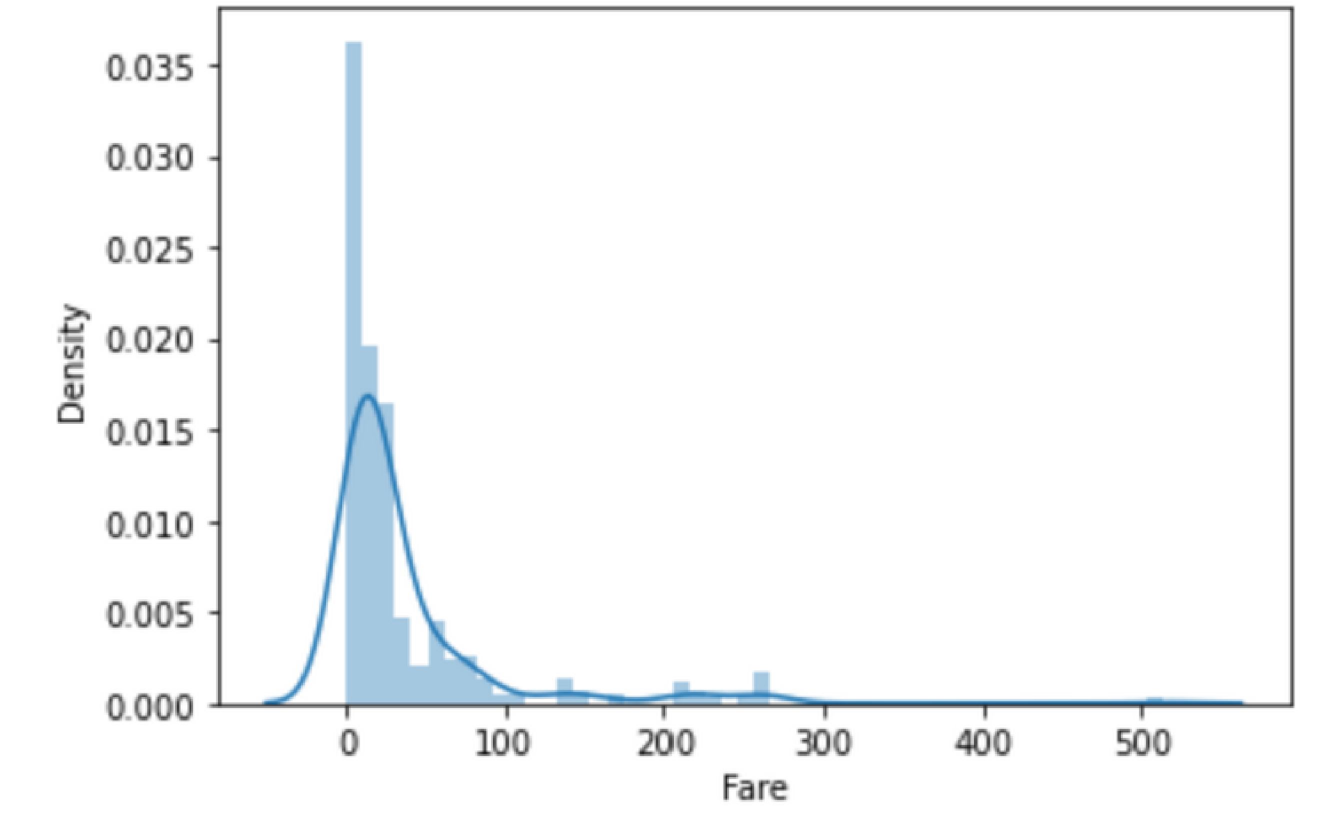
07.

Embarked : 填入出現最多次的港口(S)

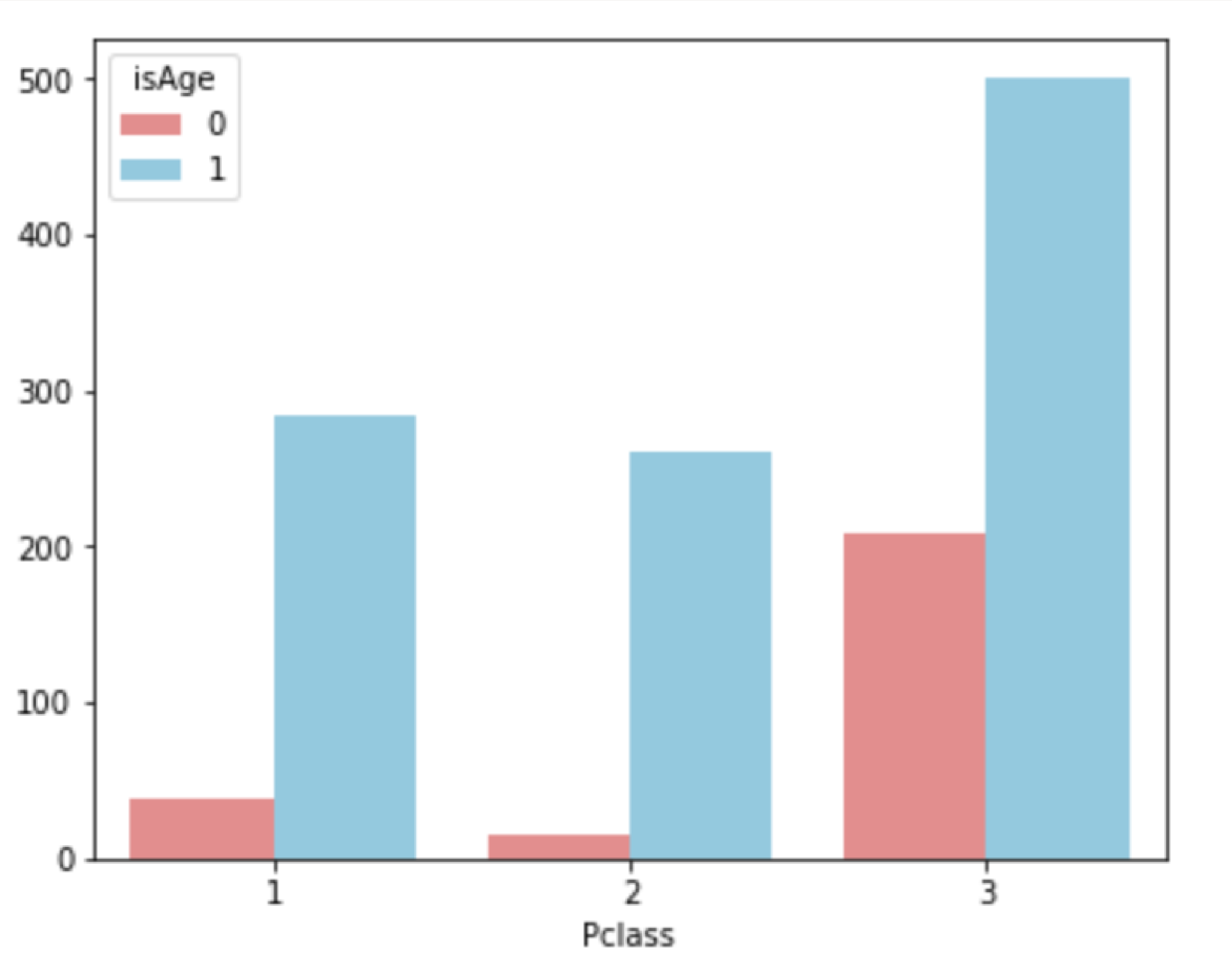
Fare & Age : 填入中位數

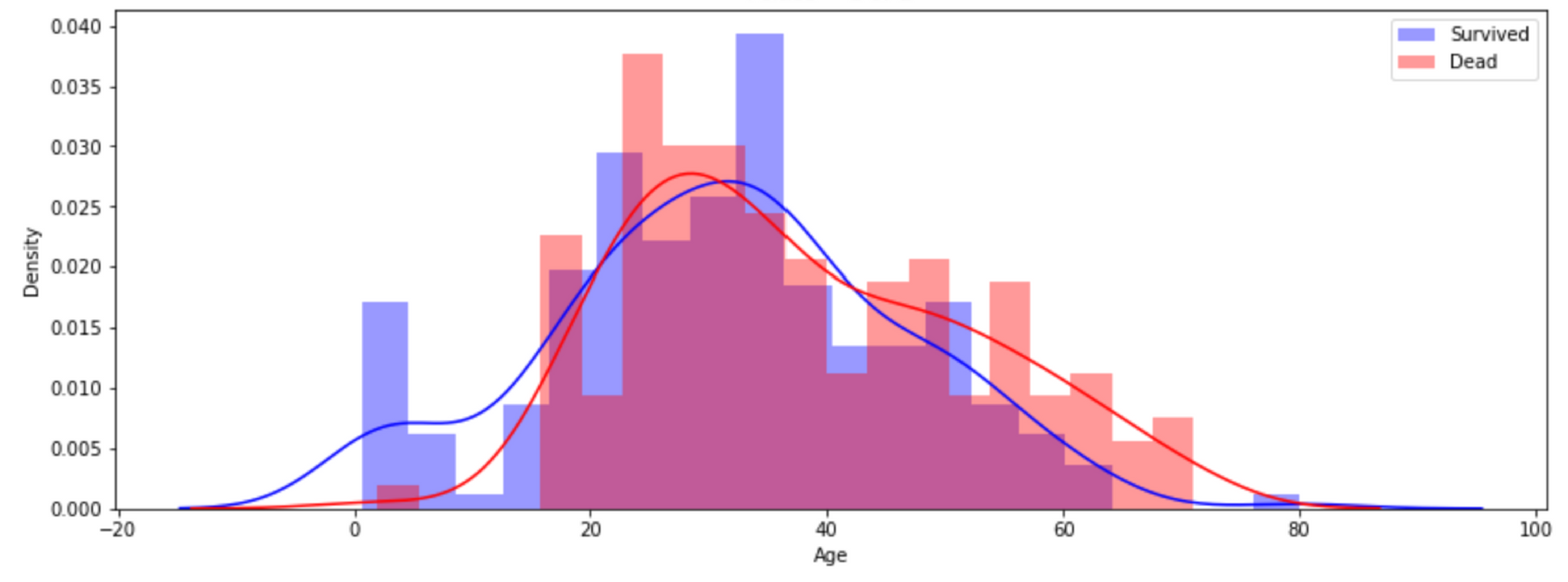
Cabin : 去掉

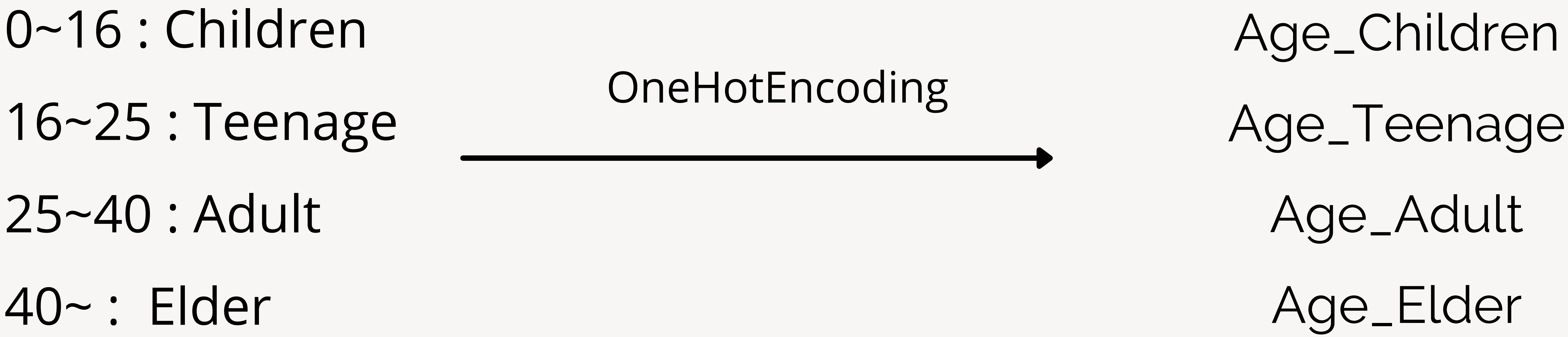




Age_bin


isAge
0
1
500
400
300
200
100
0
1
2
3
Pclass


Pclass = 1 or 2
Survived
Dead
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
−20
0
20
40
60
80
100
Age



0~16 : Children

16~25 : Teenage

25~40 : Adult

40~ : Elder

OneHotEncoding

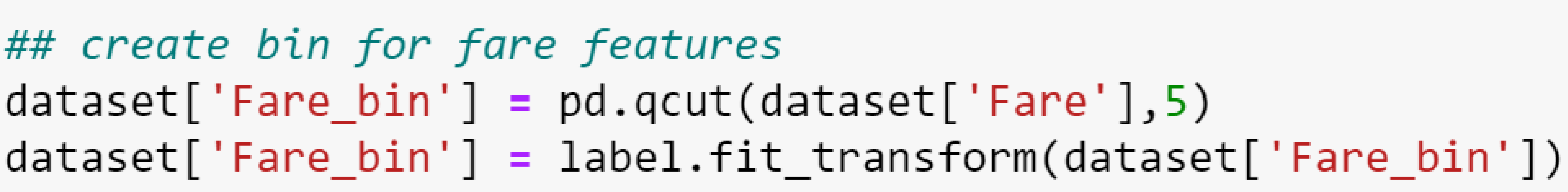
Age_Children

Age_Teenage

Age_Adult

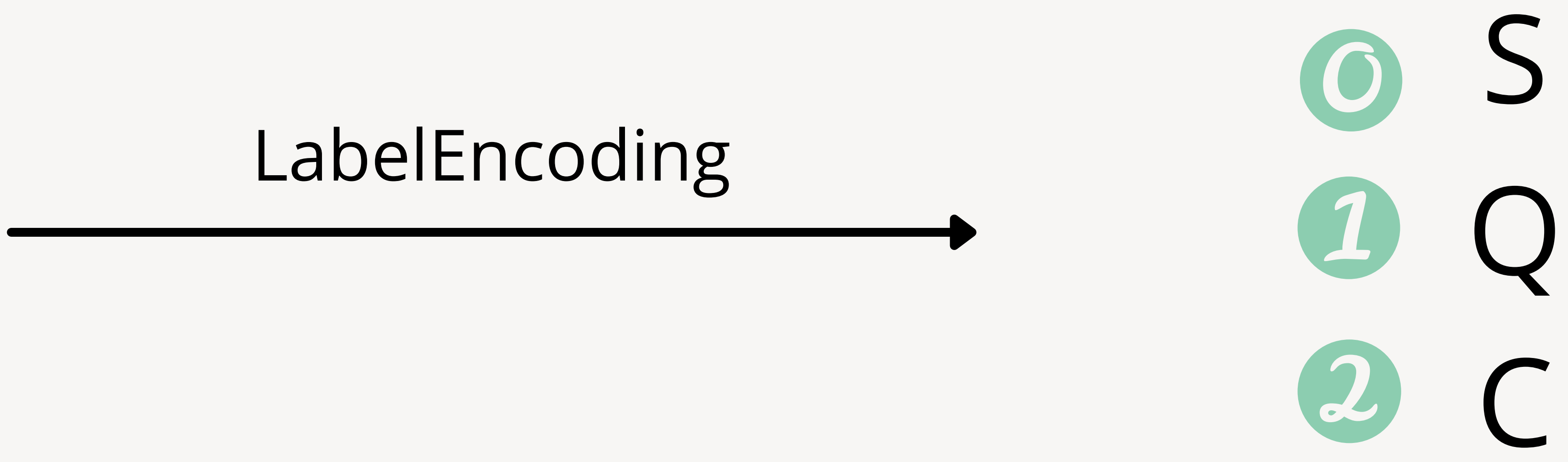
Age_Elder

Fare_bin



```
## create bin for fare features
dataset['Fare_bin'] = pd.qcut(dataset['Fare'],5)
dataset['Fare_bin'] = label.fit_transform(dataset['Fare_bin'])
```

