Titanic Classification

September 13, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
titanic_data = pd.read_csv('C:/Users/ASUS/Downloads/Titanic/train.csv')
```

[3]:
```
titanic_data
```

[3]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name     Sex   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                                Heikkinen, Miss. Laina  female  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                              Allen, Mr. William Henry    male  35.0      0
..                                                  ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
```

```
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[4]:
```
titanic_data.head()
```

[4]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

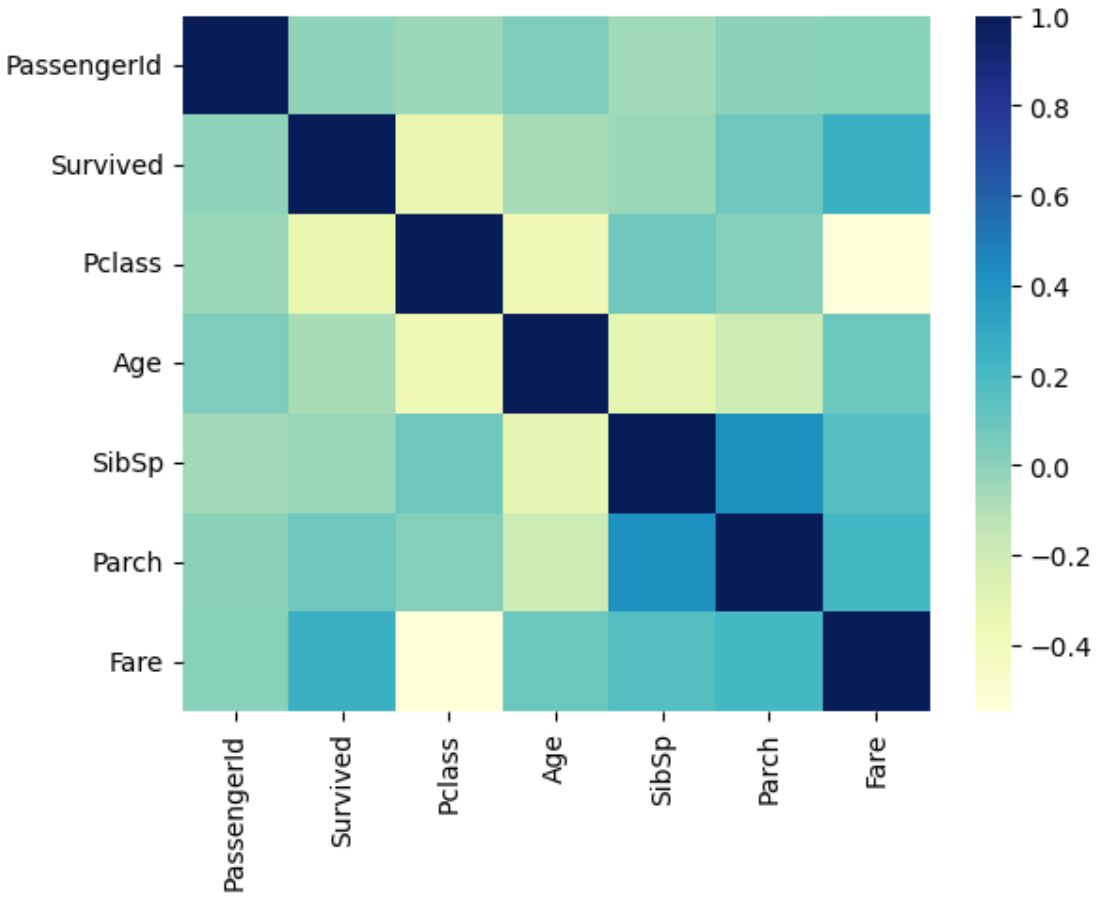
[5]:
```
titanic_data.describe()
```

[5]:
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
```

```
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

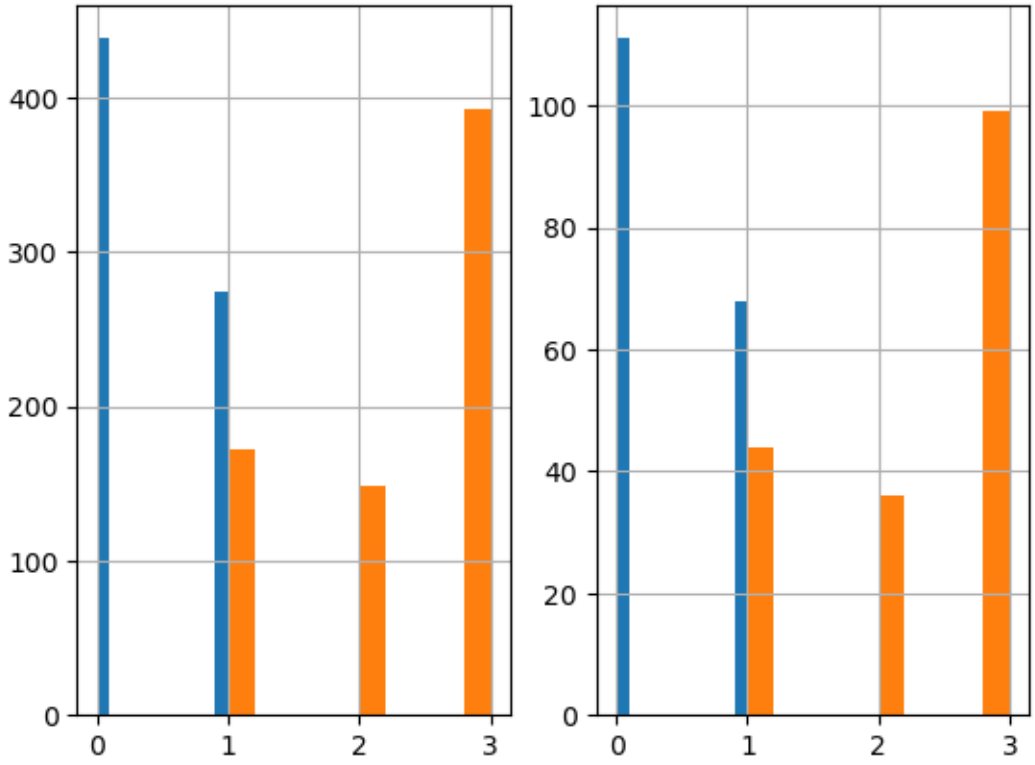
[6]:
```
import seaborn as sns

