Project 2 第四組 Titanic

Machine Learning from Disaster

楊恆軒 陳道明 杜彥陵 徐嘉澤

12.05.2019 Intelligent Integration of Enterprise

OUTLINE

Introduction

Methodology

Random Forest: Feature Selection

Neural Network with Keras

Result & Conclusion

Introduction 1

Competition on Kaggle

Titanic: Machine Learning from Disaster

This is the legendary Titanic ML competition –
the best, first challenge for you to dive into ML competitions
and familiarize yourself with how the Kaggle platform works.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, we ask you to build a predictive model that answers the **question: "what sorts of people were more likely to survive?"** using passenger data (ie name, age, gender, socio-economic class, etc).

"Untergang der Titanic", as conceived by Willy Stöwer, 1912

The dataset

Kaggle提供的資料集

Train.csv

891筆

- 891筆乘客資料
- 包含每位乘客的生死結果
- 本次project僅使用train.csv進行驗證

Test.csv

418筆

- 418筆乘客資料
- 僅提供乘客的資訊，無生死結果
- Kaggle比賽上傳使用

Model score is the percentage of passengers correctly predict,
that is known as accuracy.

Variables

Train.csv的變數

欄位變數	定義	值或特性
PassengerId	乘客ID編號	train.csv有891位 test.csv有418位 共1,309位乘客
Survived	是否生還	0 (no) / 1 (yes)
Pclass	船艙等級	1 (1st) / 2 (2nd) / 3 (3rd)
Name	姓名	包含其稱謂
Sex	性別	male / female
Age	年齡	浮點數
SibSp	在船上的兄弟姊妹和配偶人數	整數
Parch	在船上家族的父母及小孩人數	整數
Ticket	船票編號	文字
Fare	船票價格	浮點數
Cabin	船艙號碼	文字
Embarked	登船口岸	C (Cherbourg) / Q (Queenstown) / S (Southampton)

5W1H

- WHO: Jack & Rose
- WHAT
- WHY
- WHERE: somewhere in the sea
- WHEN: 1912/04/15
- HOW

Method

2

Random Forest

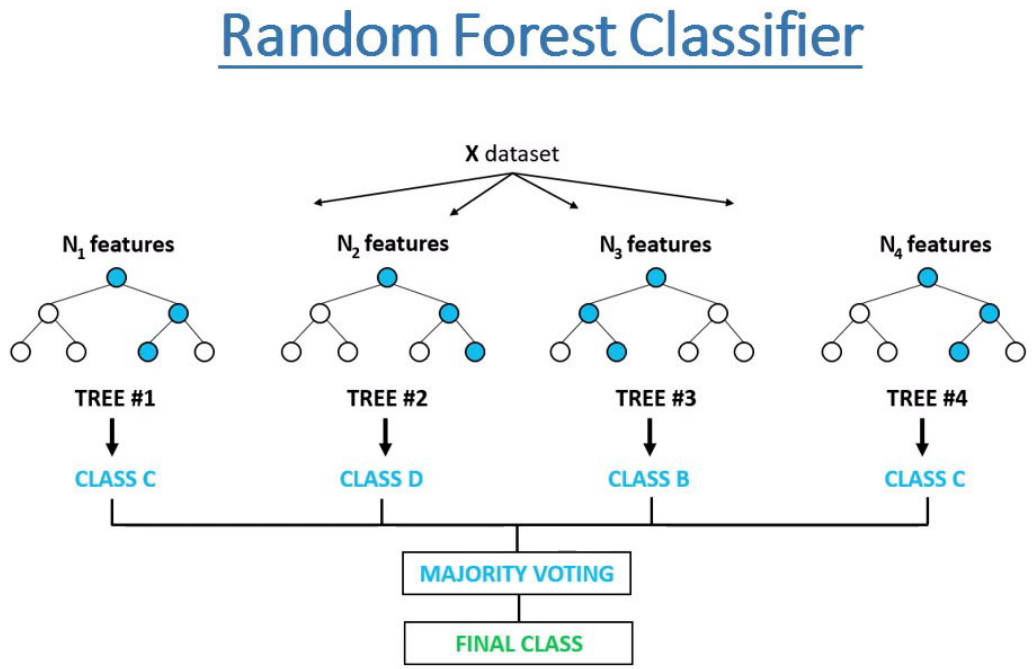
本次project所使用的演算法 (1/2)

隨機森林：多個決策樹的分類器

Bagging:
「每次抽取n個樣本，
抽後放回，
總共抽m次。」

每次用n個樣本
建構best gain決策樹，
再對m棵決策樹進行**多數決投票**。

多產生幾棵樹，降低整個森林的variance，
能有**較高的穩定性和準確性**，
避免overfitting發生。



https://www.youtube.com/watch?v=goPiwckWE9M

Package used

Random Forest 所使用的 python 套件

Numpy

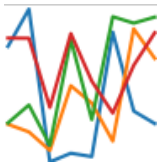
高階維度陣列與矩陣運算

Matplotlib

使用其Seaborn進階圖表繪製功能

Pandas

數據分析的data frame架構



Scikit-learn

提供許多機器學習的基本演算法

Neural Network

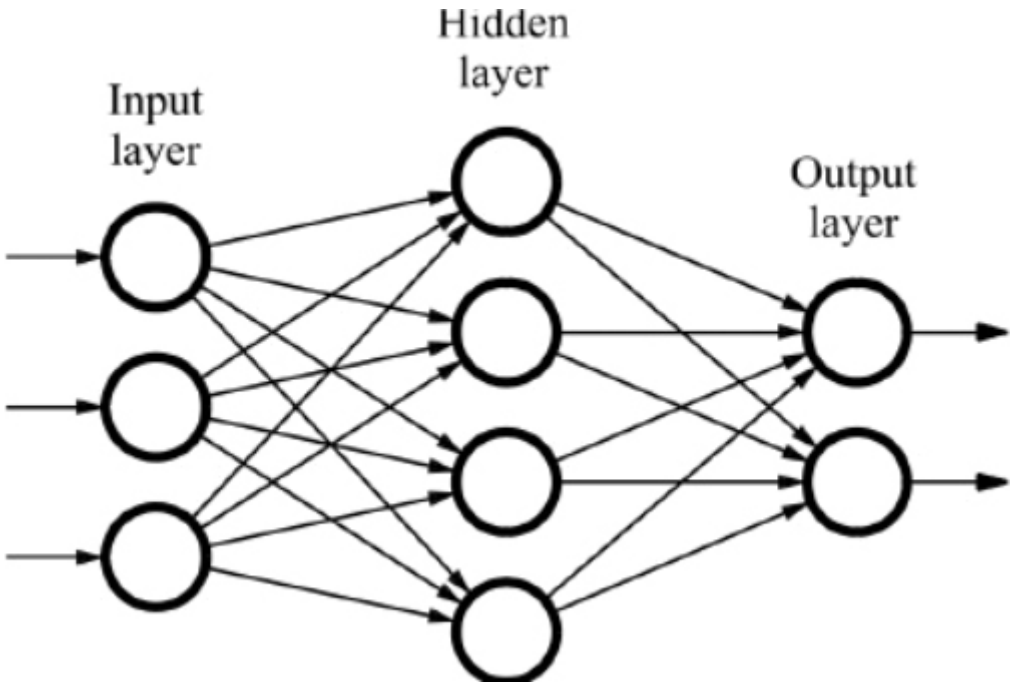
本次project所使用的演算法 (2/2)