Embarked



LabelEncoding

0 S

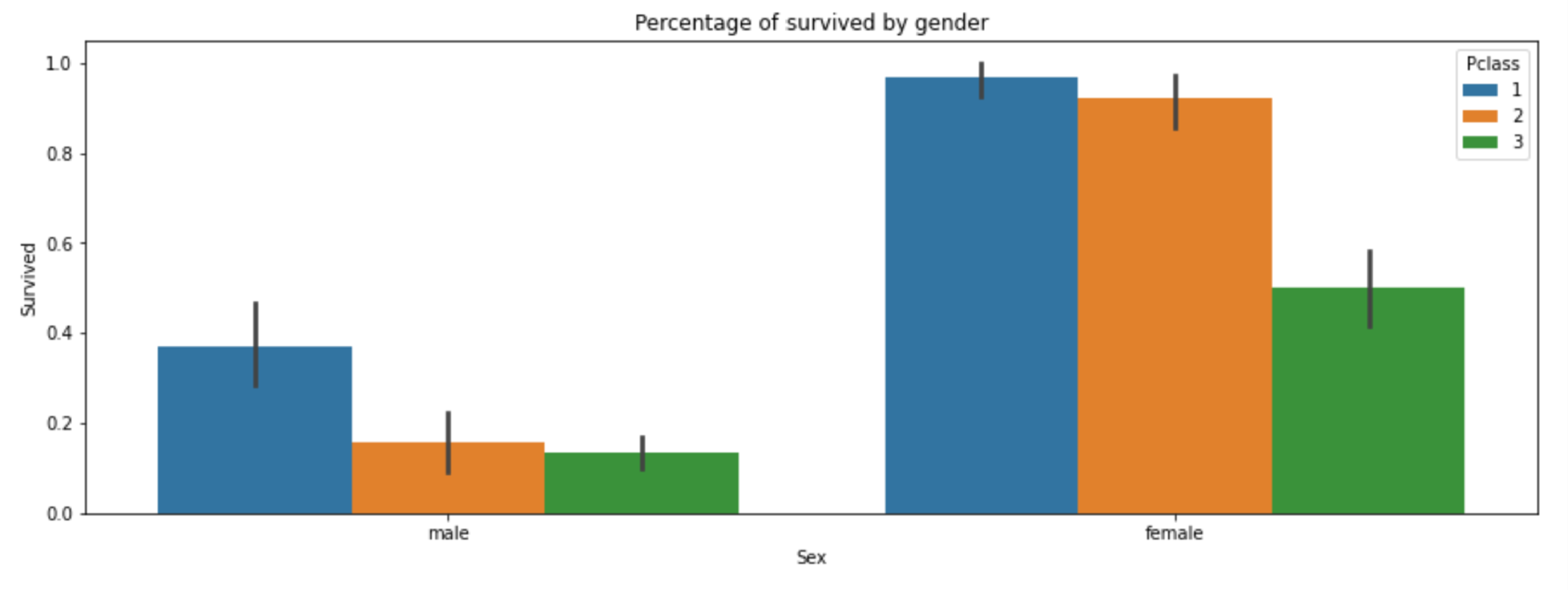
1 Q

2 C

存活率：class1>class2>class3

存活率：女>男

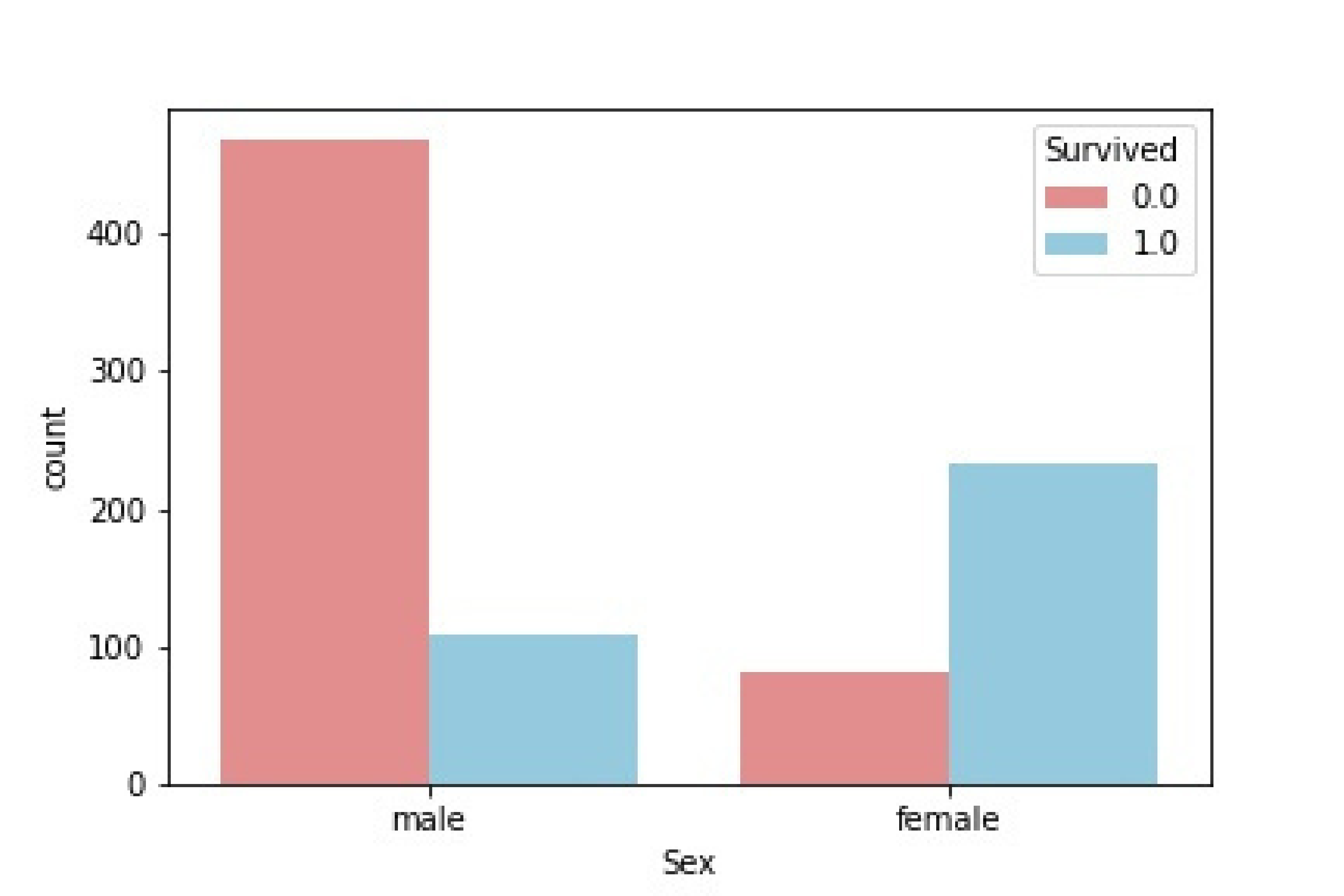
去掉P-class



Sex

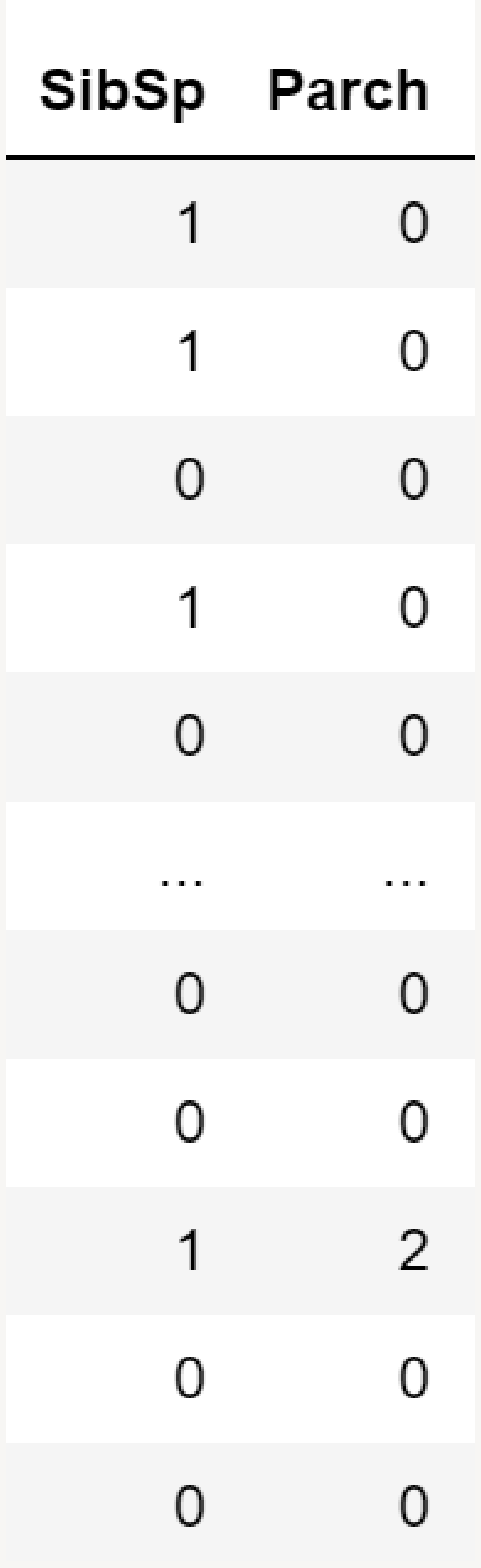
男生僅約18%存活
女生有接近75%存活率

Sex $\xrightarrow{\text{One Hot encoding}}$ 保留female



family_size = SibSp + Parch +1

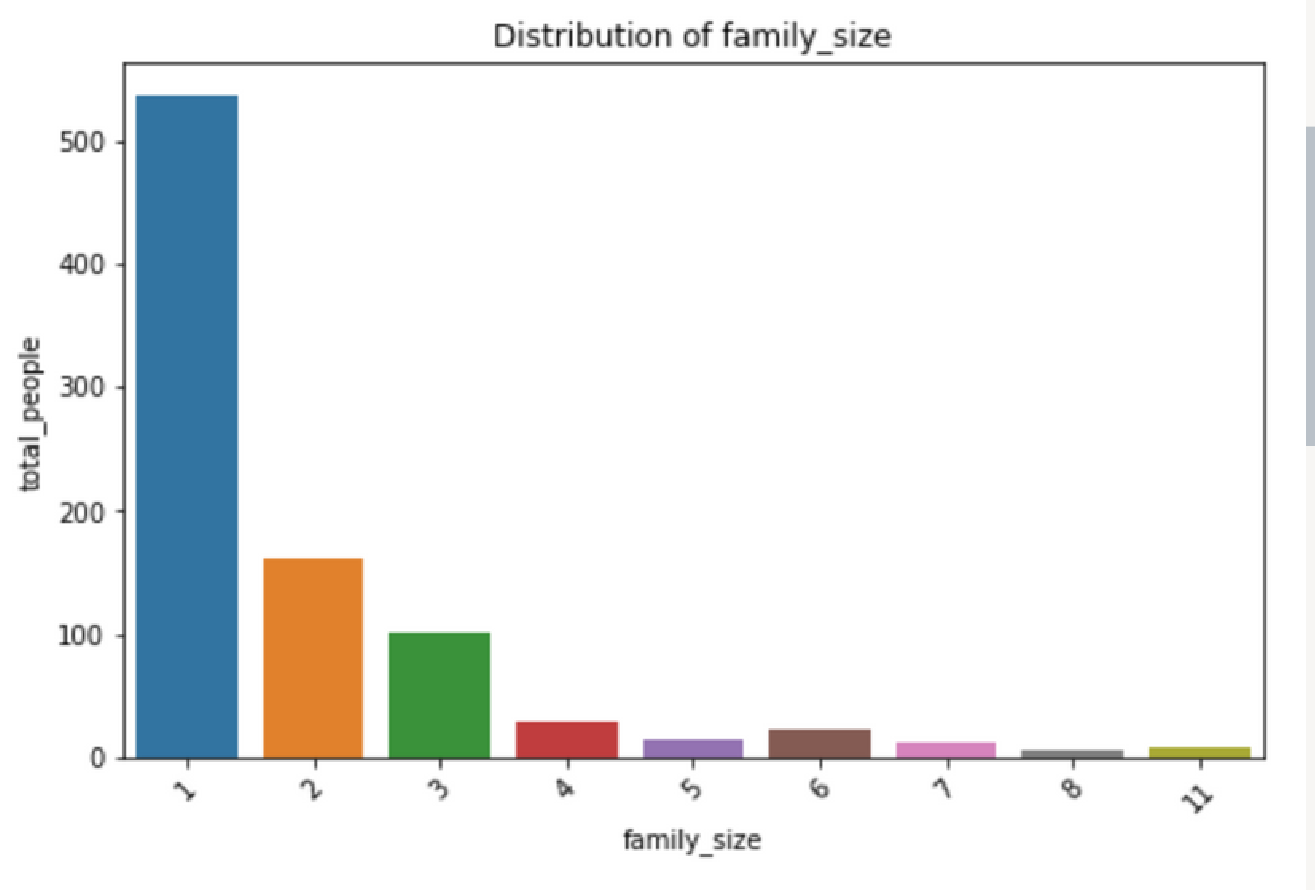
船上兄弟姊妹人數配偶



SibSp	Parch
1	0
1	0
0	0
1	0
0	0
...	...
0	0
0	0
1	2
0	0
0	0

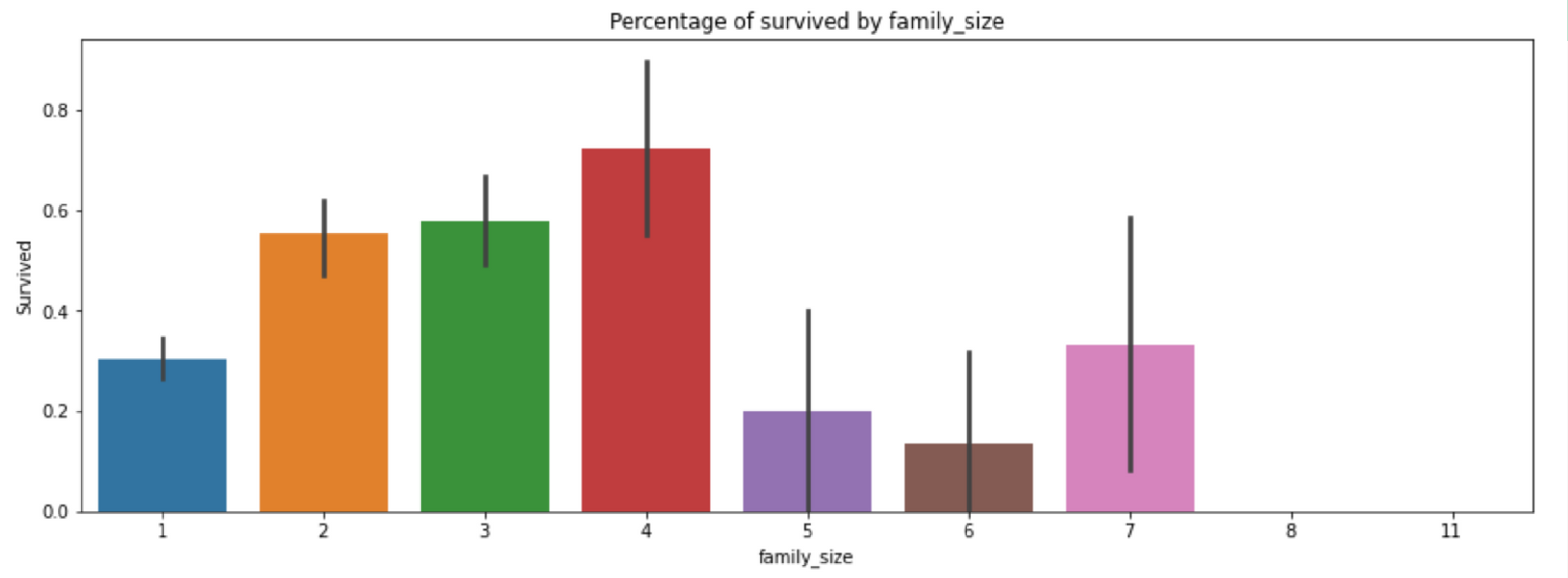
船上的父母子女人數

family_size

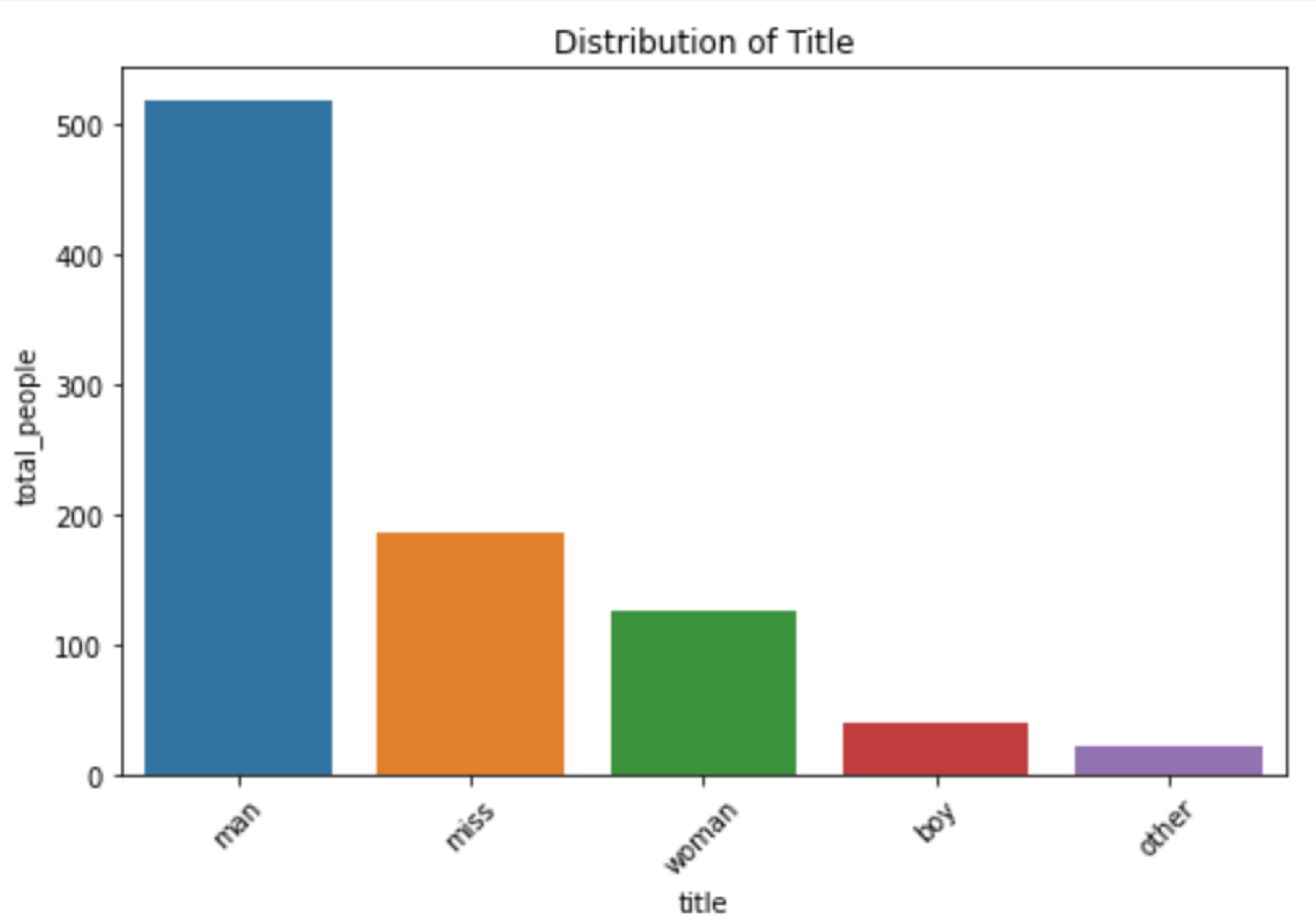


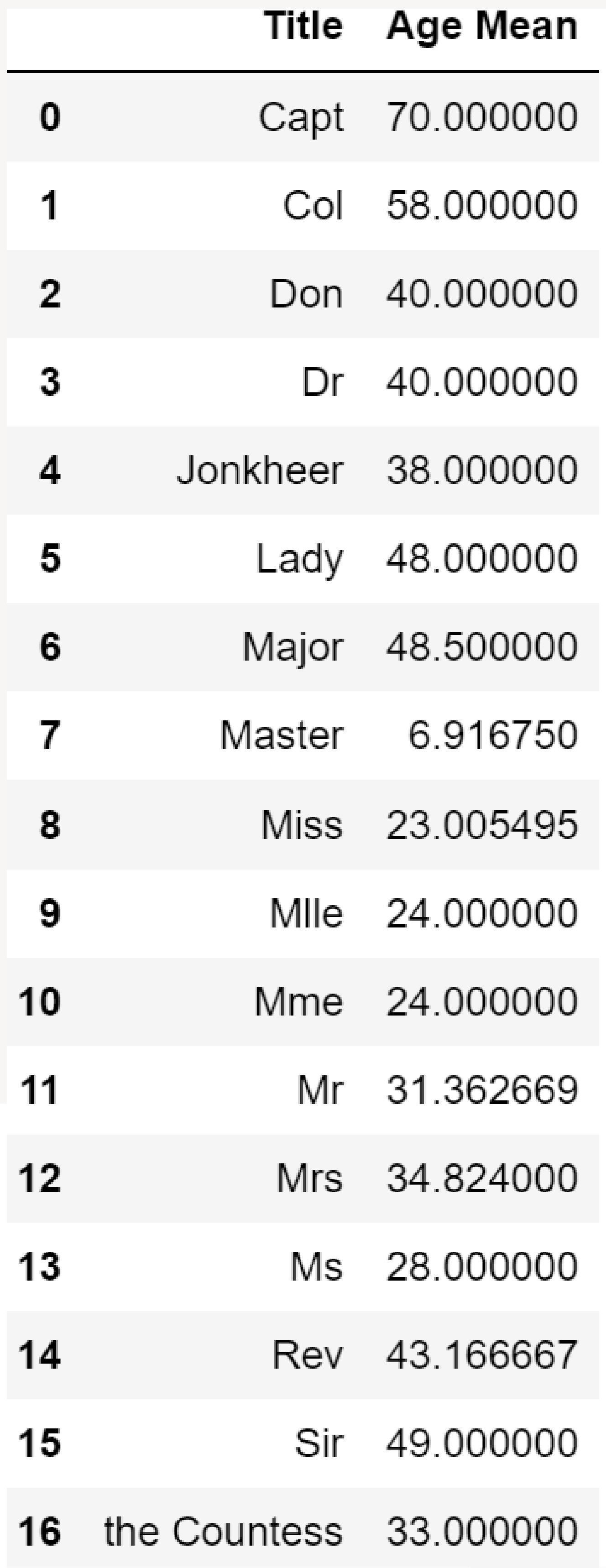
大部分的人都是一個人旅遊

家庭人數約3~4人的乘客存活率高



Boy





	Title	Age Mean
0	Capt	70.000000
1	Col	58.000000
2	Don	40.000000
3	Dr	40.000000
4	Jonkheer	38.000000
5	Lady	48.000000
6	Major	48.500000
7	Master	6.916750
8	Miss	23.005495
9	Mlle	24.000000
10	Mme	24.000000
11	Mr	31.362669
