Chapter 10 - Ex 2: Titanic - Pipeline

Xem xét việc một hành khách có sống sót hay không dựa trên bộ dữ liệu titanic (train.csv có 891 mẫu và test.csv có 418 mẫu)

Yêu cầu: Hãy đọc dữ liệu từ các tập tin này, áp dụng Logistic Regression để thực hiện việc xác định một hành khách có sống sót hay không dựa trên những thông tin được cung cấp.

1. Đọc dữ liệu train.csv, tiền xử lý dữ liệu nếu cần
2. Tạo X_train, X_test, y_train, y_test từ dữ liệu ở câu 1 với tỷ lệ dữ liệu test là 0.2
3. Áp dụng thuật toán Logistic Regression: fit model, tìm độ chính xác, đánh giá mô hình bằng kiểm tra underfiting và overfiting?
4. Đọc dữ liệu test.csv. Tiền xử lý dữ liệu như train.csv. Tìm kết quả cho dữ liệu test.
5. Ghi kết quả vào file test_pred.csv
6. Áp dụng Pipeline. Lưu kết quả khi áp dụng Pipeline vào file test_pred.csv (thêm 1 cột kết quả mới)

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import math
```

In [2]:
```
data = pd.read_csv("titanic/train.csv")
```

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [4]:
```
# Nhận xét: theo như thông tin trên, dữ liệu Age bị thiếu
# => tiến hành cập nhật các age bị thiếu bằng mean
# Thông tin Cabin thiếu nhiều thông tin => drop bỏ cột này
# Thông tin Embarked bị thiếu 2 ô => xóa 2 dòng thiếu này
```

In [5]:
```
data.mean()
```

Out[5]:
```
PassengerId    446.000000
Survived         0.383838
Pclass           2.308642
Age             29.699118
SibSp            0.523008
Parch            0.381594
Fare            32.204208
dtype: float64
```

In [6]:
```
# thay nan bằng mean
data = data.fillna(data.mean())
```

In [7]:
```
del data['Cabin']
```

In [8]:
```
data = data.dropna()
```

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  889 non-null    int64
 1   Survived     889 non-null    int64
 2   Pclass       889 non-null    int64
 3   Name         889 non-null    object
 4   Sex          889 non-null    object
 5   Age          889 non-null    float64
 6   SibSp        889 non-null    int64
 7   Parch        889 non-null    int64
 8   Ticket       889 non-null    object
 9   Fare         889 non-null    float64
 10  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 83.3+ KB
```

In [10]:
```
data.describe()
```

Out[10]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000
mean	446.000000	0.382452	2.311586	29.653446	0.524184	0.382452	32.096681
std	256.998173	0.486260	0.834700	12.968366	1.103705	0.806761	49.697504
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	224.000000	0.000000	2.000000	22.000000	0.000000	0.000000	7.895800
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.000000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [11]:
```
data.head()
```

Out[11]:

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

In [12]:
```
df = data[['Survived', 'Pclass', 'Sex', 'Age', 'SibSp',
           'Parch', 'Fare', 'Embarked']]
```

In [13]:
```
df.head()
```

Out[13]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

In [14]:
```
# Categorical boolean mask
categorical_feature_mask = df.dtypes==object
# filter categorical columns using mask and turn it into a list
categorical_cols = df.columns[categorical_feature_mask].tolist()
categorical_cols
```

Out[14]:
```
['Sex', 'Embarked']
```

In [15]:
```
df_now = pd.get_dummies(data=df, columns=categorical_cols,
                        drop_first=True)
```

In [16]:
```
df_now.head()
```

Out[16]:

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	0	3	22.0	1	0	7.2500	1	0	1
1	1	1	38.0	1	0	71.2833	0	0	0
2	1	3	26.0	0	0	7.9250	0	0	1
3	1	1	35.0	1	0	53.1000	0	0	1
4	0	3	35.0	0	0	8.0500	1	0	1

In [17]:
```
df_now.tail()
```

Out[17]:

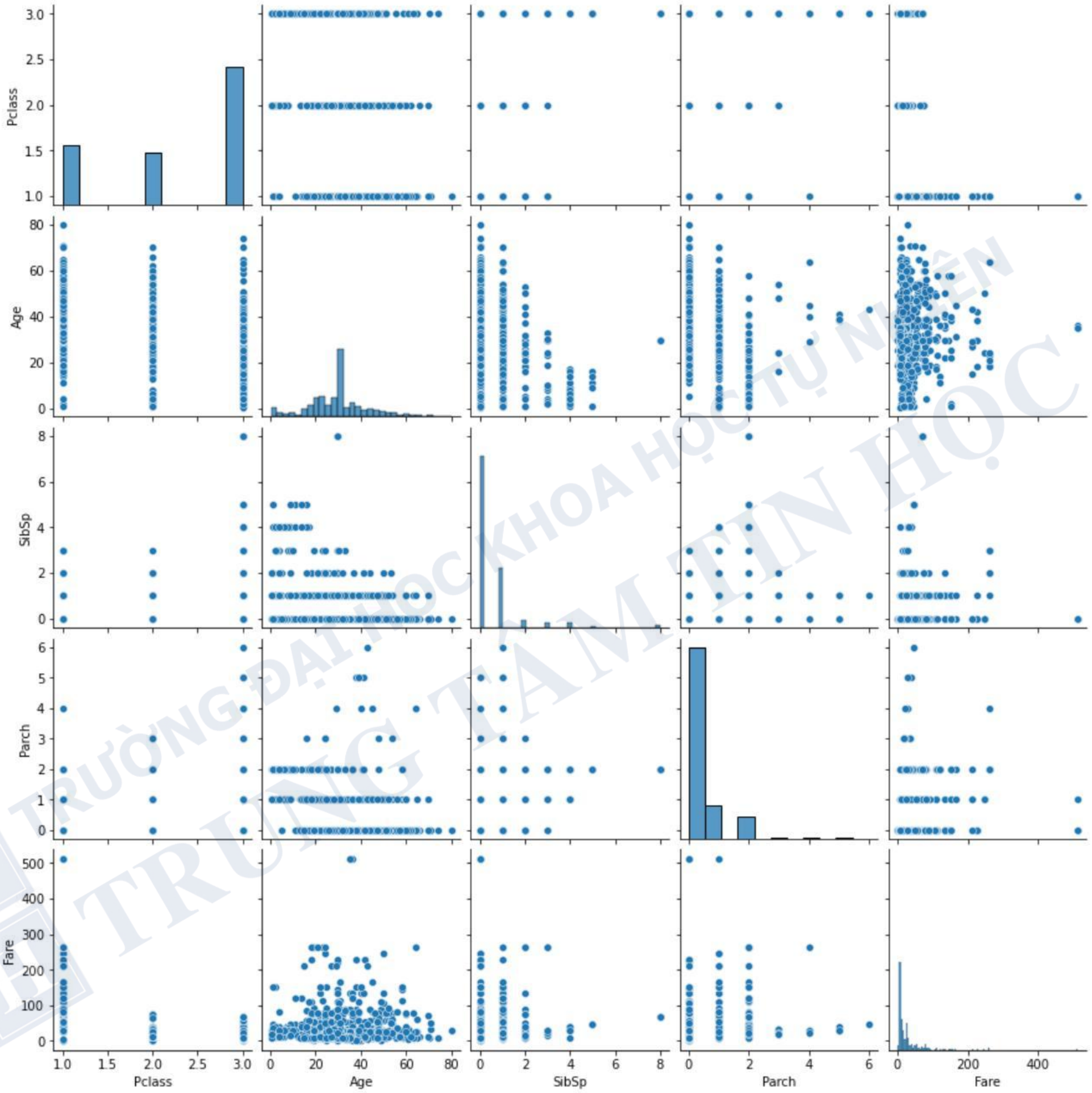
	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
886	0	2	27.000000	0	0	13.00	1	0	1
887	1	1	19.000000	0	0	30.00	0	0	1
888	0	3	29.699118	1	2	23.45	0	0	1
889	1	1	26.000000	0	0	30.00	1	0	0
890	0	3	32.000000	0	0	7.75	1	1	0

In [18]:
```
df_now.isnull().any()
```

Out[18]:
```
Survived      False
Pclass        False
Age           False
SibSp         False
Parch         False
Fare          False
Sex_male      False
Embarked_Q    False
Embarked_S    False
dtype: bool
```

In [19]:
```
import seaborn as sns
```

In [20]:
```
sns.pairplot(data[["Pclass", "Sex", "Age", "SibSp",
                   "Parch", "Fare", "Embarked"]])
plt.show()
```



In [21]:
```
X = df_now.drop('Survived', 1)
X.head()
```

Out[21]:

	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	3	22.0	1	0	7.2500	1	0	1
1	1	38.0	1	0	71.2833	0	0	0
2	3	26.0	0	0	7.9250	0	0	1
3	1	35.0	1	0	53.1000	0	0	1
4	3	35.0	0	0	8.0500	1	0	1