類神經網路：輸入-隱藏-輸出

我們選擇多層感知器(MLP)
作為訓練的架構

執行的環境為：
Tensorflow 1.x on Colab

TensorFlow



https://databricks.com/glossary/neural-network

Random Forest: Feature Selection

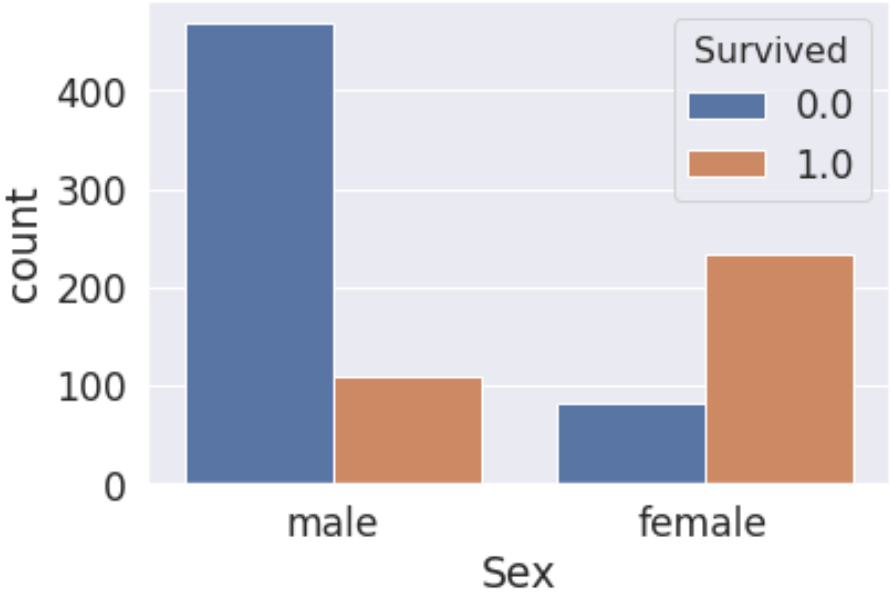
3

Variable overview

‘Sex’ vs. ‘Survived’

	Sex	Survived
0	female	0.742
1	male	0.189

大部分的男性都死亡(僅18.9%存活)；
女性則有將近四分之三(74.2%)生還。

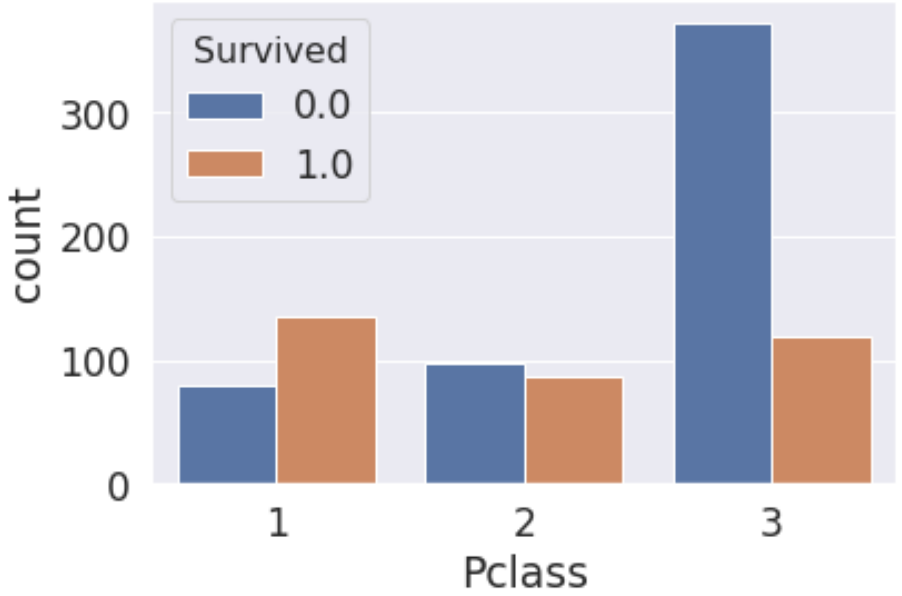


Variable overview

‘Pclass’ vs. ‘Survived’

	Pclass	Survived
0	1	0.630
1	2	0.473
2	3	0.242

一等艙的生還率最高，艙等越低的生還率越低。



Data Preprocessing

資料前處理

‘Sex’ encoding

0 ; 1

將 male 轉為 0
將 female 轉為 1

‘train.csv’

train / test

RF訓練未使用到的data
作為衡量模型的資料(OOB)

define ‘Y’