12	Mrs	34.824000
13	Ms	28.000000
14	Rev	43.166667
15	Sir	49.000000
16	the Countess	33.000000

"Mr." -> man

, "Ms.", "Mrs." -> woman

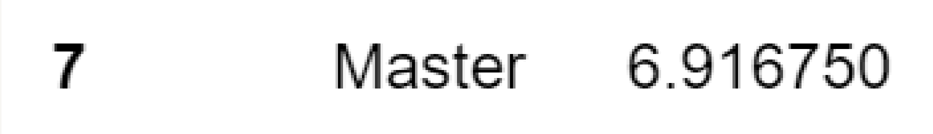
"Miss.", "Mlle.", "Mme." -> miss

"Master." -> boy

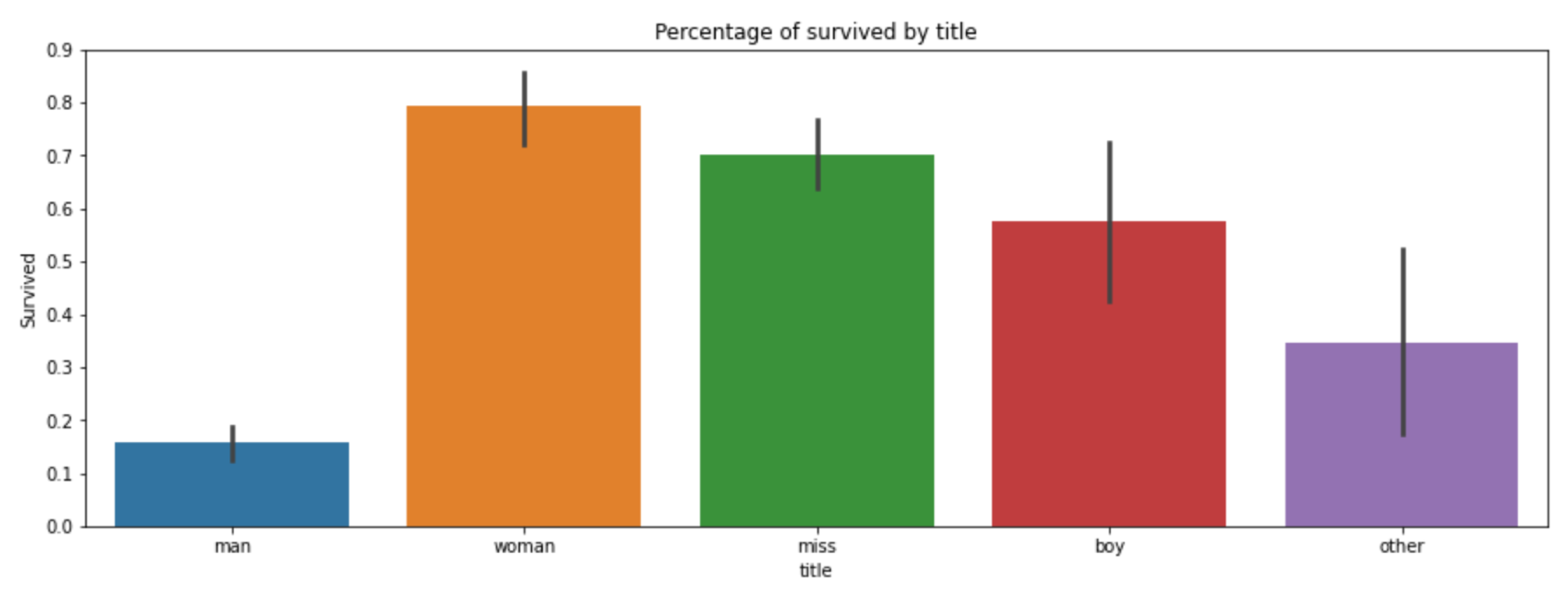
"Capt.", "Col.", "Major.", "Rev.", "Dr." ,"Jonkheer.", "Don.", "Sir.", "Countess.", "Dona.", "Lady." ->other

Master平均年齡約7歲 -> boy(男孩)

Boy存活率約60%



7	Master	6.916750



Woman_child_group

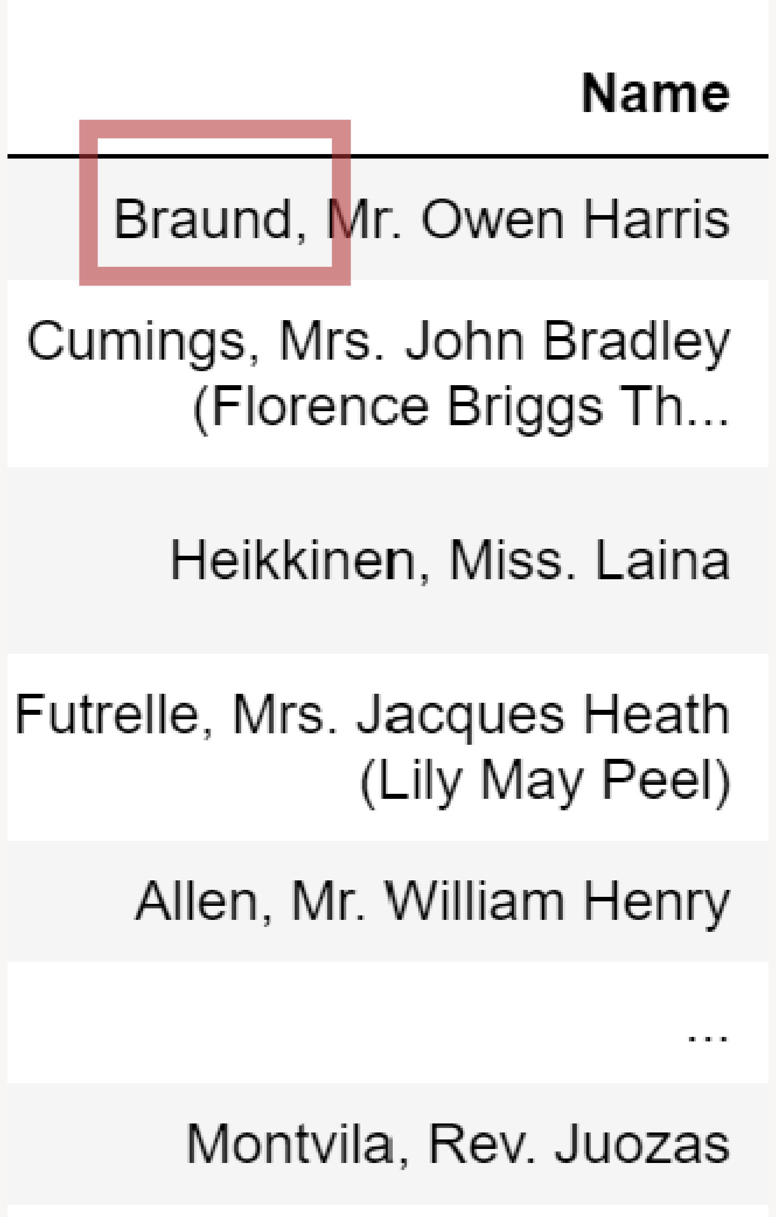
用名字和票號做分類

all_survived : 家庭的每個成員都存活

all_died : 家庭的每個成員都死亡

Woman Child Group By name

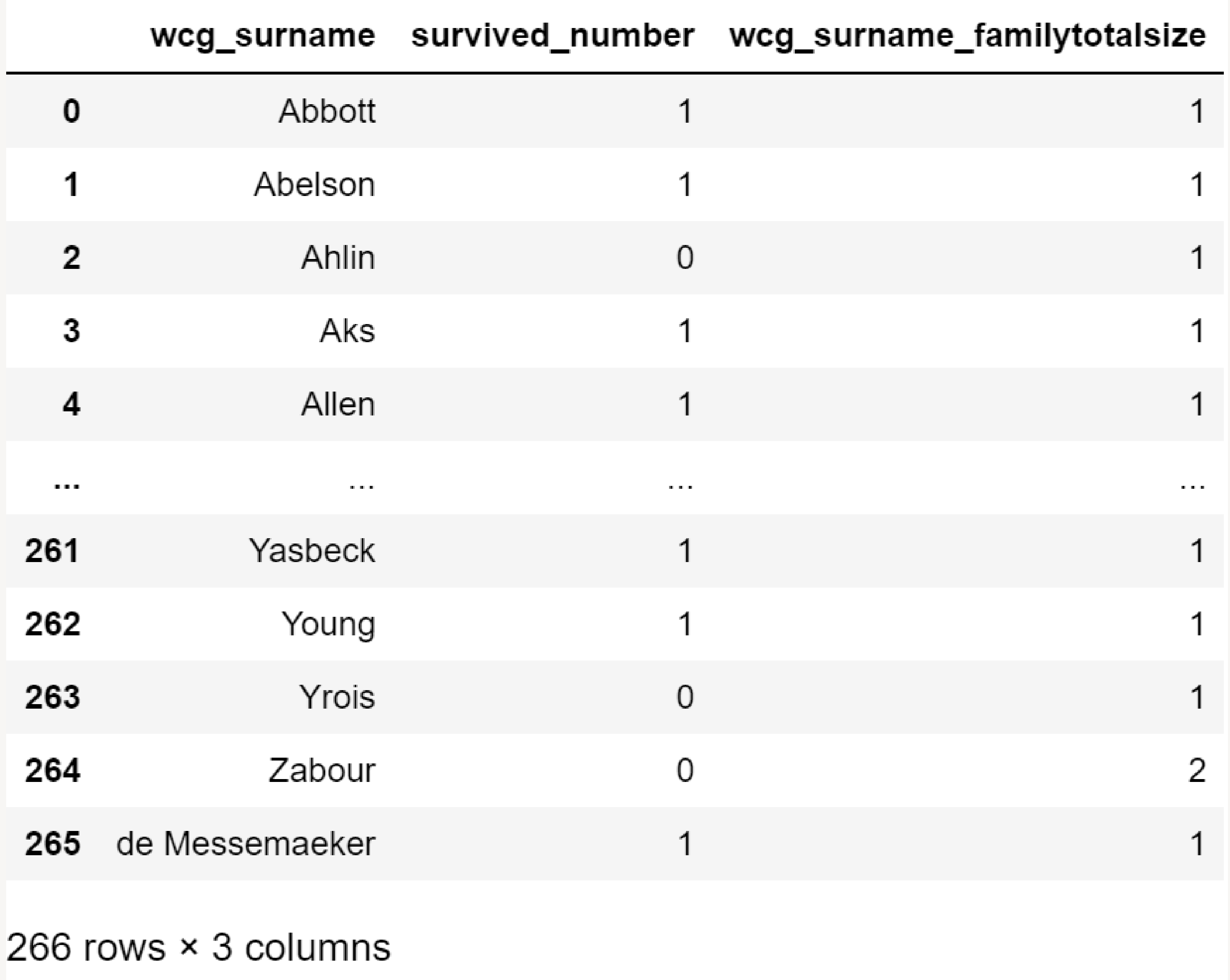
WCG_surname : 擷取名字的第一個字-姓氏



Name
Braund, Mr. Owen Harris
Cumings, Mrs. John Bradley (Florence Briggs Th...
Heikkinen, Miss. Laina
Futrelle, Mrs. Jacques Heath (Lily May Peel)
Allen, Mr. William Henry
...
Montvila, Rev. Juozas

