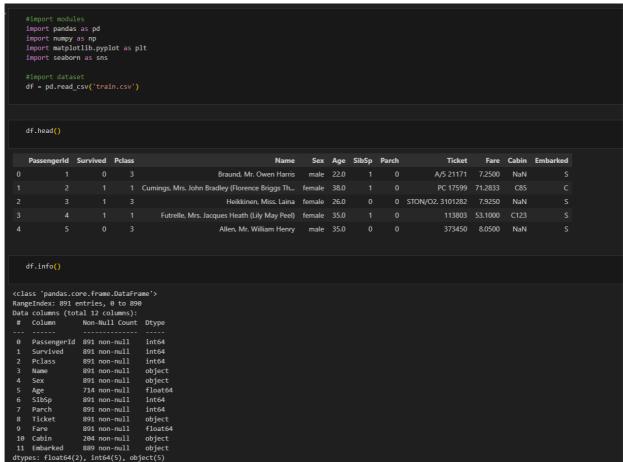
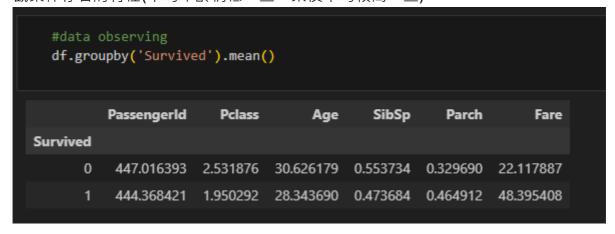
Titanic - Machine Learning from Disaster

312706034 資管碩二 張棨翔

• 讀入訓練用資料集並做初步的資料觀察



觀察倖存者的特性(平均年齡稍低一些,票價平均較高一些)



• 查看各欄位的缺失值狀況



• 調查是否有過半數資料是缺失值的欄位

```
df.isnull().sum() > len(df)/2
PassengerId
            False
Survived
            False
Pclass
            False
Sex
            False
Age
           False
SibSp
            False
Parch
           False
           False
Fare
Cabin
            True
Embarked
            False
dtype: bool
```

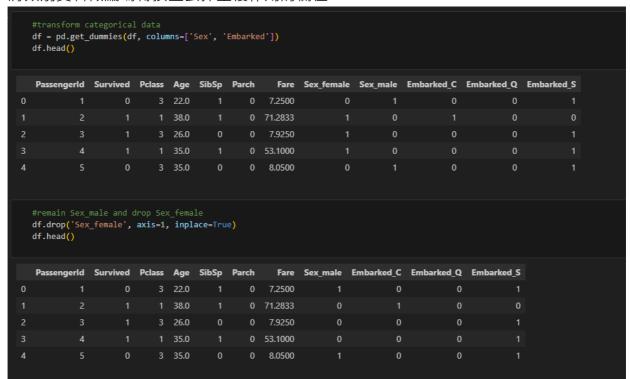
• 把Cabin欄位去掉並把Age的缺失值以平均值填補

```
df.drop('Cabin', axis=1, inplace=True)
       df.head()
       df['Age'].isnull().value_counts()
[14]
    False
             714
    True
    Name: Age, dtype: int64
       df['Age'] = df.groupby('Sex')['Age'].apply(lambda x: x.fillna(x.mean()))
                                                                                               + Code + Markdown
       df.isnull().sum() #2 missing values in Embarked
   PassengerId 0
    Survived
    Pclass
                  0
    Sex
    SibSp
    Parch
                  ø
    Embarked
    dtype: int64
```

• 把Embarked欄位的缺失值以出現次數最多的類別填補(所有缺失值處理完成)

```
#find the value with the highest frequency and fill the missing values
   df['Embarked'].value_counts().idxmax()
   df['Embarked'].fillna(df['Embarked'].value_counts().idxmax(), inplace=True)
   df['Embarked'].value_counts()
    646
    168
     77
Name: Embarked, dtype: int64
   #all missing values are filled
   df.isnull().sum()
PassengerId
              0
Survived
             0
Pclass
             0
             0
Sex
Age
SibSp
Parch
            0
            0
Embarked
            0
dtype: int64
```