Survived

預測是否生還(Y)
用其他欄位當作X

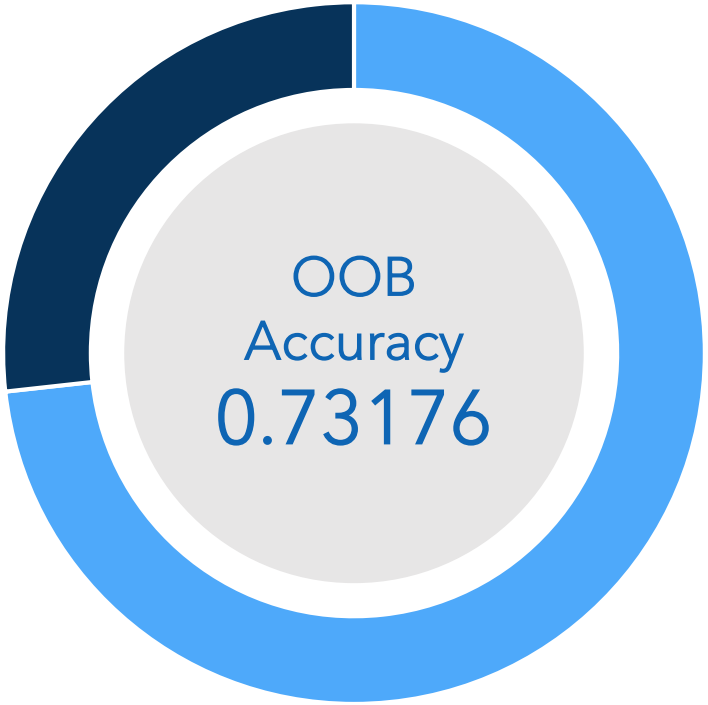
Base Model

原始模型的表現

Random Forest

X[**sex_code, Pclass, …]**, Y

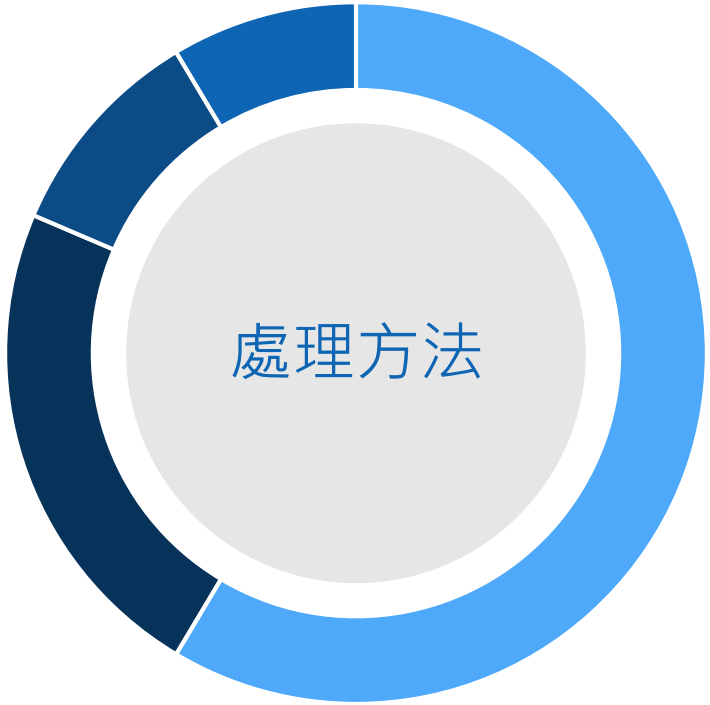
random_state = 2
n_estimators = 250
min_samples_split = 20
oob_score = True



票價資料前處理

票價數據特色

票價和艙等都是屬於彰顯乘客社會地位的一個特徵，我們主觀可以判斷買票價格較高的乘客，由於沈船資訊早就被接收到，故他們的生存機率也較高。



票價分布非常廣及傾斜

票價數據中有遺失值

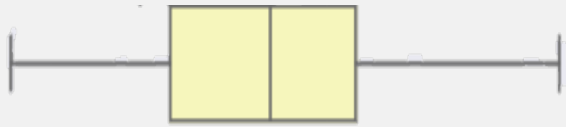
票價資料前處理

票價分布非常廣及傾斜

$$\log_a N$$

- 解決傾斜的問題
- 取log後之作圖可更加美觀

票價數據中有遺失值



- 取票價中位數置換遺失值

	Fare	
Survived	**0.0**	**1.0**
Pclass		
1	44.75	77.958
2	13.00	21.000
3	8.05	8.517

票價資料前處理

置換遺失值

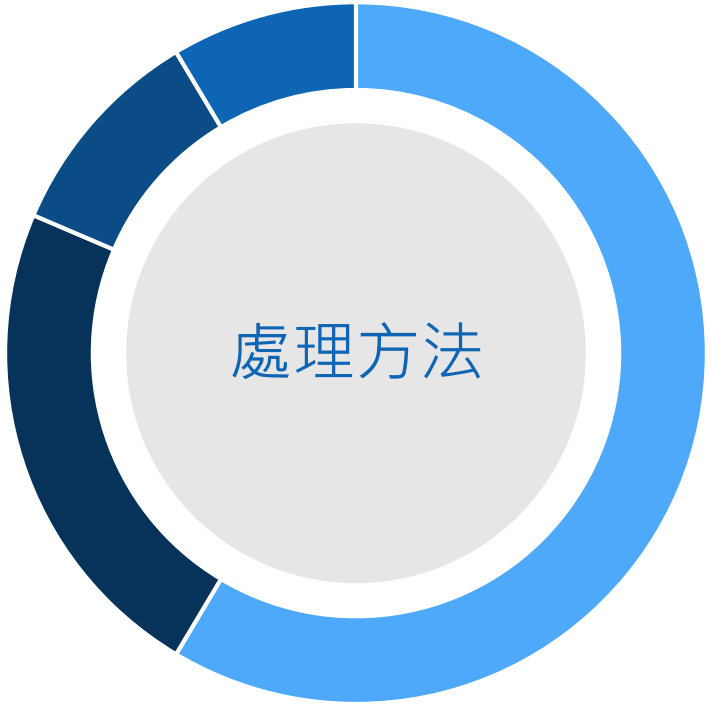
```
# Filling missing values
df_data['Fare'] = df_data['Fare'].fillna(df_data['Fare'].median())
```