Step1 : 移除不是female也不是Boy的

Step2 : 統計生存人數(survived_number)&
家庭人數(wcg_surname_familytotalsize)

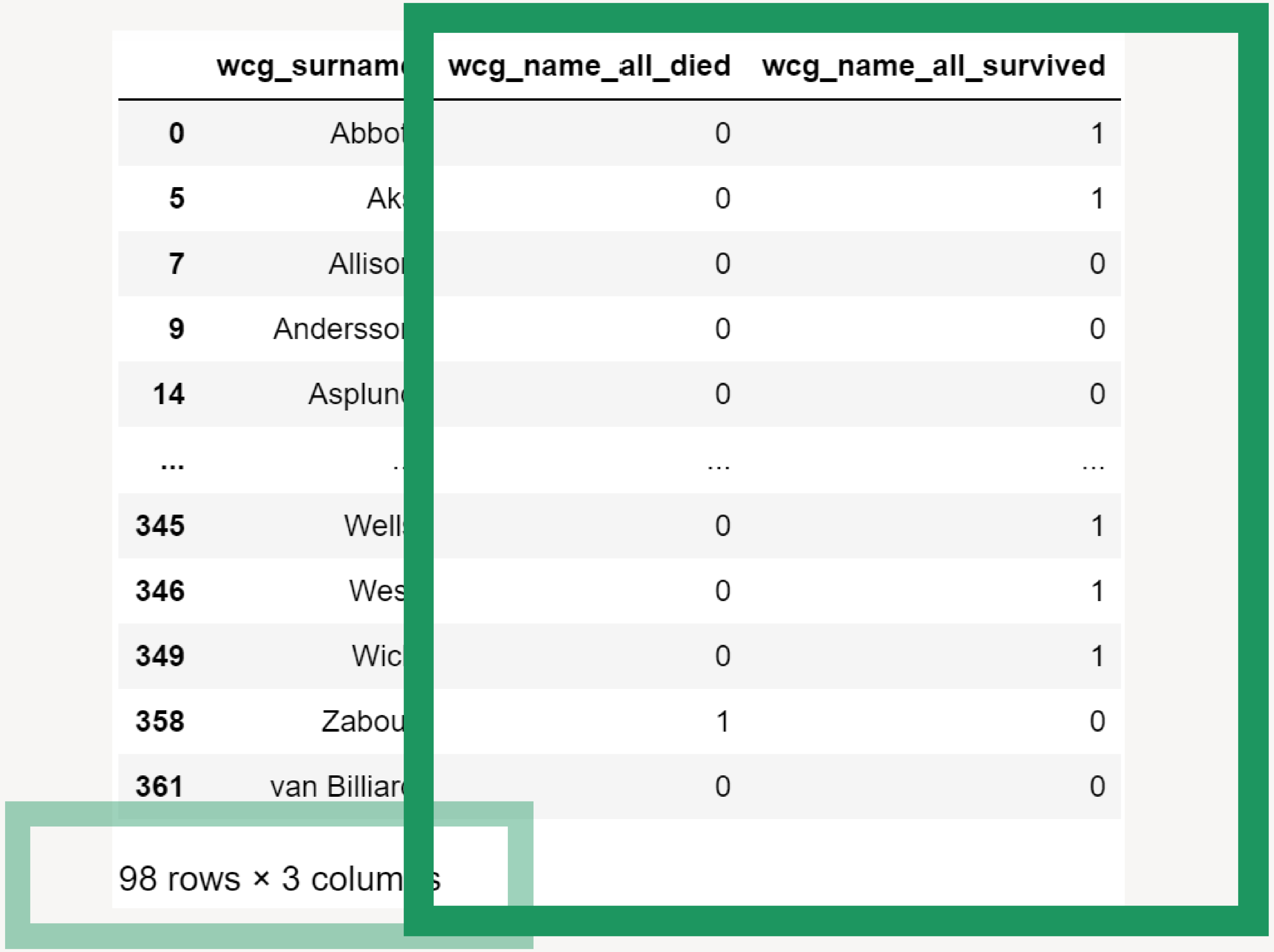


	wcg_surname	survived_number	wcg_surname_familytotalsize
0	Abbott	1	1
1	Abelson	1	1
2	Ahlin	0	1
3	Aks	1	1
4	Allen	1	1
...	...	...	...
261	Yasbeck	1	1
262	Young	1	1
263	Yrois	0	1
264	Zabour	0	2
265	de Messemaeker	1	1

266 rows × 3 columns

Step3：將每個家庭做分類，全部生存一類，全部死亡一類

Step4：保留家庭人數大於1



	wcg_surnam	wcg_name_all_died	wcg_name_all_survived
0	Abbot	0	1
5	Ak	0	1
7	Allisor	0	0
9	Anderssor	0	0
14	Asplun	0	0
...	..	...	...
345	Well	0	1
346	Wes	0	1
349	Wic	0	1
358	Zabou	1	0
361	van Billiar	0	0

98 rows × 3 colum s

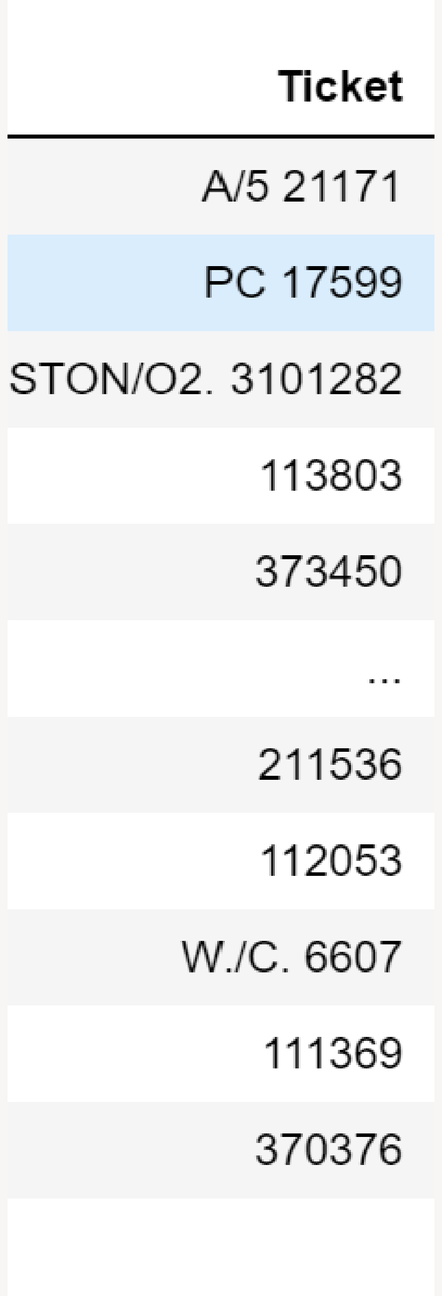
Result : by name

Total groups: 98

All died: 22

All survived: 66

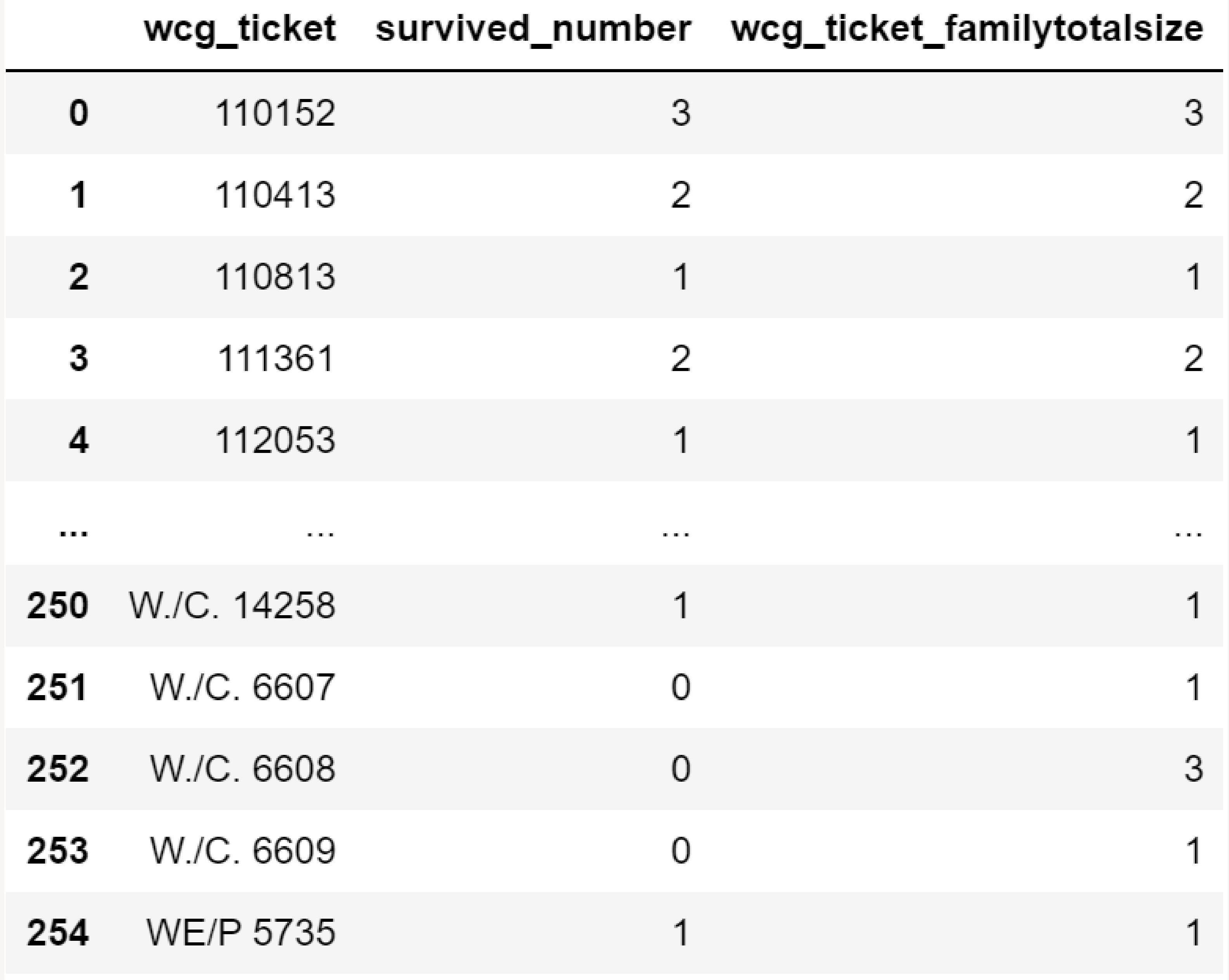
Woman Child Group By Ticket



Ticket
A/5 21171
PC 17599
STON/O2. 3101282
113803
373450
...
211536
112053
W./C. 6607
111369
370376

Step1 : 移除不是female也不是Boy的

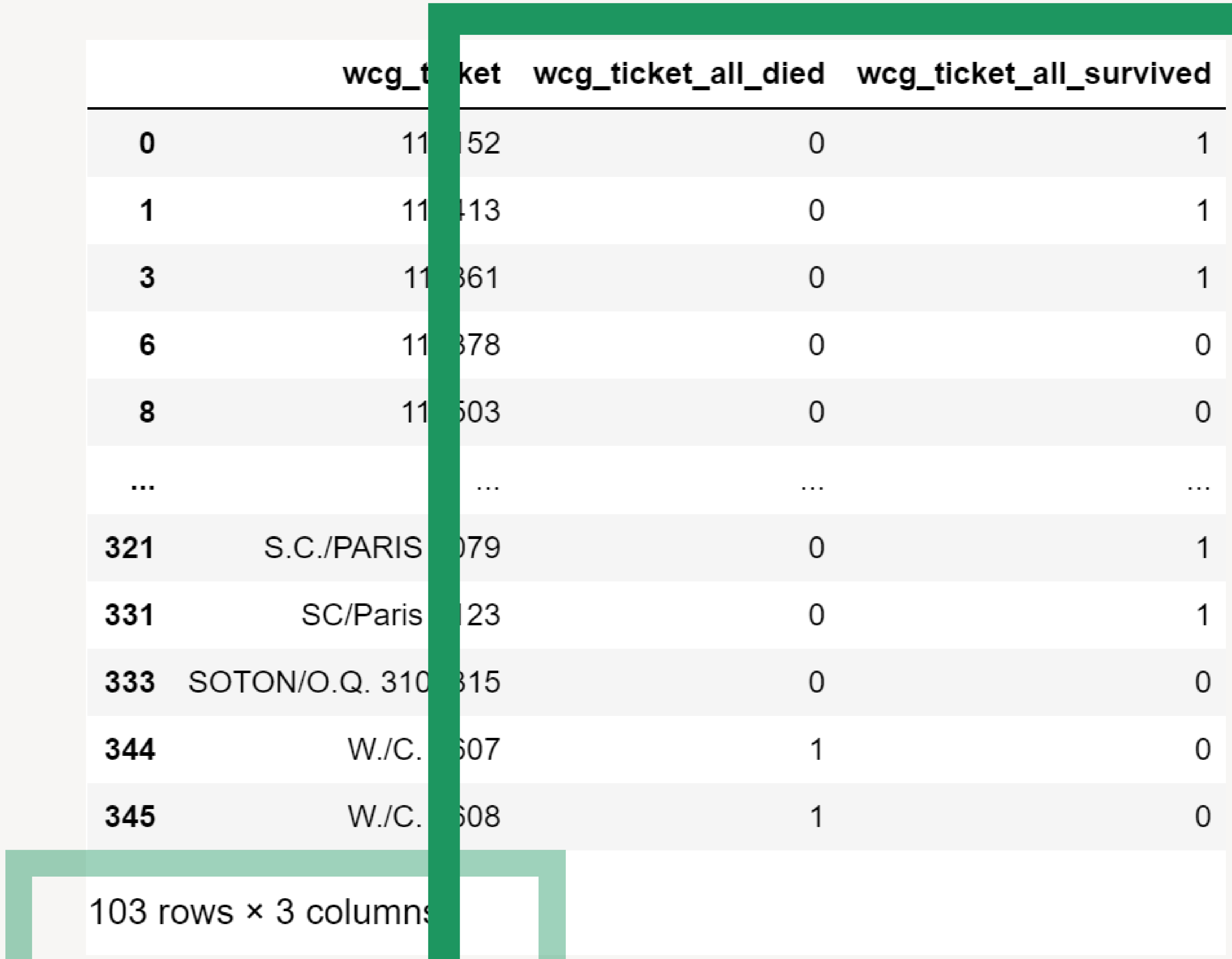
Step2 : 統計生存人數(survived_number)&
家庭人數(wcg_ticket_familytotalsize)



	wcg_ticket	survived_number	wcg_ticket_familytotalsize
0	110152	3	3
1	110413	2	2
2	110813	1	1
3	111361	2	2
4	112053	1	1
...	...	...	...
250	W./C. 14258	1	1
251	W./C. 6607	0	1
252	W./C. 6608	0	3
253	W./C. 6609	0	1
254	WE/P 5735	1	1

Step3：將每個家庭做分類，全部生存一類，全部死亡一類

Step4：保留家庭人數大於1



	wcg_ticket	wcg_ticket_all_died	wcg_ticket_all_survived
0	11[illegible]152	0	1
1	11[illegible]413	0	1
3	11[illegible]361	0	1
6	11[illegible]378	0	0
8	11[illegible]503	0	0
...	...	...	...
321	S.C./PARIS [illegible]079	0	1
331	SC/Paris [illegible]123	0	1
333	SOTON/O.Q. 310[illegible]315	0	0
344	W./C. [illegible]607	1	0
345	W./C. [illegible]608	1	0

103 rows × 3 columns

Result : by ticket

Total groups: 103

All died: 17

All survived: 77

合併namegroup與ticketgroup

all_survived :