• 將類別資料做編碼轉換並去掉重複作用的欄位



• 查看不同欄位變數之間的關聯性並準備將資料分為X和Y兩部分和準備訓練資料

```
df.corr() #correlation between variables
                                                                                  Sex_male Embarked_C Embarked_Q Embarked_S
            Passengerld Survived
                                    Pclass
                                               Age
                                                        SibSp
                                                                  Parch
              1.000000 -0.005007
                                                               -0.001652
                                                                         0.012658
                                                                                   0.042939
                                                                                               -0.001205
Passengerld
                        1.000000 -0.338481 -0.080453
                                                                                  -0.543351
                                                                                               0.168240
                                                                                                                       -0.149683
   Survived
              -0.005007
                                                    -0.035322
                                                               0.081629
                                                                         0.257307
                                                                                                           0.003650
     Pclass
              -0.035144 -0.338481
                                 1.000000 -0.330391 0.083081
                                                               0.018443 -0.549500 0.131900
                                                                                              -0.243292
                                                                                                           0.221009
                                                                                                                       0.074053
      Age
              0.035543 -0.080453 -0.330391 1.000000 -0.236920 -0.182556 0.089079 0.103236
                                                                                                           -0.019970
                                                                                                                       -0.015289
              -0.057527 -0.035322 0.083081 -0.236920 1.000000 0.414838 0.159651 -0.114631
                                                                                                                       0.068734
     SibSp
                                                                                              -0.059528
                                                                                                           -0.026354
     Parch
              -0.001652 0.081629
                                0.018443 -0.182556 0.414838
                                                               1.000000 0.216225 -0.245489
                                                                                              -0.011069
                                                                                                           -0.081228
                                                                                                                       0.060814
              0.269335
                                                                                                                       -0.162184
      Fare
                                 0.131900 0.103236 -0.114631 -0.245489 -0.182333
                                                                                                                       0.119224
  Sex male
              0.042939 -0.543351
                                                                                  1.000000
                                                                                               -0.082853
                                                                                                           -0.074115
Embarked C
              -0.001205
                       0.168240
                                 -0.243292
                                           0.031797
                                                    -0.059528
                                                               -0.011069 0.269335 -0.082853
                                                                                               1.000000
                                                                                                           -0.148258
                                                                                                                       -0.782742
Embarked_Q
                                                               -0.081228
              -0.033606
                        0.003650
                                  0.221009
                                           -0.019970
                                                     -0.026354
                                                                                  -0.074115
                                                                                               -0.148258
                                                                                                            1.000000
                                                                                                                       -0.499421
                       -0.149683
                                 0.074053 -0.015289
                                                     0.068734
                                                               0.060814
                                                                        -0.162184
                                                                                                           -0.499421
                                                                                                                        1.000000
  #prepare training data
  X = df.drop(['Survived', 'Pclass'],axis=1)
  Y = df['Survived']
  #split data to training data and testing data
  from sklearn.model_selection import train_test_split
  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
```

訓練模型(logistic regression)

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, Y_train)
predictions = lr.predict(X_test)