票價資料前處理

票價區間分類

票價數據特色

票價屬於一個連續型資料，如果不分類的話會使得特徵值無法被歸納出來，意思是如果把資料切無限段，也就是保持原有資料的樣子會使得分析沒有意義。



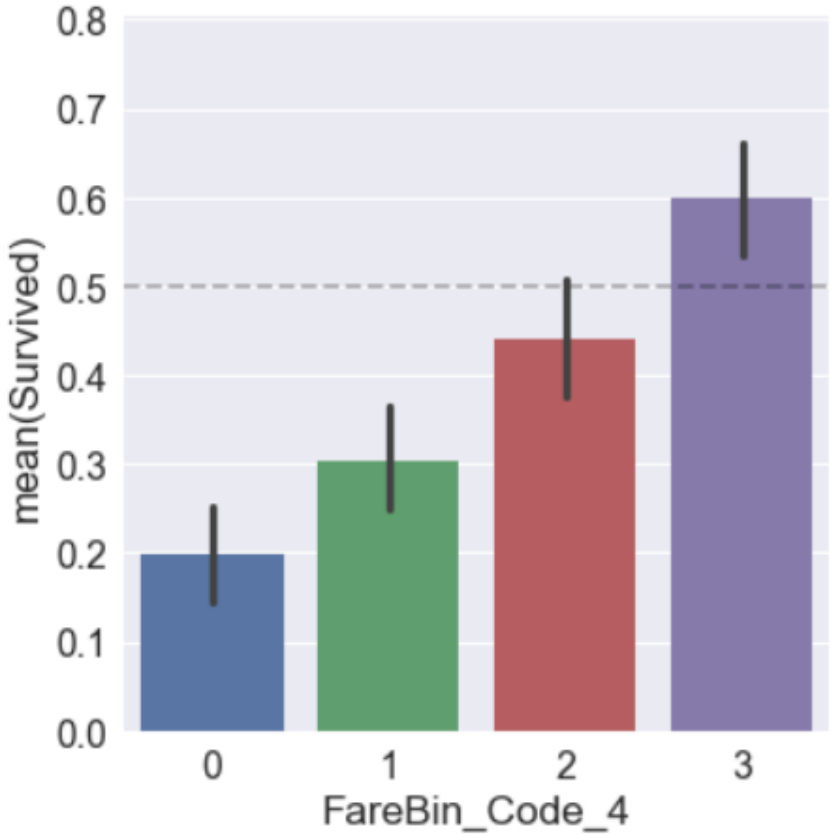
S 票價區間分類切割過於細碎

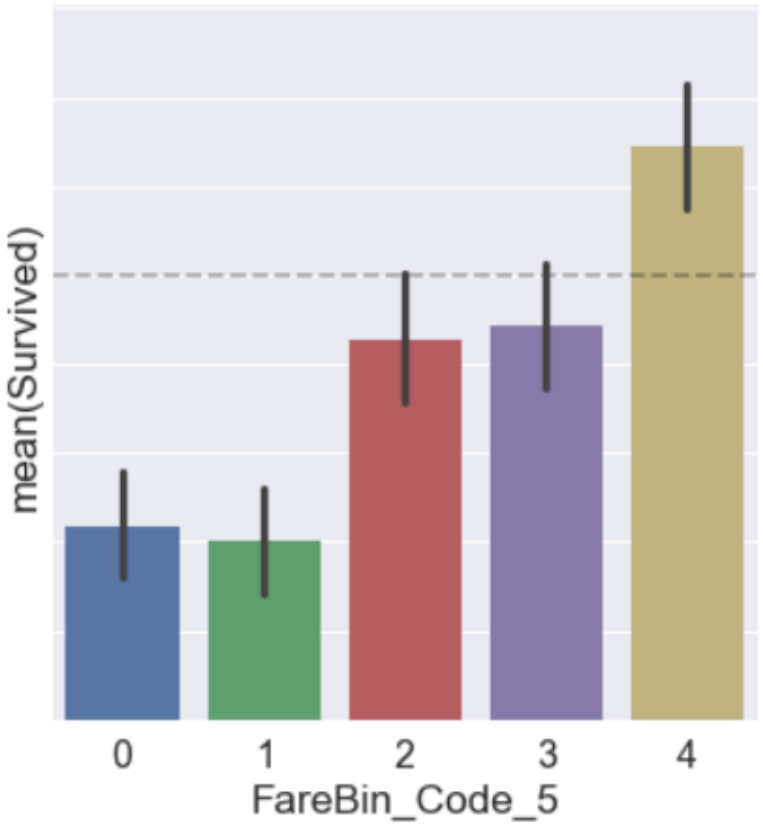
L 票價區間分類切割過於粗略

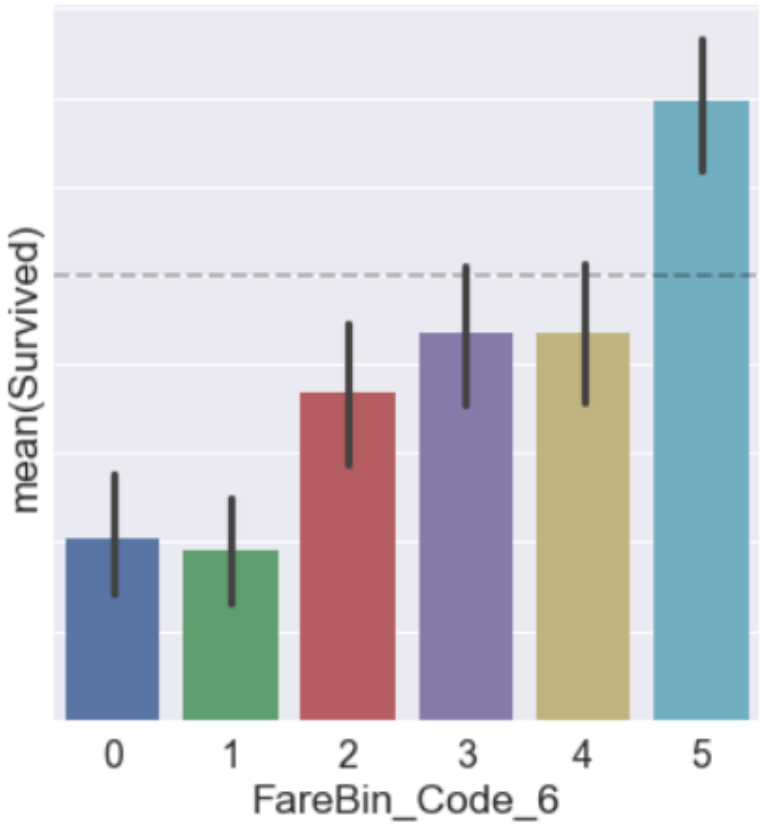
Pclass	1	2	3
FareBin_Code_4			
0	8	6	323
1	0	128	193
2	77	104	147
3	238	39	46

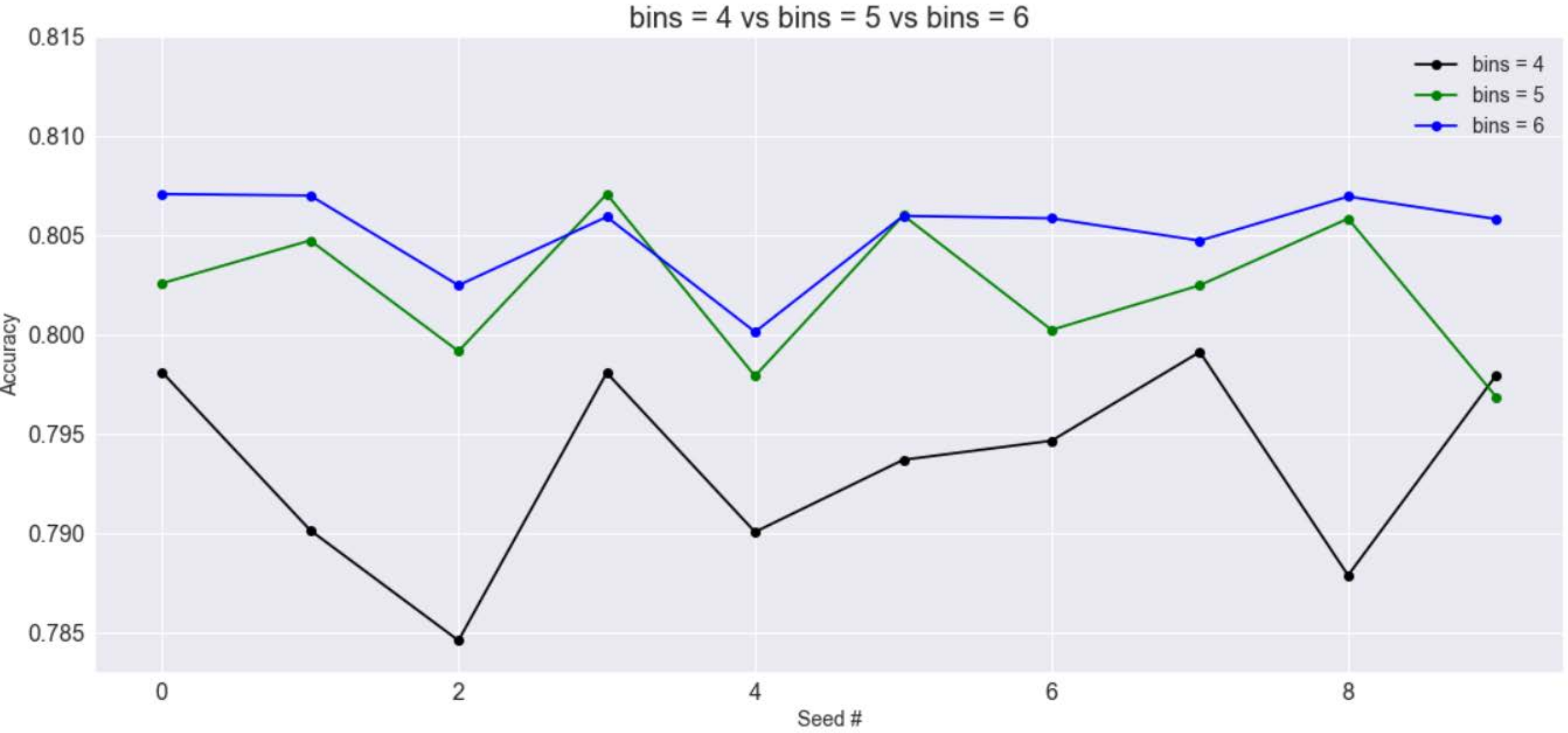
Pclass	1	2	3
FareBin_Code_5			
0	8	6	261
1	0	36	218
2	0	124	132
3	95	99	71
4	220	12	27