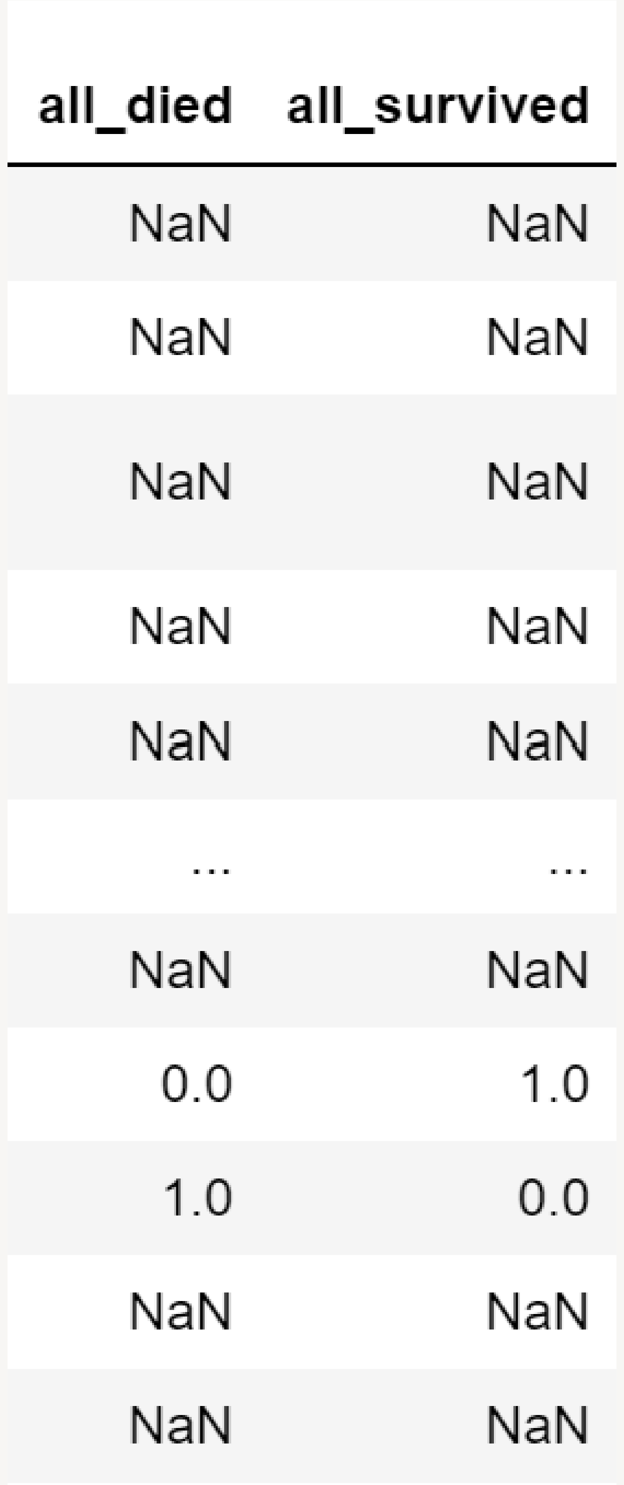
以name & ticket分類的group全部存活 -> 1

all_died :

以name & ticket分類的group全部死亡 -> 1

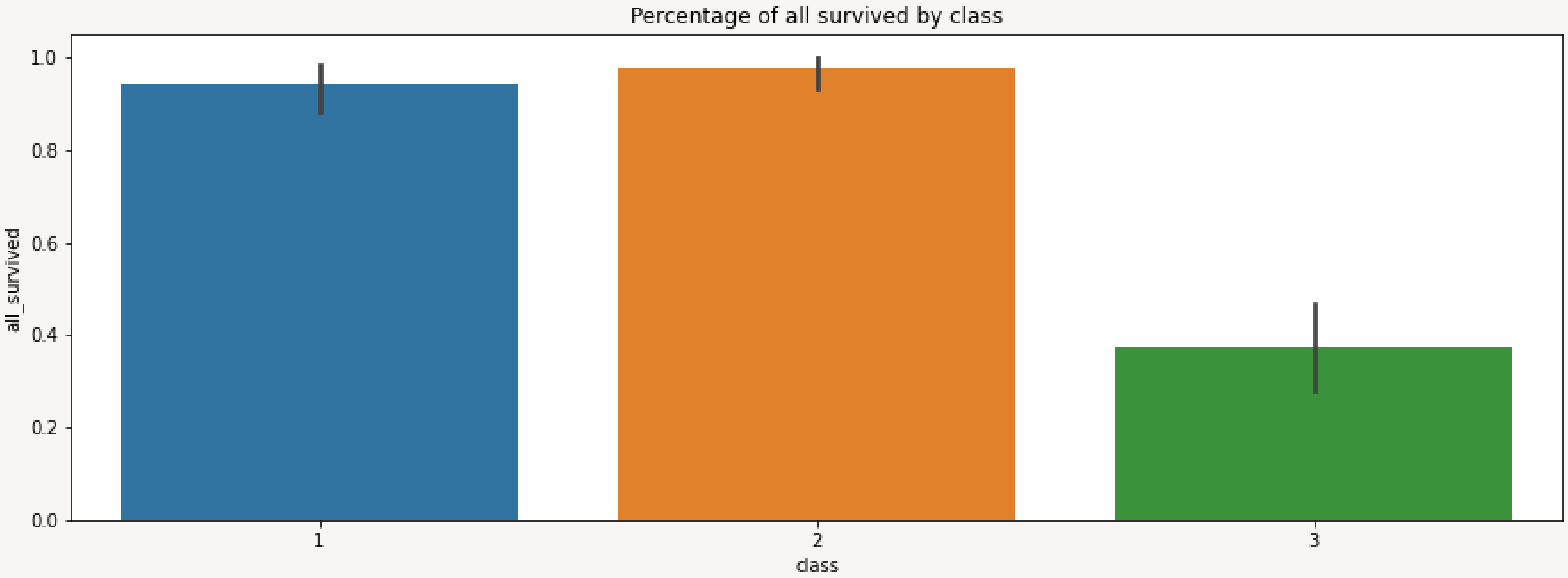
Other

put Nan

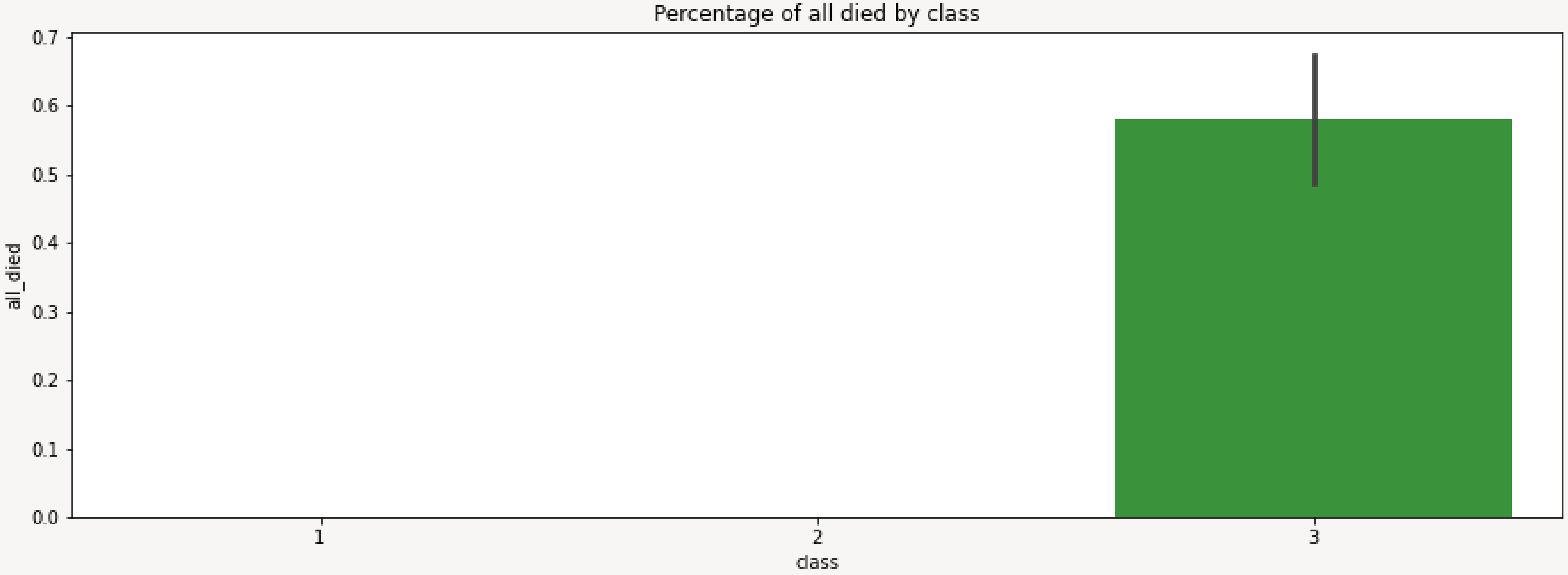


all_died	all_survived
NaN	NaN
NaN	NaN
NaN	NaN
NaN	NaN
NaN	NaN
...	...
NaN	NaN
0.0	1.0
1.0	0.0
NaN	NaN
NaN	NaN

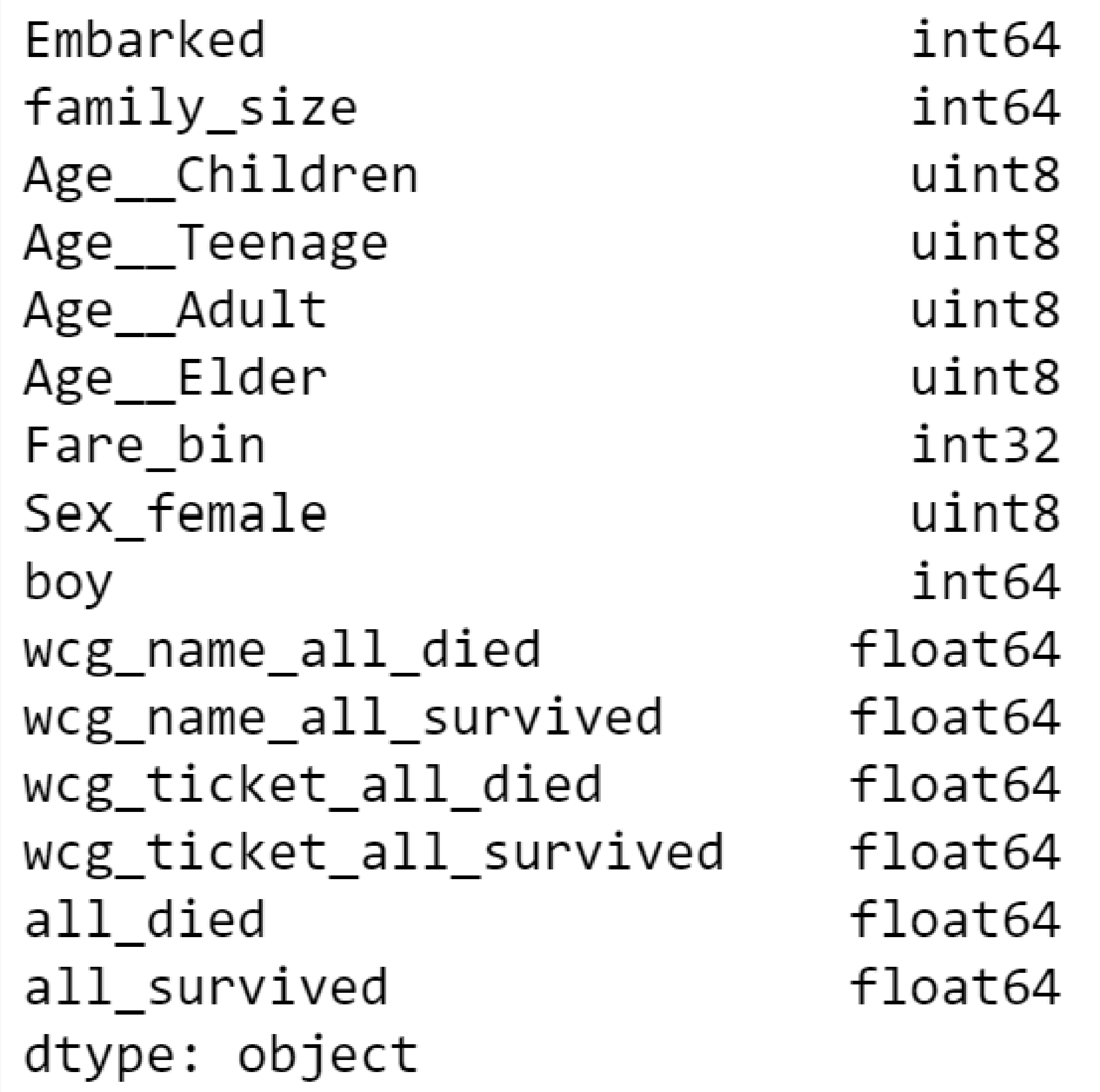
class1&2的家庭大部分都一起存活



class3的家庭大部分一起死亡



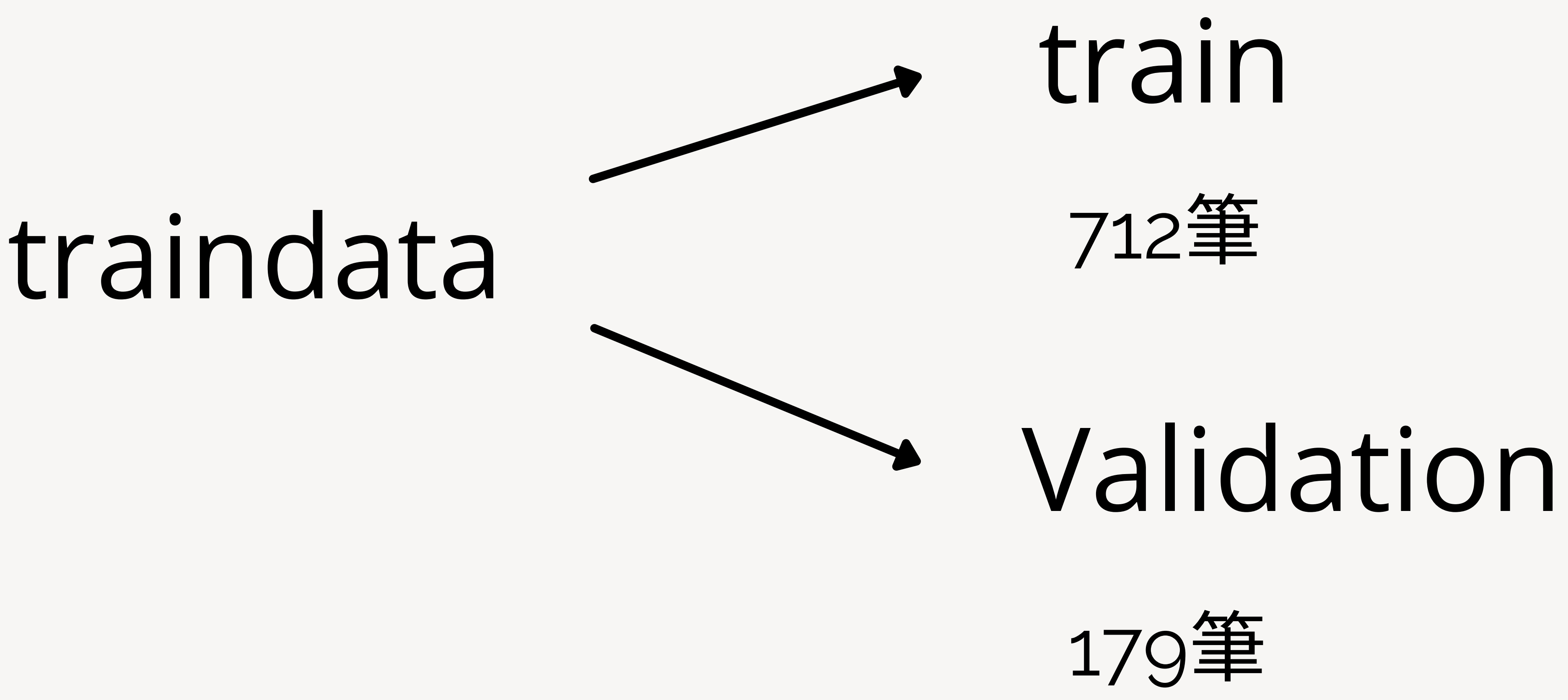
特徵介紹



```
Embarked                      int64
family_size                   int64
Age__Children                 uint8
Age__Teenage                  uint8
Age__Adult                    uint8
Age__Elder                    uint8
Fare_bin                      int32
Sex_female                    uint8
boy                           int64
wcg_name_all_died           float64
wcg_name_all_survived       float64
wcg_ticket_all_died         float64
wcg_ticket_all_survived     float64
all_died                    float64
all_survived                float64
dtype: object
```

- Embarked : 登船港口
- family_size: 家庭人數
- Age_Cildren: 年齡(0~16)
- Age_Teenage: 年齡16~25)
- Age_Adult: 年齡(25~40)
- Age_Elder: 年齡(40~)
- Fare_bin: 票價(5等分)
- Sex_female: 女生
- boy: 小男孩
- wcg_name_all_died:以name分類全部死亡
- wcg_name_all_survived: 以name分類全部存活
- wcg_ticket_all_died: 以ticket分類全部死亡
- wcg_ticket_all_survived: 以ticket分類全部存活
- all_died: 以name&ticket分類全部死亡
- all_survived: 以name&ticket分類全部存活