In [22]:
```
y = df_now['Survived']
y.head()
```

Out[22]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

Build & Test model

In [23]:
```
X_train,X_test,y_train,y_test = train_test_split(X,
                                                 y,
                                                 test_size=0.2)
```

In [24]:
```
from sklearn.linear_model import LogisticRegression
```

In [25]:
```
clf = LogisticRegression(solver='liblinear')
```

In [26]:
```
from sklearn.utils.validation import column_or_1d
```

In [27]:
```
clf.fit(X_train, y_train)
```

Out[27]:
```
LogisticRegression(solver='liblinear')
```

In [28]:
```
clf.intercept_
```

Out[28]:
```
array([3.93638966])
```

In [29]:
```
clf.coef_
```

Out[29]:
```
array([[-0.84452883, -0.02890984, -0.29228929, -0.07349696,  0.00419227,
        -2.44393531,  0.03987787, -0.28242532]])
```

In [30]:
```
print('Score train: ', clf.score(X_train, y_train))
print('Score test: ', clf.score(X_test, y_test))
```

```
Score train:  0.8101265822784810
Score test:  0.7921348314606742
```

In [31]:
```
# Mô hình trên có score của train và test gần như nhau
# và khoảng 80%: không bị overfitting và underfiiting
```

In [32]:
```
yhat_train = clf.predict(X_train)
```

In [33]:
```
yhat_test = clf.predict(X_test)
```

In [34]:
```
from sklearn.metrics import accuracy_score
```

In [35]:
```
print("Accuracy:", accuracy_score(y_test,yhat_test)*100,"%")
```

```
Accuracy: 79.21348314606742 %
```

Make prediction on Test data

In [36]:
```
df_test = pd.read_csv("titanic/test.csv")
```

In [37]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

In [38]:
```
# Nhận xét: Theo như thông tin trên, dữ liệu Age bị thiếu
# => tiến hành cập nhật các age bị thiếu bằng mean
# Thông tin Cabin thiếu nhiều thông tin => drop bỏ cột này
# Thông tin Fare bị thiếu 1 mẫu => xóa 1 dòng thiếu này
```

In [39]:
```
df_test.mean()
```

Out[39]:
```
PassengerId    1100.500000
Pclass            2.265550
Age              30.272590
SibSp             0.447368
Parch             0.392344
Fare             35.627188
dtype: float64
```