Pclass	1	2	3
FareBin_Code_6			
0	8	6	222
1	0	0	218
2	0	128	76
3	14	83	128
4	118	48	46
5	183	12	19








bins = 4 vs bins = 5 vs bins = 6
bins = 4
bins = 5
bins = 6
0.815
0.810
0.805
0.800
0.795
0.790
0.785
Accuracy
0
2
4
6
8
Seed #

票價資料前處理

預測結果

```
b4, b5, b6 = ['Sex_Code', 'Pclass','FareBin_Code_4'], ['Sex_Code','Pclass','FareBin_Code_5'],\
['Sex_Code','Pclass','FareBin_Code_6']
b4_Model = RandomForestClassifier(random_state=2,n_estimators=250,min_samples_split=20,oob_score=True)
b4_Model.fit(X[b4], Y)
b5_Model = RandomForestClassifier(random_state=2,n_estimators=250,min_samples_split=20,oob_score=True)
b5_Model.fit(X[b5], Y)
b6_Model = RandomForestClassifier(random_state=2,n_estimators=250,min_samples_split=20,oob_score=True)
b6_Model.fit(X[b6], Y)
print('b4 oob score :%.5f' %(b4_Model.oob_score_))
print('b5 oob score :%.5f '%(b5_Model.oob_score_))
print('b6 oob score : %.5f' %(b6_Model.oob_score_))
```

```
b4 oob score :0.80584
b5 oob score :0.81033
b6 oob score : 0.80135
```

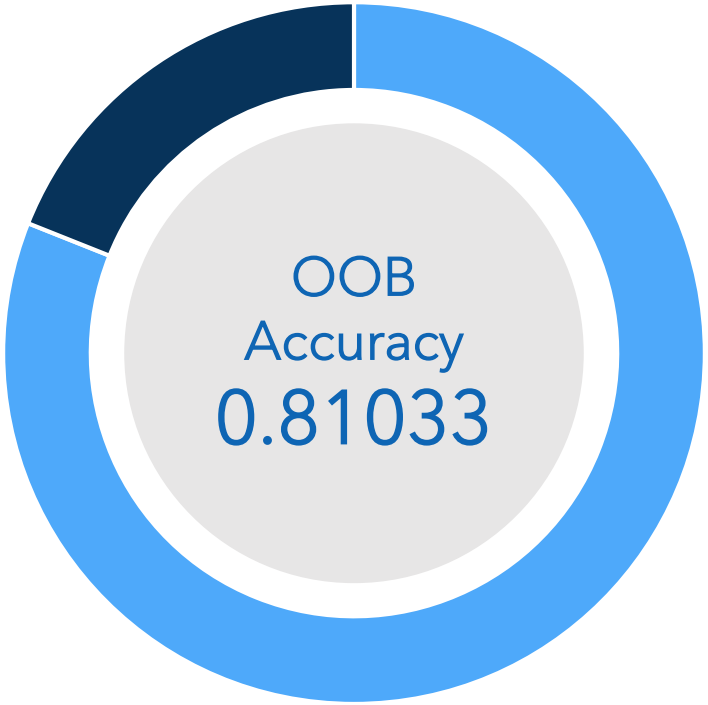
Fare Model

票價模型的表現

Random Forest

X[**FareBin_Code_5, …**], Y

random_state = 2
n_estimators = 250
min_samples_split = 20
oob_score = True



連結前處理

連結特色

再發生意外時，家人朋友常常互相幫助，雖然資料上的兄弟姊妹數(SibSp)和父母小孩數(Parch)它們與一個人是否存活沒有直接關係，但是通過和票根(Ticket)比對，可以發現他是否獨自在這艘船上，而這對於他的生存機率也有很大的影響。

票根資訊

父母小孩數資訊

兄弟姊妹數資訊

連結前處理

票根的特徵(Ticket)

```
[ ]  df_train['Ticket'].describe()
```

```
count      891
unique     681
top       1601
freq         7
Name: Ticket, dtype: object
```

建立一個新的特徵家庭人數特徵(Family_size)

```
[ ]  # Family_size
     df_data['Family_size'] = df_data['SibSp'] + df_data['Parch'] + 1
```

連結前處理

建立持有相同票根的DataFrame，並顯示姓名、票價、艙位、家庭人數

```
deplicate_ticket = []
for tk in df_data.Ticket.unique():
    tem = df_data.loc[df_data.Ticket == tk, 'Fare']
    #print(tem.count())
    if tem.count() > 1:
        #print(df_data.loc[df_data.Ticket == tk,['Name','Ticket','Fare']])
        deplicate_ticket.append(df_data.loc[df_data.Ticket == tk,['Name','Ticket','Fare','Cabin','Family_size','Survived']])
deplicate_ticket = pd.concat(deplicate_ticket)
deplicate_ticket.head(14)
```

	Name	Ticket	Fare	Cabin	Family_size	Survived
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	71.2833	C85	2	1.0
234	Cumings, Mr. John Bradley	PC 17599	71.2833	C85	2	NaN
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	53.1000	C123	2	1.0
137	Futrelle, Mr. Jacques Heath	113803	53.1000	C123	2	0.0
6	McCarthy, Mr. Timothy J	17463	51.8625	E46	1	0.0
146	Hilliard, Mr. Herbert Henry	17463	51.8625	E46	1	NaN
7	Palsson, Master. Gosta Leonard	349909	21.0750	NaN	5	0.0
24	Palsson, Miss. Torborg Danira	349909	21.0750	NaN	5	0.0
374	Palsson, Miss. Stina Viola	349909	21.0750	NaN	5	0.0
567	Palsson, Mrs. Nils (Alma Cornelia Berglund)	349909	21.0750	NaN	5	0.0
389	Palsson, Master. Paul Folke	349909	21.0750	NaN	5	NaN
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	347742	11.1333	NaN	3	1.0
172	Johnson, Miss. Eleanor Ileen	347742	11.1333	NaN	3	1.0
869	Johnson, Master. Harold Theodor	347742	11.1333	NaN	3	1.0