拆分數據集



traindata

train

712筆

Validation

179筆

標準化 : 全部特徵

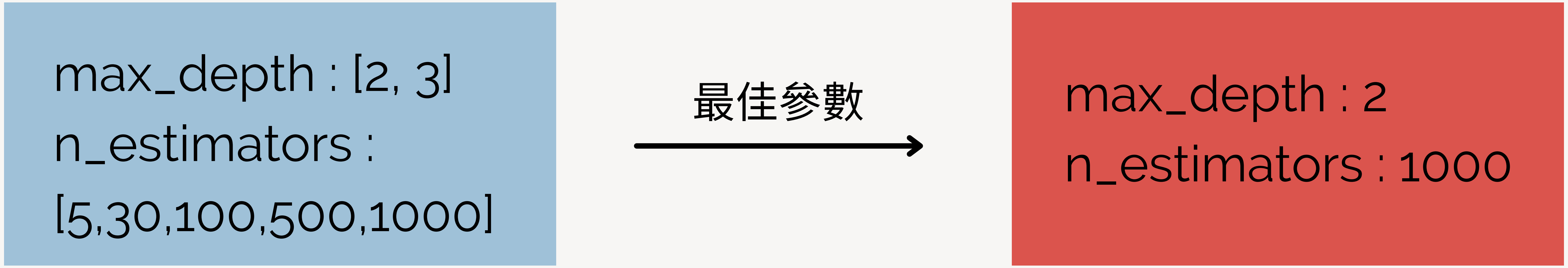
Xgboost
(極限梯度提升演算法)

GridSearchCV

在所有候選的參數選擇中，通過循環遍歷，
嘗試每一種可能性，表現最好的參數就是最終的結果。

優點 : 可以在指定的參數範圍內找到精度最高的參數

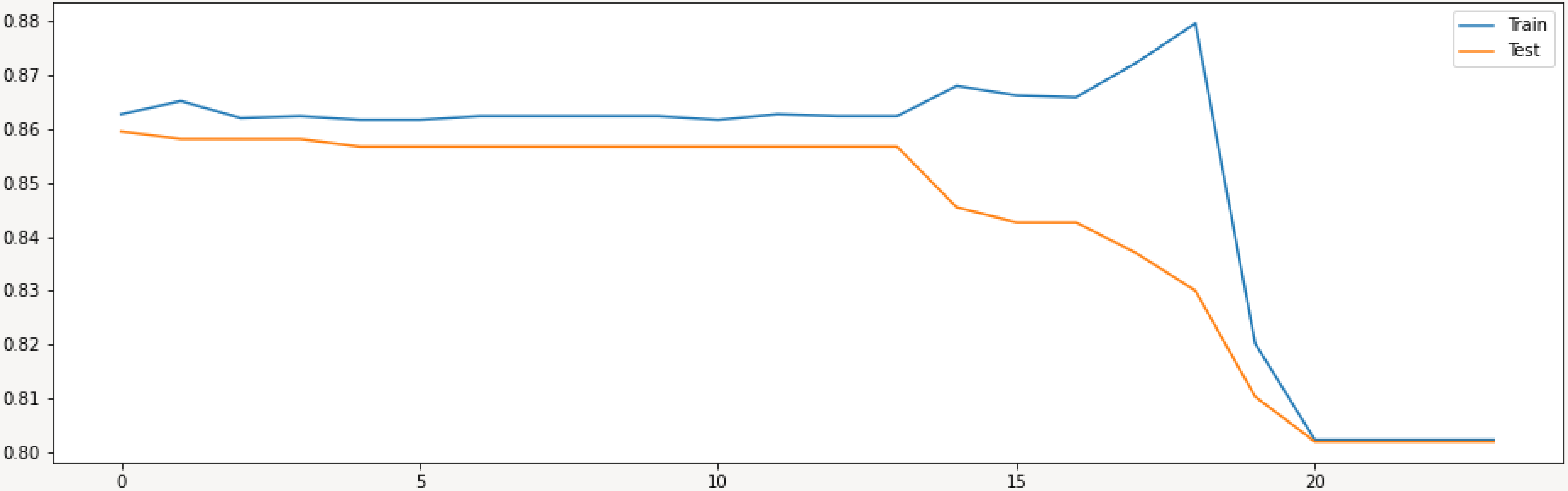
缺點 : 在面對大數據集和多參數的情況下非常耗時



max_depth : [2, 3]
n_estimators :
[5,30,100,500,1000]

最佳參數

max_depth : 2
n_estimators : 1000



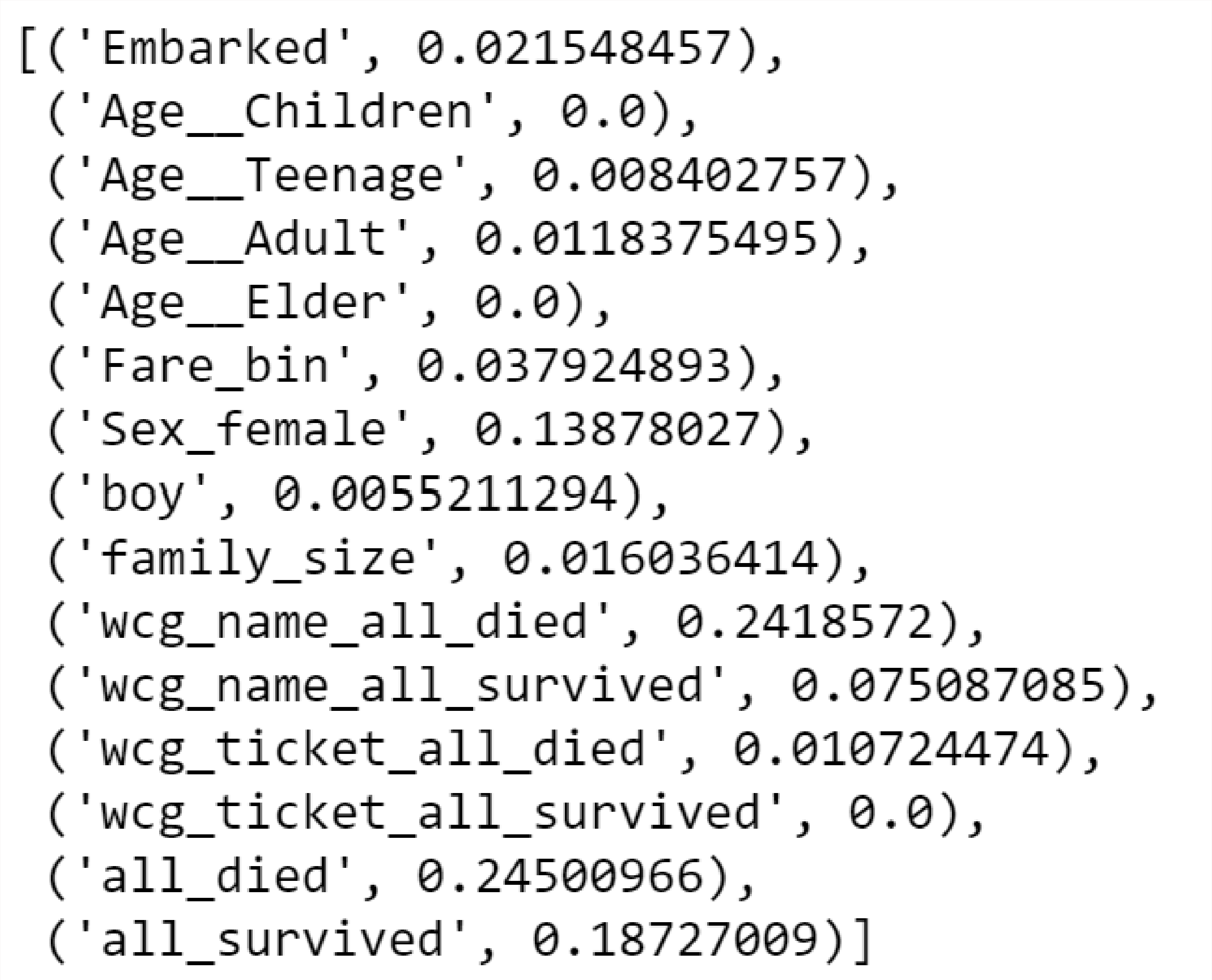
建立模型

全部特徵

使用 KFold 得到的準確率 :

trainSet : 0.85

ValidationSet :   0.84

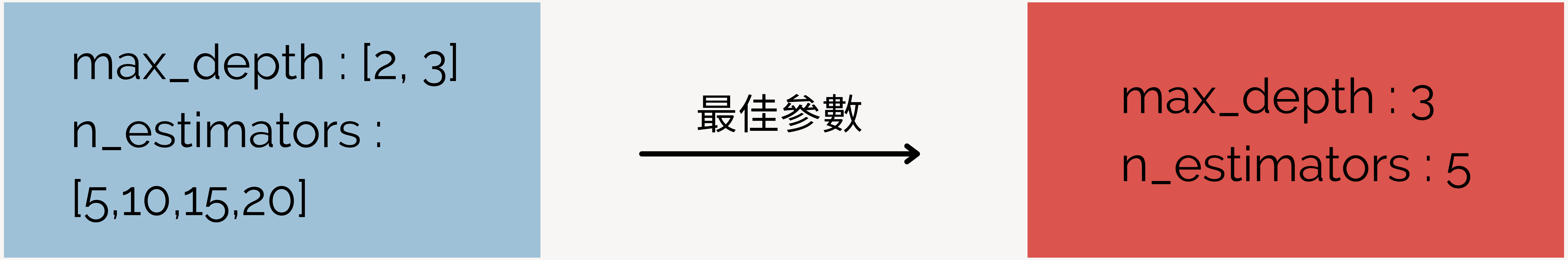


```
[('Embarked', 0.021548457),
 ('Age__Children', 0.0),
 ('Age__Teenage', 0.008402757),
 ('Age__Adult', 0.0118375495),
 ('Age__Elder', 0.0),
 ('Fare_bin', 0.037924893),
 ('Sex_female', 0.13878027),
 ('boy', 0.0055211294),
 ('family_size', 0.016036414),
 ('wcg_name_all_died', 0.2418572),
 ('wcg_name_all_survived', 0.075087085),
 ('wcg_ticket_all_died', 0.010724474),
 ('wcg_ticket_all_survived', 0.0),
 ('all_died', 0.24500966),
 ('all_survived', 0.18727009)]
```

選出特徵重要性較高的特徵

- Sex_female -（女生）
- all_died -（以name&ticket分類全部死亡）
- all_survived -（以name&ticket分類全部存活）

GridSearch



max_depth : [2, 3]
n_estimators : [5,10,15,20]

最佳參數

max_depth : 3
n_estimators : 5

建模

- Sex_female - （女生）
- all_died - （以name&ticket分類全部死亡）
- all_survived - （以name&ticket分類全部存活）

max_depth : 3
n_estimators : 5

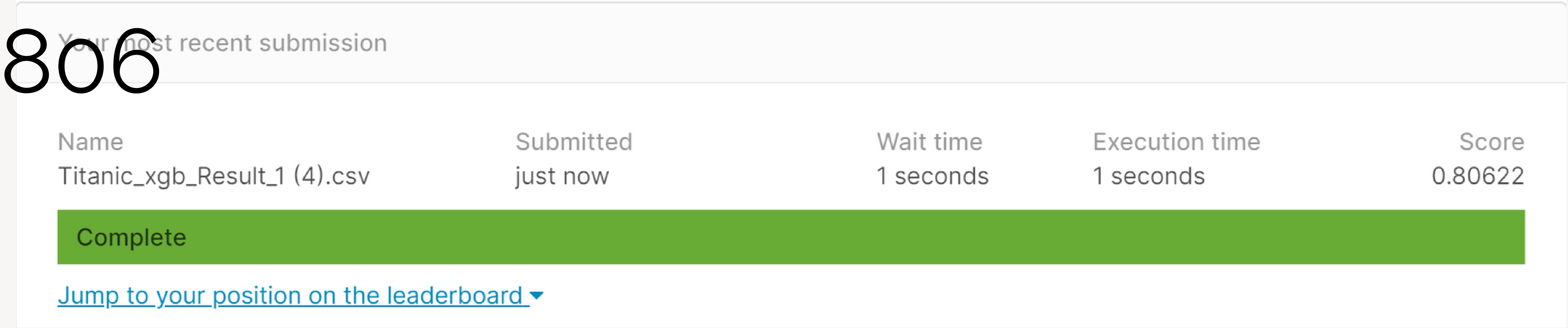
使用 KFold 得到的準確率 :

trainset : 0.85

ValidationSet : 0.82

Kaggle :

0.806



Your most recent submission

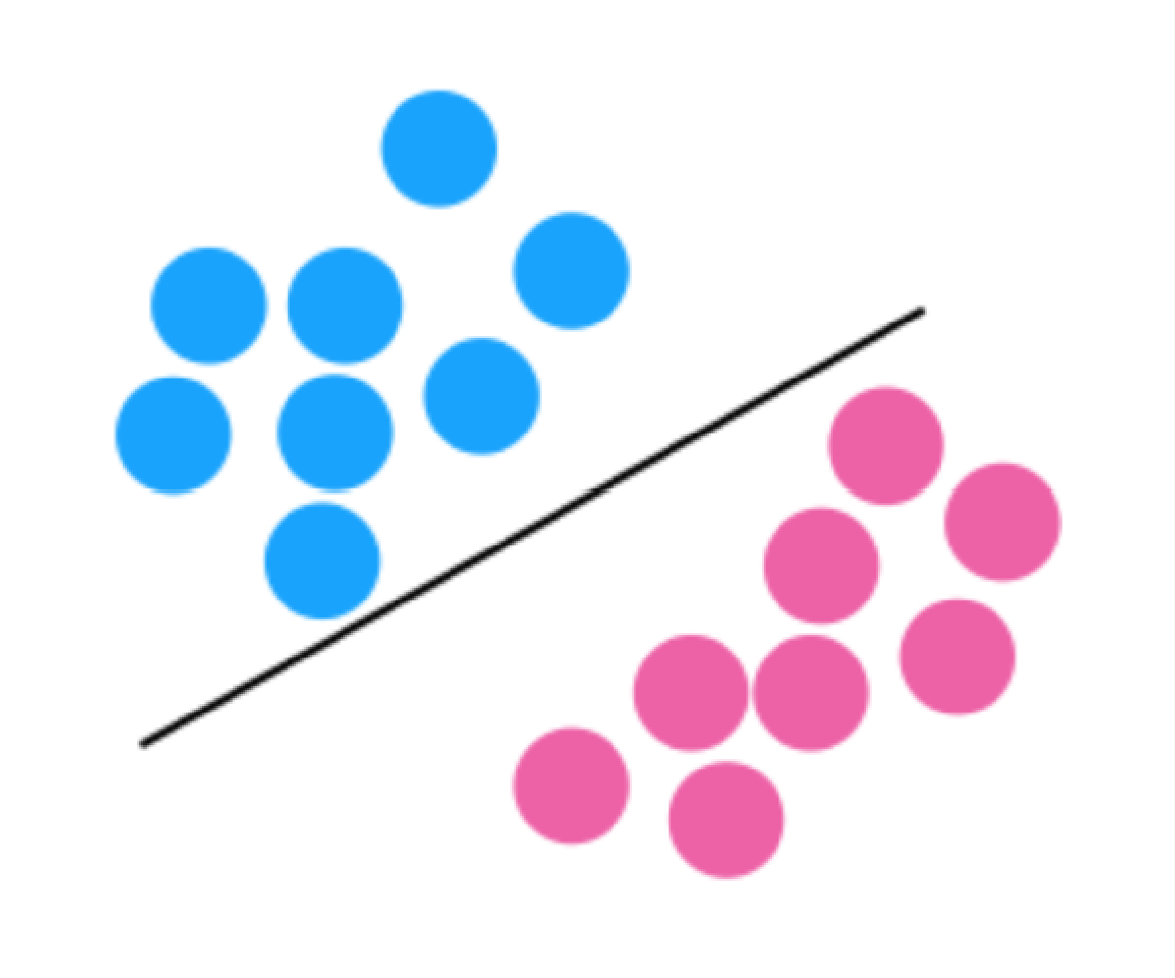
Name	Submitted	Wait time	Execution time	Score
Titanic_xgb_Result_1 (4).csv	just now	1 seconds	1 seconds	0.80622