sns.heatmap(titanic_data.corr(), cmap="YlGnBu")
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_26076\1602845089.py:3: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(titanic_data.corr(), cmap="YlGnBu")
```



[7]:
```
from sklearn.model_selection import StratifiedShuffleSplit

split = StratifiedShuffleSplit(n_splits=1, test_size=0.2)
for train_indices, test_indices in split.split(titanic_data,␣
 ↪titanic_data[["Survived", "Pclass", "Sex"]]):
    strat_train_set = titanic_data.loc[train_indices]
    strat_test_set = titanic_data.loc[test_indices]
```

[8]:
```
plt.subplot(1,2,1)
strat_train_set['Survived'].hist()
strat_train_set['Pclass'].hist()

plt.subplot(1,2,2)
strat_test_set['Survived'].hist()
strat_test_set['Pclass'].hist()

plt.show()
```



[9]:
```
strat_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 362 to 16
```

```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Pclass       712 non-null    int64
 3   Name         712 non-null    object
 4   Sex          712 non-null    object
 5   Age          573 non-null    float64
 6   SibSp        712 non-null    int64
 7   Parch        712 non-null    int64
 8   Ticket       712 non-null    object
 9   Fare         712 non-null    float64
 10  Cabin        157 non-null    object
 11  Embarked     711 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 72.3+ KB
```

[10]:
```python
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.impute import SimpleImputer

class AgeImputer(BaseEstimator, TransformerMixin):

    def fit(self, x, y=None):
        return self

    def transform(self, X):
        imputer = SimpleImputer(strategy="mean")
        X['Age']=imputer.fit_transform(X[["Age"]])
        return X
```

[11]:
```python
from sklearn.preprocessing import OneHotEncoder

class FeatureEncoder(BaseEstimator, TransformerMixin):

    def fit(self, X):
        return self

    def transform(self, X):
        encoder = OneHotEncoder()
        matrix = encoder.fit_transform(X[['Embarked']]).toarray()

        column_names = ["C", "S", "Q", "N"]

        for i in range(len(matrix.T)):
            X[column_names[i]] = matrix.T[i]
```

```
        matrix = encoder.fit_transform(X[['Sex']]).toarray()

        column_names = ["Female", "Male"]

        for i in range(len(matrix.T)):
            X[column_names[i]] = matrix.T[i]

        return X
```

[12]:
```
class FeatureDropper(BaseEstimator, TransformerMixin):

    def fit(self, X, y = None):
        return self

    def transform(self, X):
        return X.drop(["Embarked", "Name", "Ticket", "Cabin", "Sex", "N"], axis␣
↪= 1, errors="ignore")
```

[13]:
```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([("ageimputer", AgeImputer()),
                     ("featureencoder", FeatureEncoder()),
                     ("featuredropper", FeatureDropper())])
```

[14]:
```
strat_train_set = pipeline.fit_transform(strat_train_set)
```

[15]:
```
strat_train_set
```

[15]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	C	\
362	363	0	3	45.000000	0	1	14.4542	1.0	
532	533	0	3	17.000000	1	1	7.2292	1.0	
699	700	0	3	42.000000	0	0	7.6500	0.0	
160	161	0	3	44.000000	0	1	16.1000	0.0	
828	829	1	3	29.585951	0	0	7.7500	0.0	
..	...	...	...	...	...	...	...	...	
507	508	1	1	29.585951	0	0	26.5500	0.0	
805	806	0	3	31.000000	0	0	7.7750	0.0	
181	182	0	2	29.585951	0	0	15.0500	1.0	
737	738	1	1	35.000000	0	0	512.3292	1.0	
16	17	0	3	2.000000	4	1	29.1250	0.0	

	S	Q	Female	Male
362	0.0	0.0	1.0	0.0
532	0.0	0.0	0.0	1.0
699	0.0	1.0	0.0	1.0
160	0.0	1.0	0.0	1.0
828	1.0	0.0	0.0	1.0

```
..   ...  ...     ...   ...
507  0.0  1.0     0.0   1.0
805  0.0  1.0     0.0   1.0
181  0.0  0.0     0.0   1.0
737  0.0  0.0     0.0   1.0
16   1.0  0.0     0.0   1.0

[712 rows x 12 columns]
```

[17]:
```
strat_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 362 to 16
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Pclass       712 non-null    int64
 3   Age          712 non-null    float64
 4   SibSp        712 non-null    int64
 5   Parch        712 non-null    int64
 6   Fare         712 non-null    float64
 7   C            712 non-null    float64
 8   S            712 non-null    float64
 9   Q            712 non-null    float64
 10  Female       712 non-null    float64
 11  Male         712 non-null    float64
dtypes: float64(7), int64(5)
memory usage: 72.3 KB
```

[21]:
```
from sklearn.preprocessing import StandardScaler

X = strat_train_set.drop(['Survived'], axis=1)
Y = strat_train_set['Survived']

scaler = StandardScaler()
X_data = scaler.fit_transform(X)
Y_data = Y.to_numpy()
```

[22]:
```
X_data
```

[22]:
```
array([[-3.45543672e-01,  8.27893418e-01,  1.18806281e+00, ...,
        -1.63985340e+00,  1.35941164e+00, -1.35941164e+00],
       [ 3.23200629e-01,  8.27893418e-01, -9.70082596e-01, ...,
        -1.63985340e+00, -7.35612358e-01,  7.35612358e-01],
       [ 9.80143559e-01,  8.27893418e-01,  9.56832949e-01, ...,
         6.09810609e-01, -7.35612358e-01,  7.35612358e-01],
```