連結前處理

依照觀察來創建一個新的特徵(Connected_Survival)

```
# the same ticket family or friends
df_data['Connected_Survival'] = 0.5 # default
for _, df_grp in df_data.groupby('Ticket'):
    if (len(df_grp) > 1):
        for ind, row in df_grp.iterrows():
            smax = df_grp.drop(ind)['Survived'].max()
            smin = df_grp.drop(ind)['Survived'].min()
            passID = row['PassengerId']
            if (smax == 1.0):
                df_data.loc[df_data['PassengerId'] == passID, 'Connected_Survival'] = 1
#print
print('people keep the same ticket: %.0f '%len(deplicate_ticket))
print("people have connected information : %.0f"
      %(df_data[df_data['Connected_Survival']!=0.5].shape[0]))
df_data.groupby('Connected_Survival')[['Survived']].mean().round(3)
```

```
people keep the same ticket: 596
people have connected information : 294
```

Connected_Survival	Survived
0.5	0.283
1.0	0.728

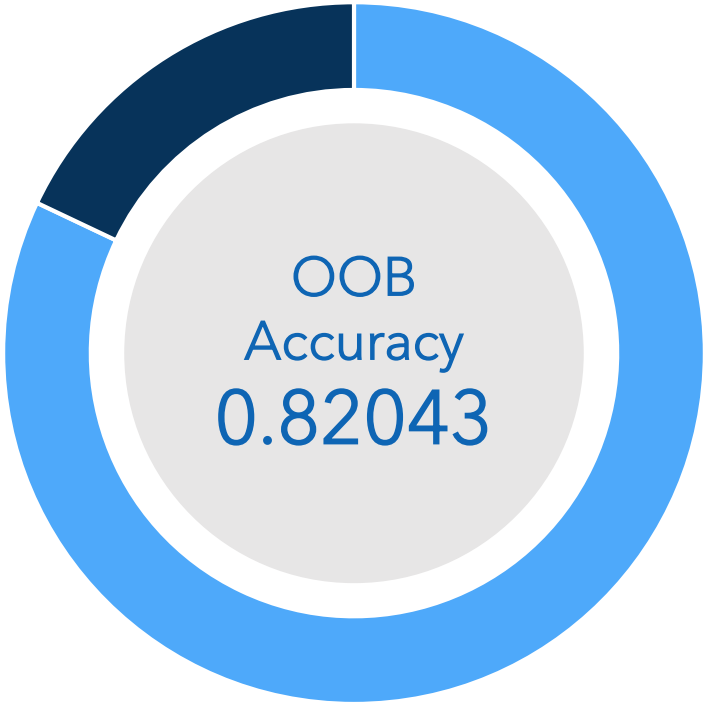
Connect Model

連結模型的表現

Random Forest

X[**Connected_Survival, …]**, Y

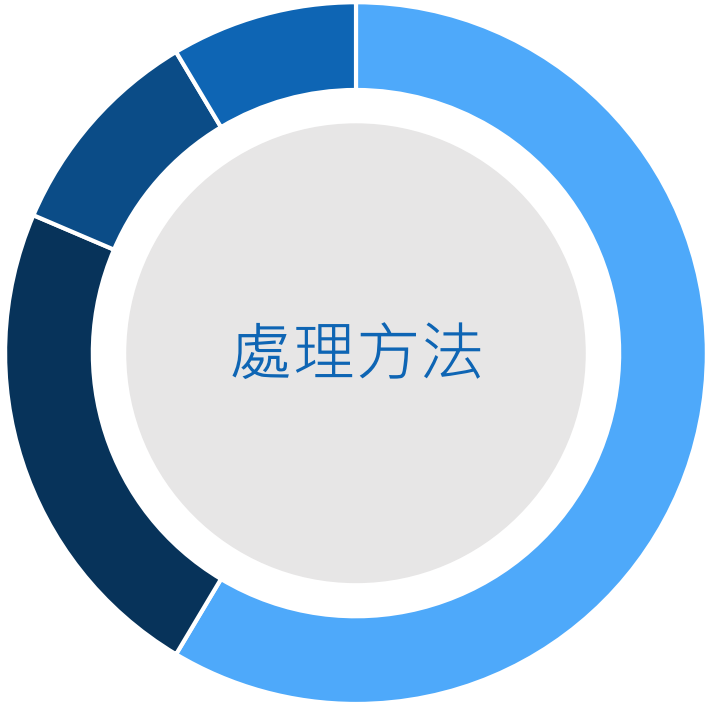
random_state = 2
n_estimators = 250
min_samples_split = 20
oob_score = True