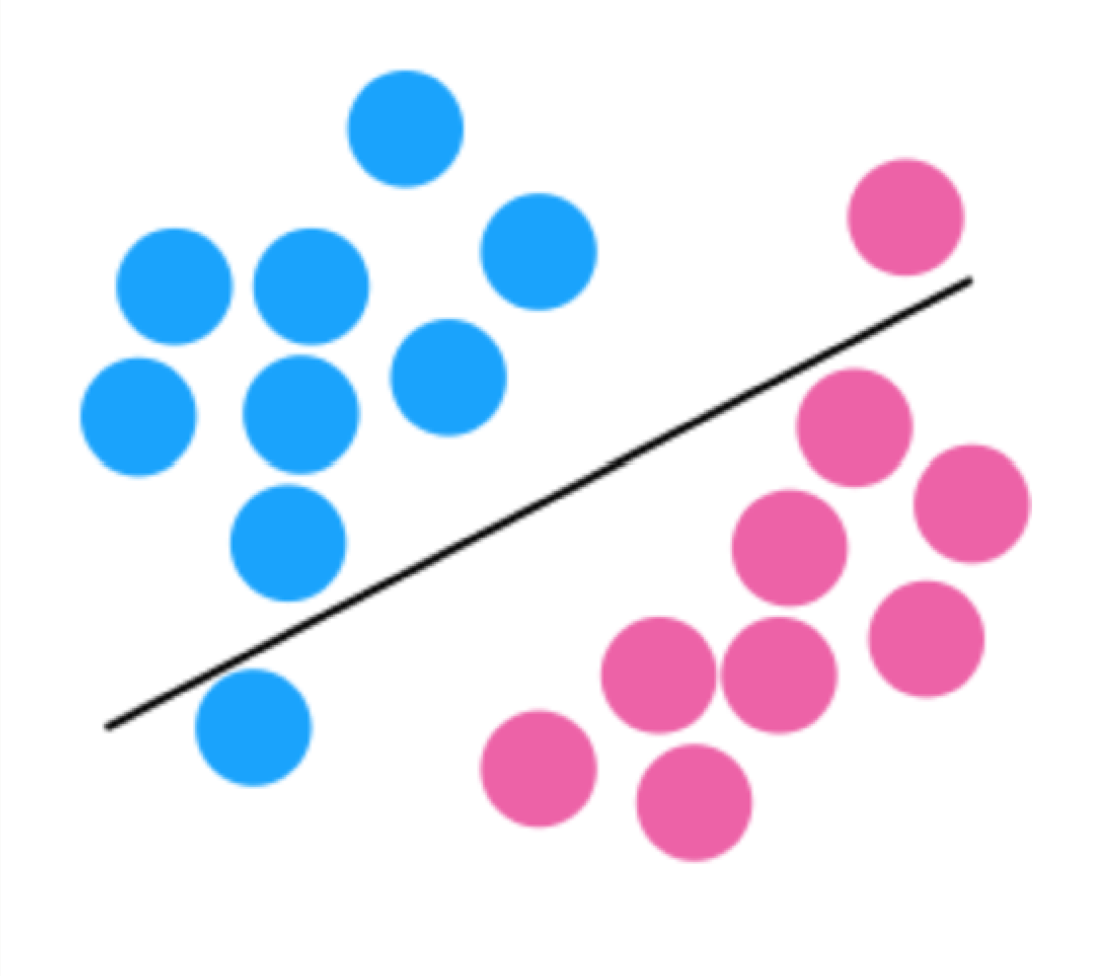
Complete

Jump to your position on the leaderboard ▾

SVM(支持向量機)

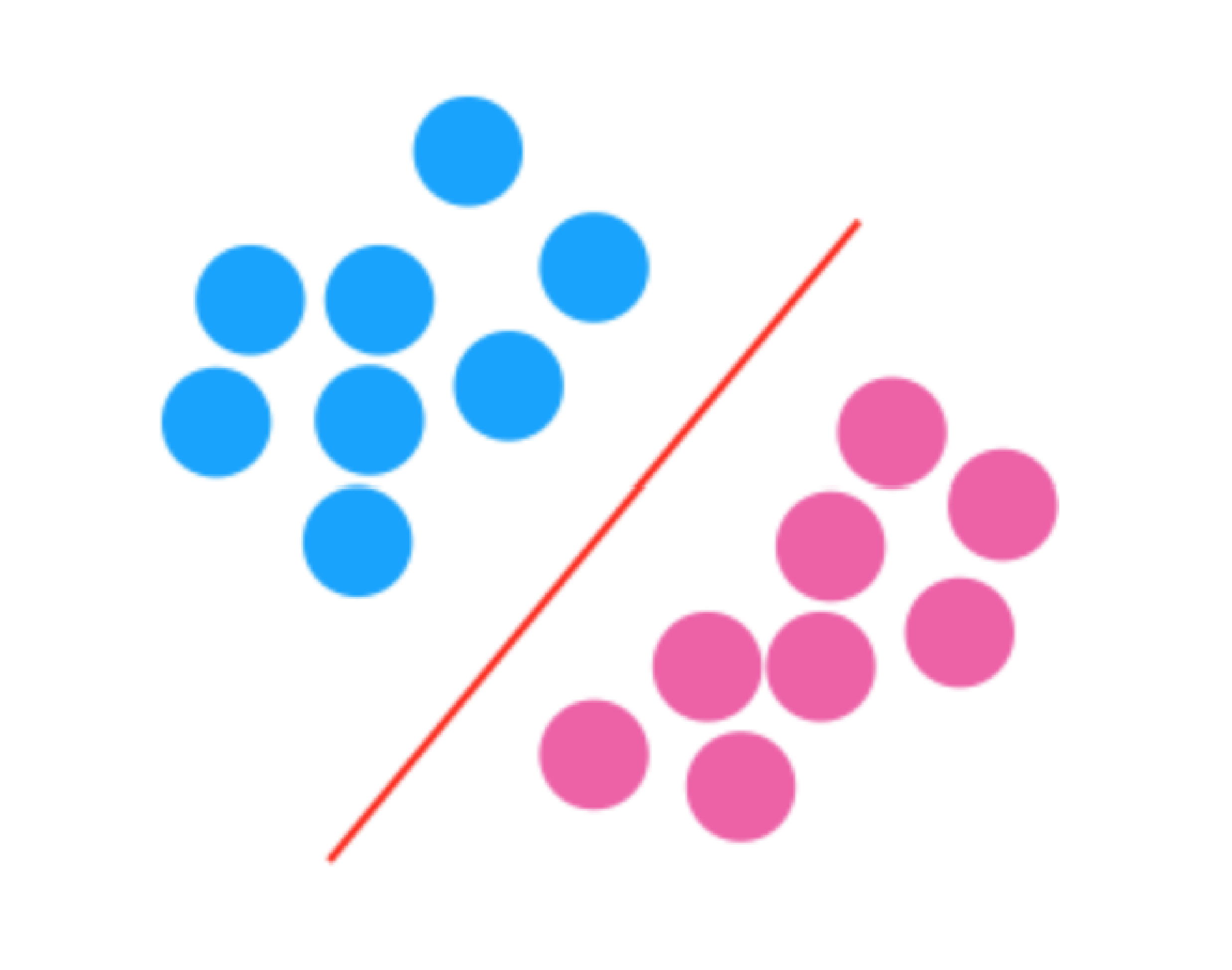
我們依照Logistic Regression來將兩種不同顏色的球分類

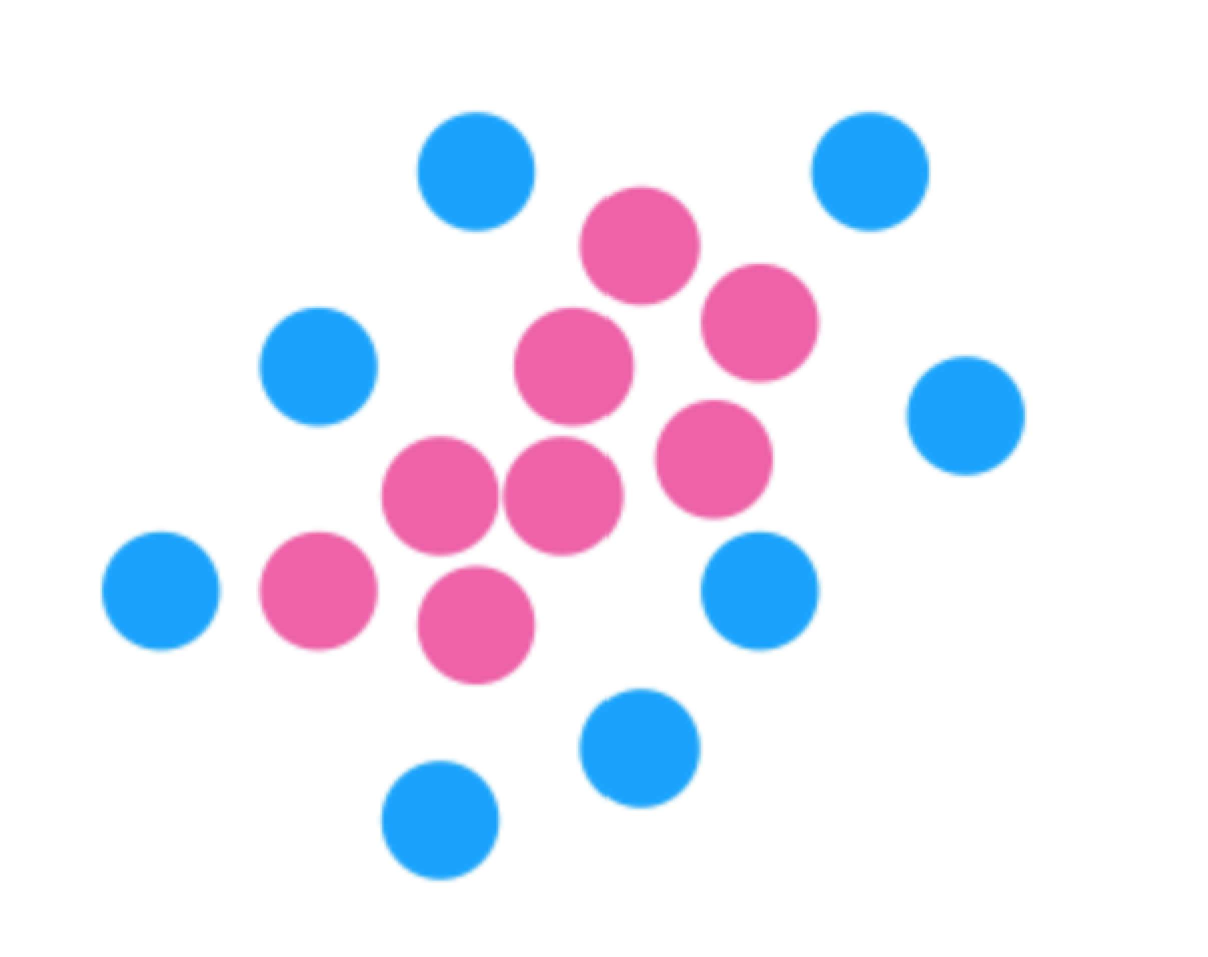




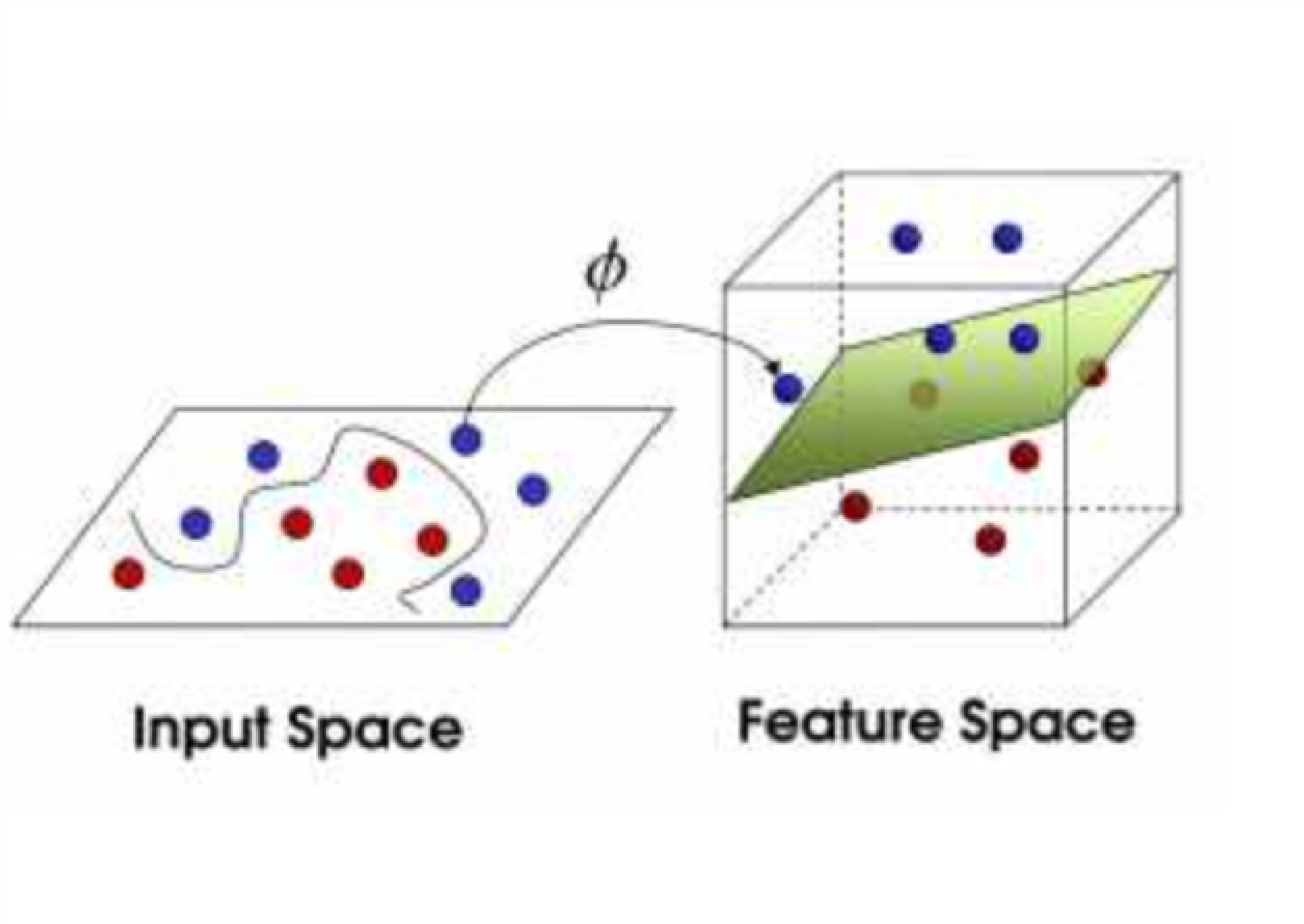
但當增加新的球時，產生了問題

進行微調



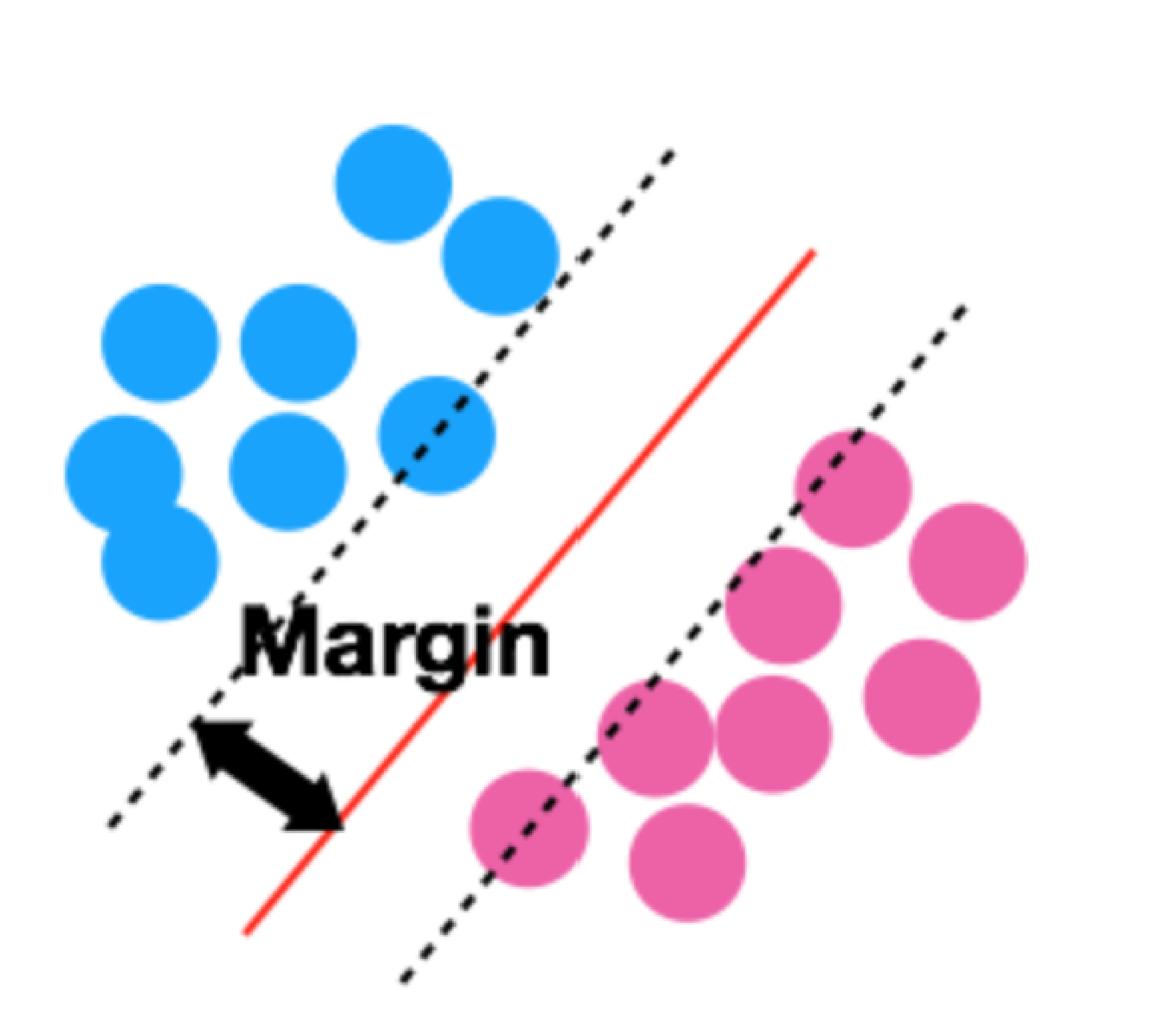


若今天變得不容易分類，可以用甚麼方法?