```
       ...,
       [-1.05755966e+00, -3.70196244e-01,  2.73831169e-16, ...,
        -1.63985340e+00, -7.35612358e-01,  7.35612358e-01],
       [ 1.12962758e+00, -1.56828591e+00,  4.17296597e-01, ...,
        -1.63985340e+00, -7.35612358e-01,  7.35612358e-01],
       [-1.70663501e+00,  8.27893418e-01, -2.12623192e+00, ...,
        -1.63985340e+00, -7.35612358e-01,  7.35612358e-01]])
```

[23]:
```
Y_data
```

[23]:
```
array([0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1,
       1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1,
       1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0], dtype=int64)
```

[27]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
clf = RandomForestClassifier()

param_gird = [
    {"n_estimators": [10,100,200,500], "max_depth": [None,5,10],␣
 ↪"min_samples_split": [2,3,4]}
]

grid_search = GridSearchCV(clf, param_gird, cv=3, scoring="accuracy",␣
 ↪return_train_score=True)
grid_search.fit(X_data, Y_data)
```

[27]:
```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [None, 5, 10],
                          'min_samples_split': [2, 3, 4],
                          'n_estimators': [10, 100, 200, 500]}],
             return_train_score=True, scoring='accuracy')
```

[28]:
```
final_clf = grid_search.best_estimator_
```

[29]:
```
final_clf
```

[29]:
```
RandomForestClassifier(max_depth=5, min_samples_split=4)
```

[30]:
```
strat_test_set = pipeline.fit_transform(strat_test_set)
```

[31]:
```
strat_test_set
```

[31]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	C	\
769	770	0	3	32.000000	0	0	8.3625	0.0	
857	858	1	1	51.000000	0	0	26.5500	0.0	
63	64	0	3	4.000000	3	2	27.9000	0.0	
872	873	0	1	33.000000	0	0	5.0000	0.0	
15	16	1	2	55.000000	0	0	16.0000	0.0	
..	...	...	...	...	...	...	...	...	
880	881	1	2	25.000000	0	1	26.0000	0.0	
716	717	1	1	38.000000	0	0	227.5250	1.0	
711	712	0	1	30.159007	0	0	26.5500	0.0	
326	327	0	3	61.000000	0	0	6.2375	0.0	
462	463	0	1	47.000000	0	0	38.5000	0.0	

	S	Q	Female	Male
769	0.0	1.0	0.0	1.0
857	0.0	1.0	0.0	1.0
63	0.0	1.0	0.0	1.0
872	0.0	1.0	0.0	1.0
15	0.0	1.0	1.0	0.0
..	...	...	...	...

```
880  0.0  1.0     1.0   0.0
716  0.0  0.0     1.0   0.0
711  0.0  1.0     0.0   1.0
326  0.0  1.0     0.0   1.0
462  0.0  1.0     0.0   1.0

[179 rows x 12 columns]
```

[35]:
```
X_test = strat_test_set.drop(['Survived'], axis=1)
Y_test = strat_test_set['Survived']

scaler = StandardScaler()
X_data_test = scaler.fit_transform(X_test)
Y_data_test = Y_test.to_numpy()
```