年齡資料分析

年齡數據特色

由於年齡的缺失值較多，又因為準確率受性別及艙等的影響，所以若年齡缺失值在兩大特徵中的分布不一，可能會影響到後續的預測，故分別與艙等及性別做分析。

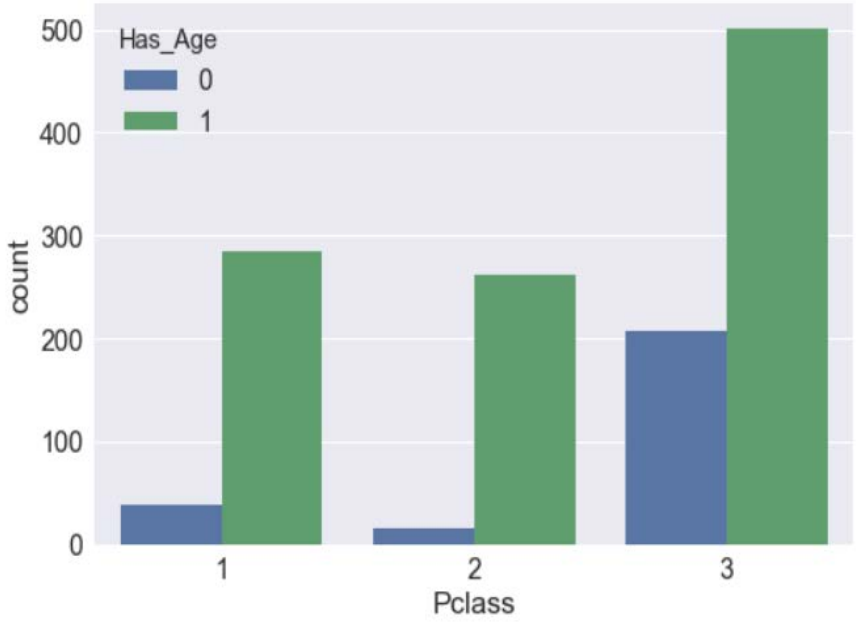


年齡與艙等交叉分析

年齡與性別交叉分析

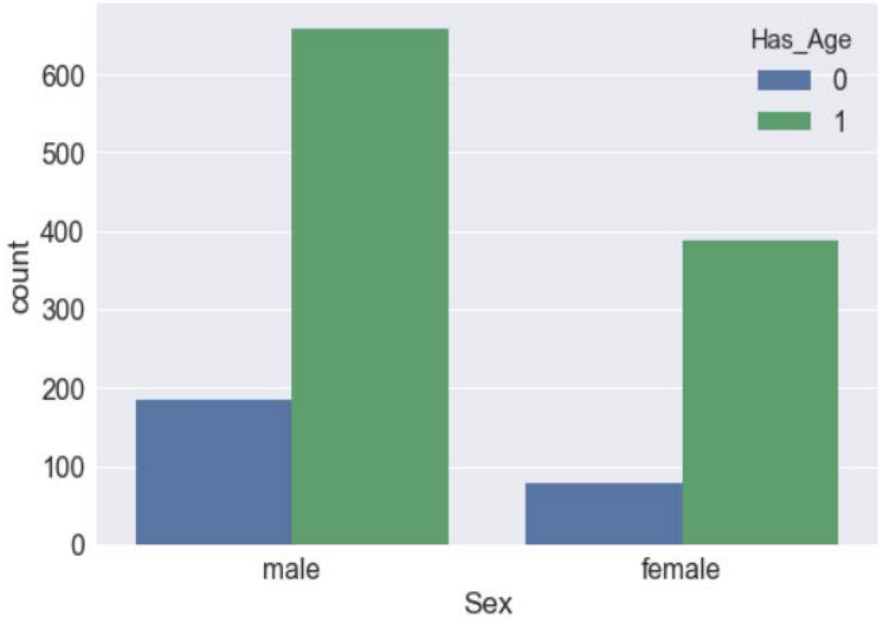
年齡資料分析

年齡與艙等之交叉分析



年齡缺失值大部分在3等艙

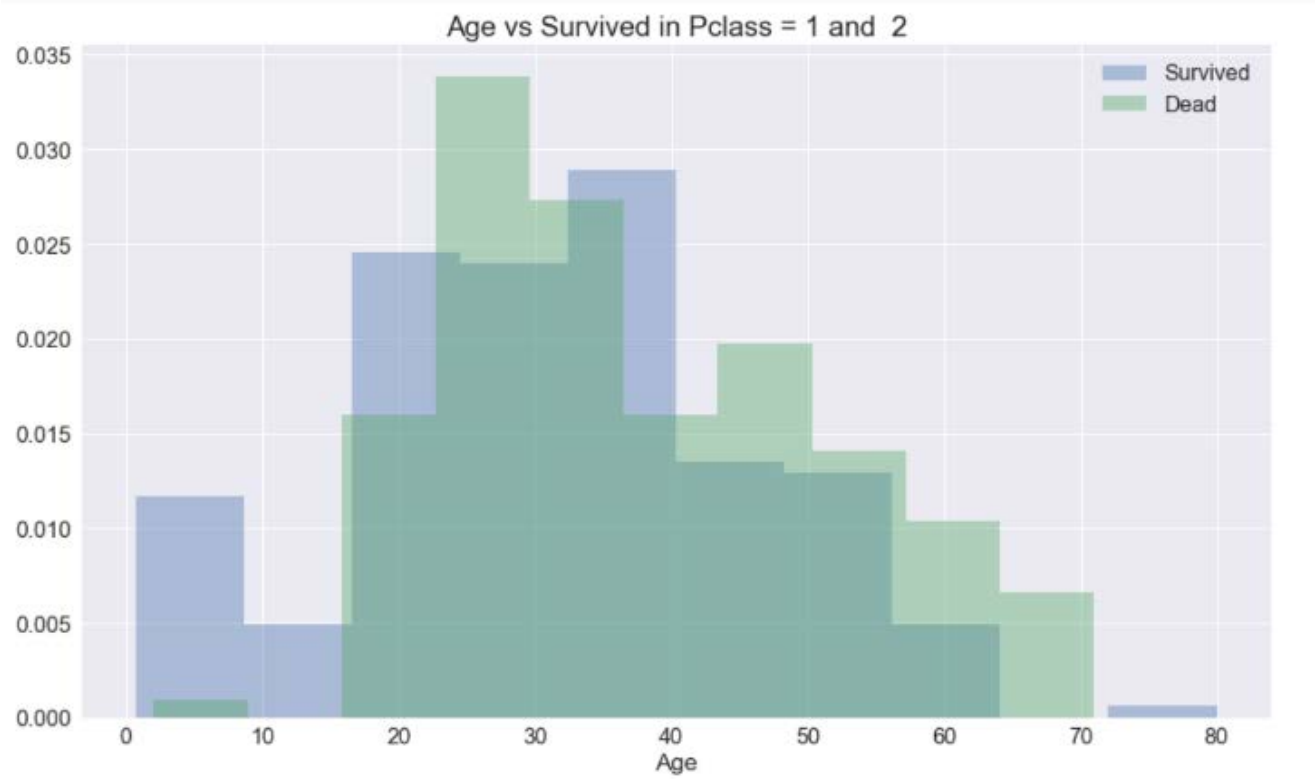
年齡與性別之交叉分析



年齡缺失值性別比 男:女=21.9%:16.7%

年齡資料分析

1、2艙之中，年齡對存活與否的影響



大約<16歲的部分生存率較高
大約>16歲的部分對年齡來說不是一個顯著的特徵
大約70~80歲的部分不列入採計

→

將資料分為<16歲及>16歲的2元特徵
<16歲為1；>16歲為0

年齡資料分析

以稱謂中位數來填補缺失值

```
# extracted title using name
df_data['Title'] = df_data.Name.str.extract(' ([A-Za-z]+)\.', expand=False)
df_data['Title'] = df_data['Title'].replace(['Capt', 'Col', 'Countess', 'Don',
                                             'Dr', 'Dona', 'Jonkheer',
                                              'Major','Rev','Sir'],'Rare')
df_data['Title'] = df_data['Title'].replace(['Mlle', 'Ms','Mme'],'Miss')
df_data['Title'] = df_data['Title'].replace(['Lady'],'Mrs')
df_data['Title'] = df_data['Title'].map({"Mr":0, "Rare" : 1, "Master" : 2,"Miss" : 3, "Mrs" : 4 })
Ti = df_data.groupby('Title')['Age'].median()
Ti
```

started 20:08:34 2018-06-16, finished in 29ms

```
Title
0    29.0
1    47.0
2     4.0
3    22.0
4    36.0
Name: Age, dtype: float64
```

0: 先生 - 29歲
1: 罕見稱謂 - 47歲
2: 小孩 - 4歲
3: 小姐- 22歲
4: 女士 - 36歲

```
Ti_pred = df_data.groupby('Title')['Age'].median().values
df_data['Ti_Age'] = df_data['Age']
# Filling the missing age
for i in range(0,5):
 # 0 1 2 3 4 5
    df_data.loc[(df_data.Age.isnull()) & (df_data.Title == i),'Ti_Age'] = Ti_pred[i]
df_data['Ti_Age'] = df_data['Ti_Age'].astype('int')
df_data['Ti_Minor'] = ((df_data['Ti_Age']) < 16.0) * 1
```