In [40]:
```
# thay nan bằng mean
df_test = df_test.fillna(df_test.mean())
```

In [41]:
```
del df_test['Cabin']
```

In [42]:
```
df_test = df_test.dropna()
```

In [43]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 418 entries, 0 to 417
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          418 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         418 non-null    float64
 9   Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(4)
memory usage: 35.9+ KB
```

In [44]:
```
df_test.describe()
```

Out[44]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	12.634534	0.896760	0.981429	55.840500
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	23.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	30.272590	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	35.750000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [45]:
```
df_test.head()
```

Out[45]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	S

In [46]:
```
df_test = df_test[['Pclass', 'Sex', 'Age', 'SibSp',
                   'Parch', 'Fare', 'Embarked']]
```

In [47]:
```
df_test.head()
```

Out[47]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	34.5	0	0	7.8292	Q
1	3	female	47.0	1	0	7.0000	S
2	2	male	62.0	0	0	9.6875	Q
3	3	male	27.0	0	0	8.6625	S
4	3	female	22.0	1	1	12.2875	S

In [48]:
```
# Categorical boolean mask
categorical_feature_mask = df_test.dtypes==object
# filter categorical columns using mask and turn it into a list
categorical_cols = df_test.columns[categorical_feature_mask].tolist()
categorical_cols
```

Out[48]:
```
['Sex', 'Embarked']
```

In [49]:
```
df_test_now = pd.get_dummies(data=df_test,
                             columns=categorical_cols,
                             drop_first=True)
```

In [50]:
```
df_test_now.head()
```

Out[50]:

	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	3	34.5	0	0	7.8292	1	1	0
1	3	47.0	1	0	7.0000	0	0	1
2	2	62.0	0	0	9.6875	1	1	0
3	3	27.0	0	0	8.6625	1	0	1
4	3	22.0	1	1	12.2875	0	0	1

In [51]:
```
df_test_now.tail()
```

Out[51]:

	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
413	3	30.27259	0	0	8.0500	1	0	1
414	1	39.00000	0	0	108.9000	0	0	0
415	3	38.50000	0	0	7.2500	1	0	1
416	3	30.27259	0	0	8.0500	1	0	1
417	3	30.27259	1	1	22.3583	1	0	0

In [52]:
```
df_test_now.isnull().any()
```

Out[52]:
```
Pclass        False
Age           False
SibSp         False
Parch         False
Fare          False
Sex_male      False
Embarked_Q    False
Embarked_S    False
dtype: bool
```

In [53]:
```
X_test_now = df_test_now
X_test_now.head()
```

Out[53]:

	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	3	34.5	0	0	7.8292	1	1	0
1	3	47.0	1	0	7.0000	0	0	1
2	2	62.0	0	0	9.6875	1	1	0
3	3	27.0	0	0	8.6625	1	0	1
4	3	22.0	1	1	12.2875	0	0	1

In [54]:
```
Yhat_test_now = clf.predict(X_test_now)
```

In [55]:
```
df_test_now['Survived'] = Yhat_test_now
```

In [56]:
```
df_test_now.head()
```

Out[56]:

	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S	Survived
0	3	34.5	0	0	7.8292	1	1	0	0
1	3	47.0	1	0	7.0000	0	0	1	0
2	2	62.0	0	0	9.6875	1	1	0	0
3	3	27.0	0	0	8.6625	1	0	1	0
4	3	22.0	1	1	12.2875	0	0	1	1

In [57]:
```
df_test['Survived'] = Yhat_test_now
```

In [58]:
```
df_test.head()
```

Out[58]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Survived
0	3	male	34.5	0	0	7.8292	Q	0
1	3	female	47.0	1	0	7.0000	S	0
2	2	male	62.0	0	0	9.6875	Q	0
3	3	male	27.0	0	0	8.6625	S	0
4	3	female	22.0	1	1	12.2875	S	1

In [59]:
```
df_test.to_csv('titanic/test_pred.csv')
```

Pipeline

In [60]:
```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
```

In [61]:
```
X = df.drop('Survived', 1)
X.head()
```

Out[61]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.0	1	0	7.2500	S
1	1	female	38.0	1	0	71.2833	C
2	3	female	26.0	0	0	7.9250	S
3	1	female	35.0	1	0	53.1000	S
4	3	male	35.0	0	0	8.0500	S

In [62]:
```
y = df['Survived']
y.head()
```

Out[62]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [63]:
```
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.2)
```

In [64]:
```
Input=[('column_tr', make_column_transformer((OneHotEncoder(),
                                              ['Sex', 'Embarked']),
                                             remainder='passthrough')),
       ('model', LogisticRegression(solver='liblinear'))]
```

In [65]:
```
pipe = Pipeline(Input)
```

In [66]:
```
pipe.fit(X_train, y_train)
```

Out[66]:
```
Pipeline(steps=[('column_tr',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(),
                                                  ['Sex', 'Embarked'])])),
                ('model', LogisticRegression(solver='liblinear'))])
```

In [67]:
```
pipe.predict(X_test)
```

Out[67]:
```
array([0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
       0, 0], dtype=int64)
```

In [68]:
```
pipe.score(X_train, y_train)
```

Out[68]:
```
0.8016877637130801
```

In [69]:
```
pipe.score(X_test, y_test)
```

Out[69]:
```
0.7865168539325843
```

In [70]:
```
X_test.head()
```

Out[70]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
349	3	male	42.000000	0	0	8.6625	S
556	1	female	48.000000	1	0	39.6000	C
852	3	female	9.000000	1	1	15.2458	C
407	2	male	3.000000	1	1	18.7500	S
475	1	male	29.699118	0	0	52.0000	S

In [71]:
```
data_pipe = df_test[['Pclass', 'Sex', 'Age', 'SibSp',
                     'Parch', 'Fare', 'Embarked']]
```

In [72]:
```
y_new = pipe.predict(data_pipe)
```

In [73]:
```
# Lưu cả kết quả với Pipeline vào file (với 1 cột mới)
```

In [74]:
```
df_test['Survived_pipe'] = y_new
```

In [76]:
```
df_test.head()
```

Out[76]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Survived	Survived_pipe
0	3	male	34.5	0	0	7.8292	Q	0	0
1	3	female	47.0	1	0	7.0000	S	0	0
2	2	male	62.0	0	0	9.6875	Q	0	0
3	3	male	27.0	0	0	8.6625	S	0	0
4	3	female	22.0	1	1	12.2875	S	1	1

In [75]:
```
df_test.to_csv('titanic/test_pred.csv')
```

In [ ]:
```

```