d:\Python\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html

Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result(
```

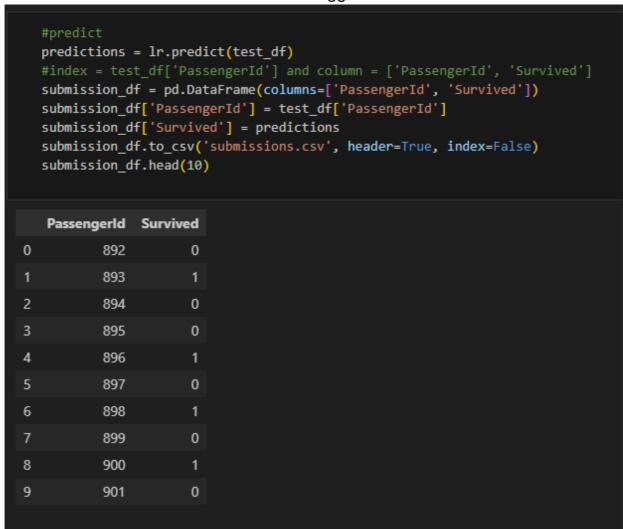
• 讀入測試集資料並且處理缺失值

```
test_df = pd.read_csv('test.csv')
    test_df.head()
                                                                         Sex Age SibSp Parch
    Passengerld Pclass
                                                              Name
                                                                                                        Ticket
                                                                                                                   Fare Cabin Embarked
                                                                       male 34.5
                                                                                                       330911
                                                                                                                 7.8292
                                                                                                                           NaN
                                                                                                                                           Q
                                                     Kelly, Mr. James
                                      Wilkes, Mrs. James (Ellen Needs) female 47.0
                                                                                                       363272
                                                                                                                 7 0000
                                                                                                                           NaN
                                           Myles, Mr. Thomas Francis male 62.0
                                                                                                       240276
                                                                                                                 9.6875
                                                                        male 27.0
             895
                                                     Wirz, Mr. Albert
                                                                                                      315154
                                                                                                                 8.6625
                                                                                                                           NaN
                       3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
   test_df.isnull().sum()
PassengerId
                   0
Pclass
                   0
Name
                   A
Sex
                  86
Age
SibSp
Parch
Ticket
                  0
Fare
Cabin
                 327
Embarked
dtype: int64
   test_df['Age'] = test_df.groupby('Sex')['Age'].apply(lambda x: x.fillna(x.mean()))
test_df['Fare'] = test_df.groupby('Sex')['Fare'].apply(lambda x: x.fillna(x.mean()))
```

• 準備測試用資料(欄位值編碼和缺失值處理並去除不需要的欄位)

```
#prepare test data
test_df['Sex_male'] = np.where(test_df['Sex'] == 'male', 1, 0)
test_df.drop('Sex', axis=1, inplace=True)
test_df['Embarked_C'] = np.where(test_df['Embarked'] == 'C', 1, 0)
test_df['Embarked_Q'] = np.where(test_df['Embarked'] == 'Q', 1, 0)
test_df['Embarked_S'] = np.where(test_df['Embarked'] == 'S', 1, 0)
test_df.drop('Embarked', axis=1, inplace=True)
test_df.head()
                                                                                                                                       Fare Cabin Sex_male Embarked_C Embarked_Q Embarked_S
  Passengerld Pclass
                                                                              Name Age SibSp Parch
                                                                                                                        Ticket
                                                                 Kelly, Mr. James 34.5
                                            Wilkes, Mrs. James (Ellen Needs) 47.0
                                                                                                                                     7 0000
                                                                                                                                                   NaN
                                        Myles, Mr. Thomas Francis 62.0
            894
                                                                                                                0 240276
                                                                                                                                     9.6875
                                                                                                                                                   NaN
            896
                          3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) 22.0 1 1 3101298 12.2875
                                                                                                                                                   NaN
#drop the columns model will not use
test_df.drop(['Name', 'Ticket', 'Cabin', 'Pclass'], axis=1, inplace=True)
test_df.head()
  Passengerld Age SibSp Parch
                                                     Fare Sex_male Embarked_C Embarked_Q Embarked_S
            892 34.5
            893 47.0
                                                   7.0000
            894 62.0
                                                   8.6625
            896 22.0
                                            1 12.2875
```

• 使用訓練出來的模型進行預測並產生上傳到kaggle的資料



• 資料成功上傳到kaggle