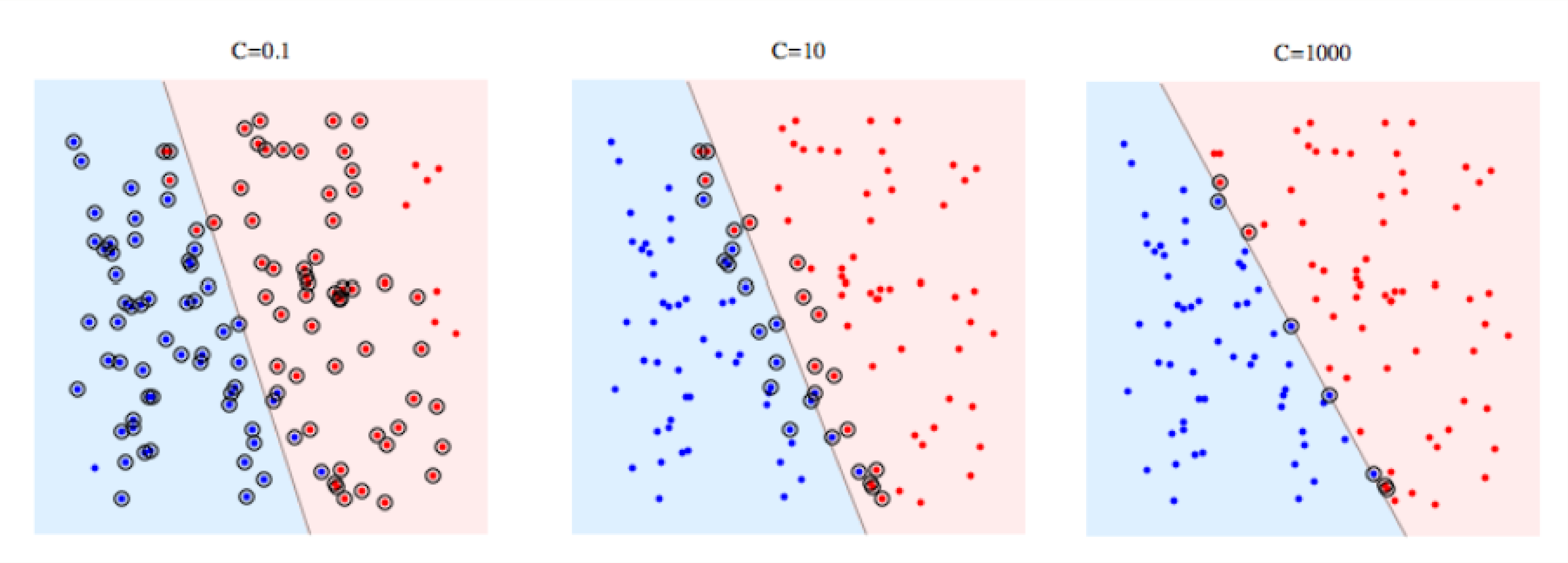
把樣本映射到高維度空間，找到一個超平面將這些樣本做有效的切割

以直線來說，首先紅色的線會創造兩條黑色平行於紅色線的虛線，並讓黑線平移碰到最近的一個點，紅線到黑線的距離稱為Margin，而SVM就是透過去找Margin最大的那個紅線，來找最好的線

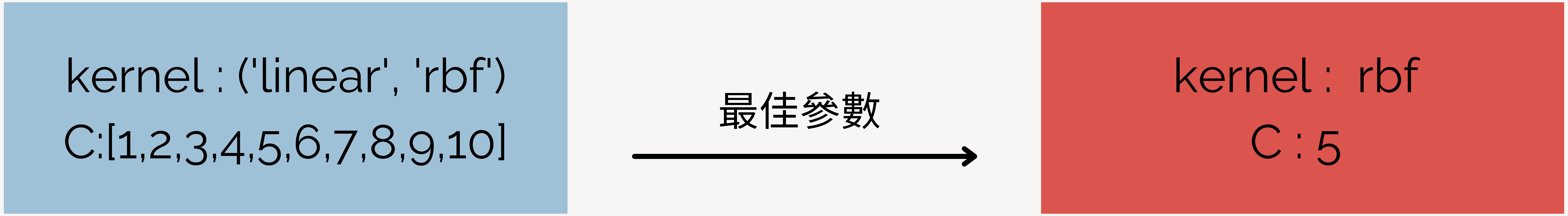


SVM 的參數 C ：控制錯誤分類的懲罰（Penalty）

C越小，代表容錯越大，越多support vectors，可以追求更大的margin
C越大，代表容錯越小，越少support vectors，容易overfitting



GridSearch



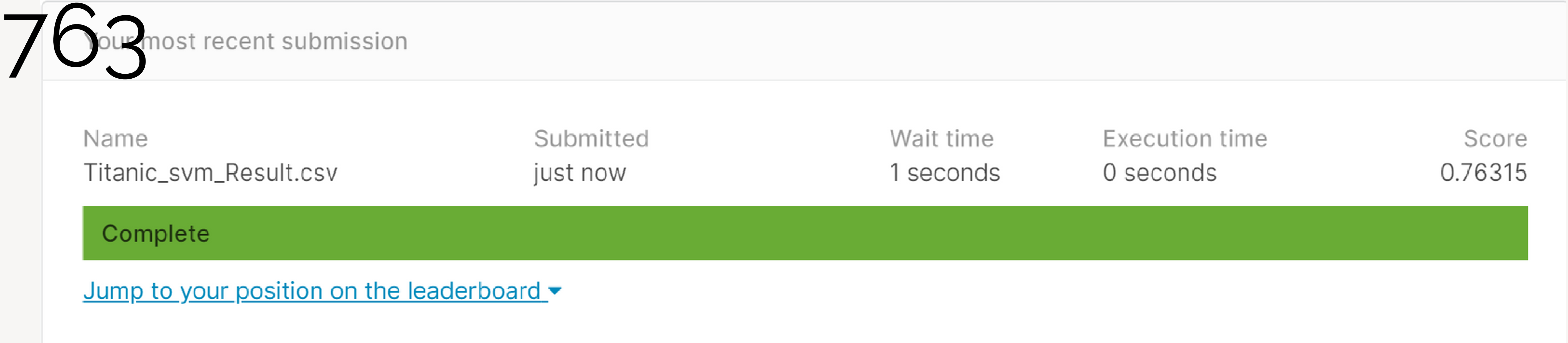
kernel : ('linear', 'rbf')
C:[1,2,3,4,5,6,7,8,9,10]

最佳參數

kernel : rbf
C : 5

使用 KFold 得到的準確率 :
trainset : 0.83
ValidationSet :   0.82

Kaggle :
0.763



Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Titanic_svm_Result.csv	just now	1 seconds	0 seconds	0.76315

Complete

Jump to your position on the leaderboard ▾

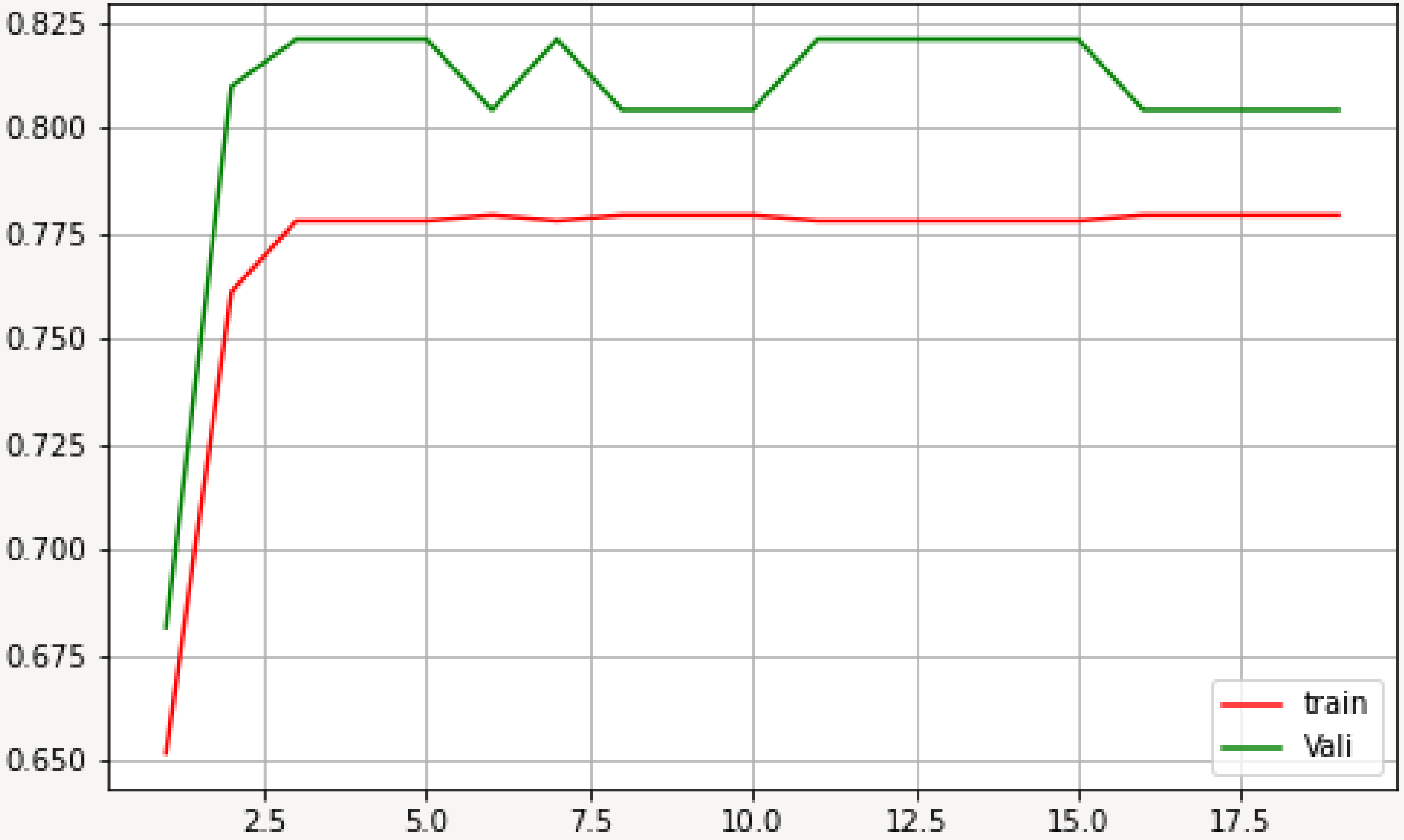
KNN

特徵篩選:KBest

```
3.07656882e+01,
4.65283564e-01,
9.85624300e+00,
1.40326010e+00,
5.28829682e-01,
1.76779953e-03,
7.27959399e+01,
1.25821192e+02,
6.99963774e+00
```

```
'Embarked',
'family_size',
'Age__Children',
'Age__Teenage',
'Age__Adult',
'Age__Elder',
'Fare_bin',
'Sex_female',
'boy'
```

準確率：1~20個鄰居


0.825
0.800
0.775
0.750
0.725
0.700
0.675
0.650
2.5
5.0
7.5
10.0
12.5
15.0
17.5
train
Vali

建模

n_neighbors : 5

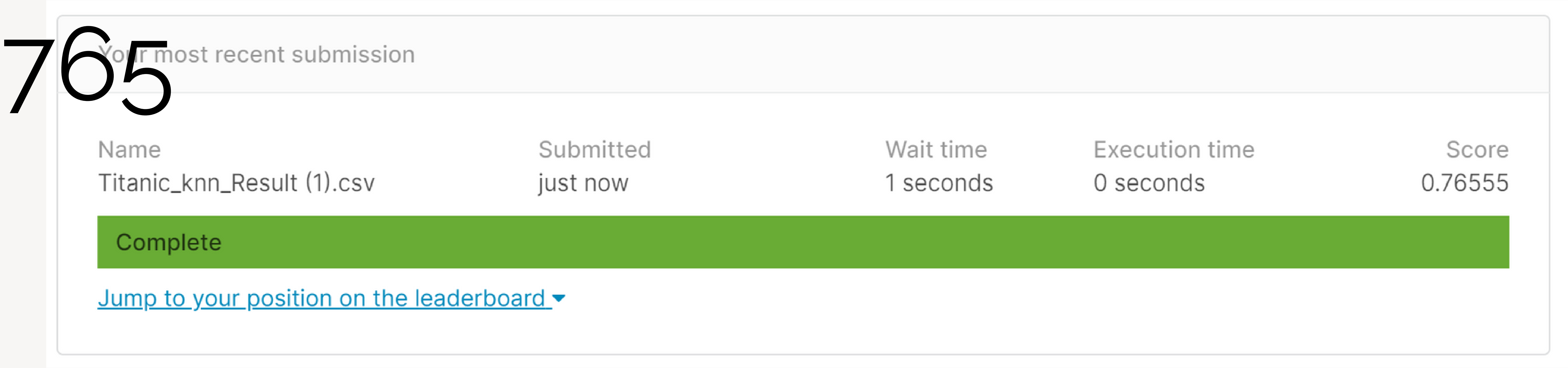
使用 KFold 得到的準確率 :

trainset : 0.79

ValidationSet :   0.77

Kaggle :

0.765



Your most recent submission

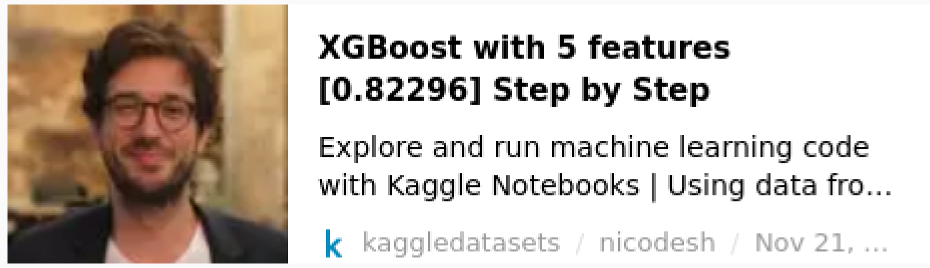
Name	Submitted	Wait time	Execution time	Score
Titanic_knn_Result (1).csv	just now	1 seconds	0 seconds	0.76555

Complete

Jump to your position on the leaderboard

Resource

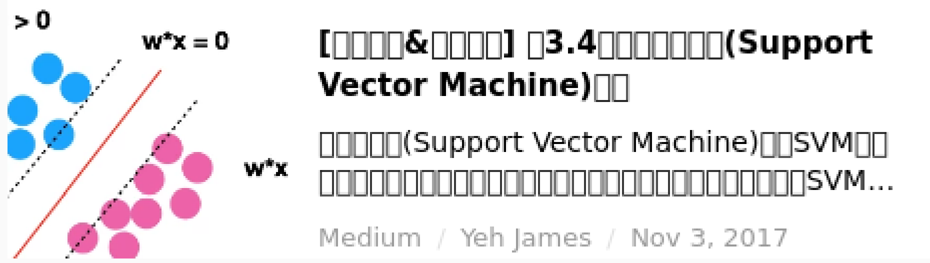
- http://www.ngensis.com/titanic/TIT-34.JPG



XGBoost with 5 features [0.82296] Step by Step

Explore and run machine learning code with Kaggle Notebooks | Using data fro…

kaggledatasets / nicodesh / Nov 21, …



[□□□□&□□□□] □3.4□□□□□□□(Support Vector Machine)□□

□□□□□(Support Vector Machine)□□SVM□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□SVM…

Medium / Yeh James / Nov 3, 2017

[□□□□□□] Kaggle□□-□□□□□□□□(Top 3%)

□□□□□□□□□□□□□

Medium / YL / Jun 16, 2018