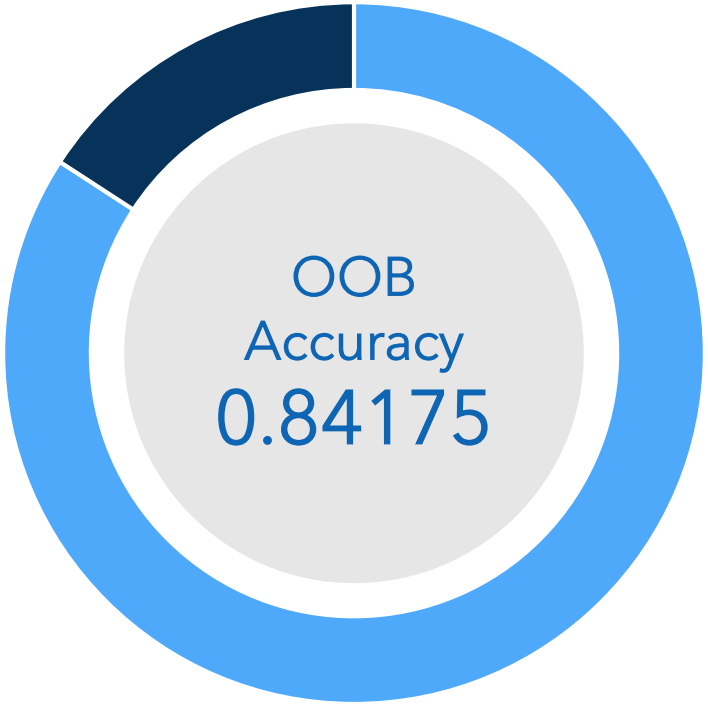
Age Model

年齡模型的表現

Random Forest

X[Ti_Minor, …], Y

random_state = 2
n_estimators = 250
min_samples_split = 20
oob_score = True



RF Performance

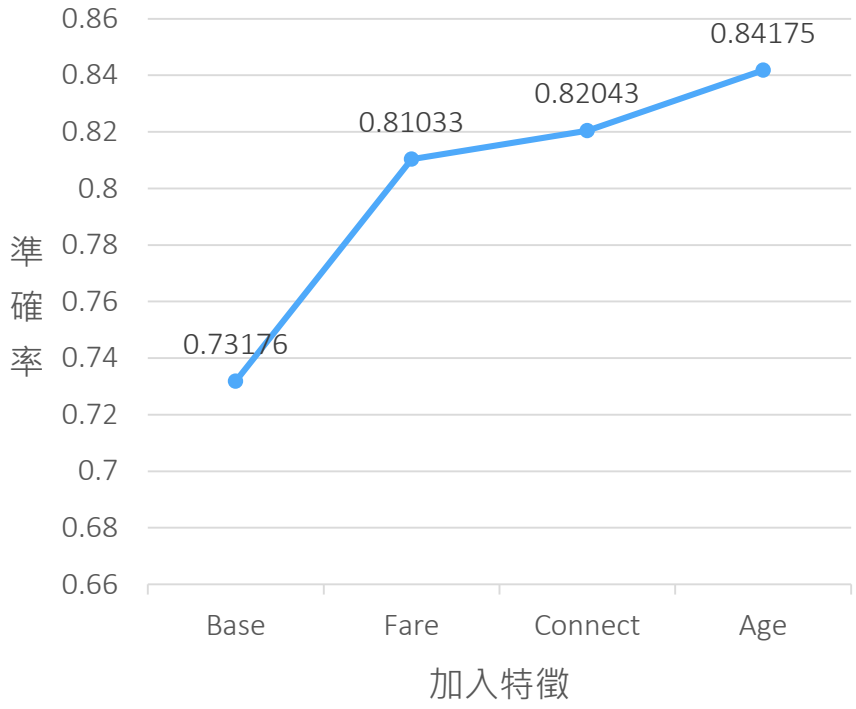
隨機森林的模型表現

以 accuracy 作為指標

依序考量了**票價、連結、年齡**，
經過處理後的特徵值，皆使model的準確率提高。

我們也曾考慮，
將**家庭人數(family_size)**當作一個特徵，
但實際測試後模型表現並未顯著提升。

由Random Forest，我們找出了五個有效Features：
Sex_Code, Pclass, FareBin_Code_5,
Connected_Survival, Ti_Minor



Neural Network with Keras

4

Neural Network Model

類神經網路的架構

Layer	Output	Activation Function
Dense_1	9	ReLU
Dense_2	9	ReLU
Dense_3	5	ReLU
Dense_4	1	Sigmoid

Optimizer
adam

Loss function
binary_crossentropy

Metrics
accuracy

Neural Network Model

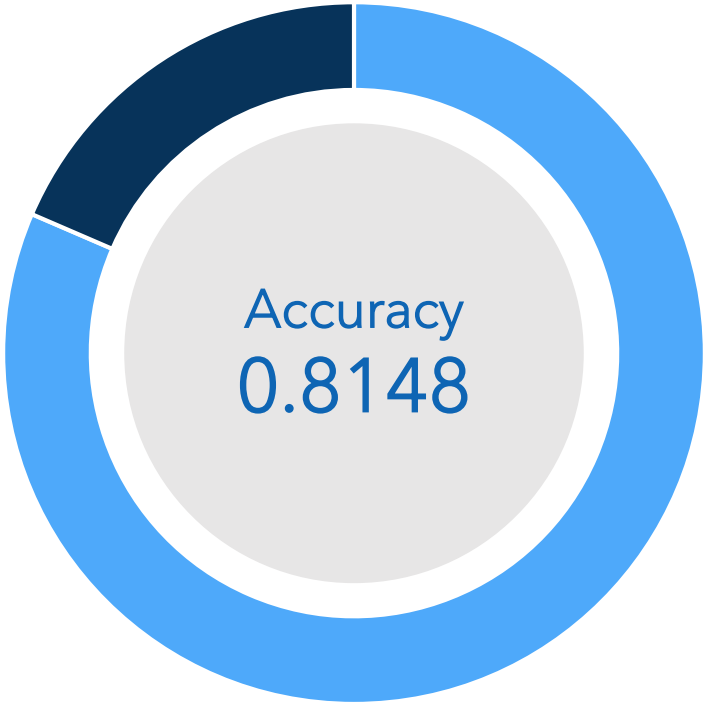
類神經網路的表現

Neural Network

X[**from RF**], Y

Batch size = 32

Epochs = 200



Result & Conclusion

5

Comparison

Random Forest vs. Neural Network

Random Forest

84.18%

- Machine Learning 方法
- 特徵值、結果易解釋
- 在Titanic資料集表現良好

Neural Network

81.48%

- Deep Learning 方法
- 無法得知運算及決策過程
- 未發揮其大資料集分析優勢

Titanic 的資料集，較適合使用 RF 作為分析工具。

Future Works

Hyperparameter tuning

Other algorithm (e.g. XGBoost)

Other application

References

Titanic: Machine Learning from Disaster | Kaggle
https://www.kaggle.com/c/titanic

[機器學習專案] Kaggle競賽-鐵達尼號生存預測(Top 3%)
https://medium.com/@yulongtsai/https-medium-com-yulongtsai-titanic-top3-8e64741cc11f

Neural Network with Keras for Kaggle's Titanic Dataset
https://github.com/liyenhsu/Neural-Network-with-Keras-for-Kaggle-Titanic-Dataset/blob/master/titanic.ipynb


THANK YOU