[36]:
```
final_clf.score(X_data_test, Y_data_test)
```

[36]:
```
0.8156424581005587
```

[37]:
```
final_data = pipeline.fit_transform(titanic_data)
```

[38]:
```
final_data
```

[38]:
```
     PassengerId  Survived  Pclass        Age  SibSp  Parch     Fare    C  \
0              1         0       3  22.000000      1      0   7.2500  0.0
1              2         1       1  38.000000      1      0  71.2833  1.0
2              3         1       3  26.000000      0      0   7.9250  0.0
3              4         1       1  35.000000      1      0  53.1000  0.0
4              5         0       3  35.000000      0      0   8.0500  0.0
..           ...       ...     ...        ...    ...    ...      ...  ...
886          887         0       2  27.000000      0      0  13.0000  0.0
887          888         1       1  19.000000      0      0  30.0000  0.0
888          889         0       3  29.699118      1      2  23.4500  0.0
889          890         1       1  26.000000      0      0  30.0000  1.0
890          891         0       3  32.000000      0      0   7.7500  0.0

       S    Q  Female  Male
0    0.0  1.0     0.0   1.0
1    0.0  0.0     1.0   0.0
2    0.0  1.0     1.0   0.0
3    0.0  1.0     1.0   0.0
4    0.0  1.0     0.0   1.0
..   ...  ...     ...   ...
886  0.0  1.0     0.0   1.0
887  0.0  1.0     1.0   0.0
888  0.0  1.0     1.0   0.0
889  0.0  0.0     0.0   1.0
```

```
890  1.0  0.0     0.0   1.0

[891 rows x 12 columns]
```

[41]:
```
X_final = final_data.drop(['Survived'], axis=1)
Y_final = final_data['Survived']

scaler = StandardScaler()
X_data_final = scaler.fit_transform(X_final)
Y_data_final = Y_final.to_numpy()
```

[42]:
```
prod_clf = RandomForestClassifier()


param_gird = [
    {"n_estimators": [10,100,200,500], "max_depth": [None,5,10],␣
 ↪"min_samples_split": [2,3,4]}
]

grid_search = GridSearchCV(prod_clf, param_gird, cv=3, scoring="accuracy",␣
 ↪return_train_score=True)
grid_search.fit(X_data_final, Y_data_final)
```

[42]:
```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [None, 5, 10],
                          'min_samples_split': [2, 3, 4],
                          'n_estimators': [10, 100, 200, 500]}],
             return_train_score=True, scoring='accuracy')
```

[43]:
```
prod_final_clf = grid_search.best_estimator_
```

[45]:
```
prod_final_clf
```

[45]:
```
RandomForestClassifier(max_depth=5, min_samples_split=3, n_estimators=200)
```

[46]:
```
titanic_test_data = pd.read_csv('C:/Users/ASUS/Downloads/Titanic/test.csv')
```

[48]:
```
titanic_test_data
```

[48]:
```
     PassengerId  Pclass                                          Name  \
0            892       3                              Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                              Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..           ...     ...                                           ...
413         1305       3                           Spector, Mr. Woolf
```

```
414         1306       1                 Oliva y Ocana, Dona. Fermina
415         1307       3                 Saether, Mr. Simon Sivertsen
416         1308       3                          Ware, Mr. Frederick
417         1309       3                     Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embarked
0      male  34.5      0      0              330911    7.8292   NaN        Q
1    female  47.0      1      0              363272    7.0000   NaN        S
2      male  62.0      0      0              240276    9.6875   NaN        Q
3      male  27.0      0      0              315154    8.6625   NaN        S
4    female  22.0      1      1             3101298   12.2875   NaN        S
..      ...   ...    ...    ...                 ...       ...   ...      ...
413    male   NaN      0      0           A.5. 3236    8.0500   NaN        S
414  female  39.0      0      0            PC 17758  108.9000  C105        C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416    male   NaN      0      0              359309    8.0500   NaN        S
417    male   NaN      1      1                2668   22.3583   NaN        C

[418 rows x 11 columns]
```

[49]:
```
final_test_data = pipeline.fit_transform(titanic_test_data)
```

[50]:
```
final_test_data
```

[50]:
```
     PassengerId  Pclass       Age  SibSp  Parch      Fare    C    S    Q  \
0            892       3  34.50000      0      0    7.8292  0.0  1.0  0.0
1            893       3  47.00000      1      0    7.0000  0.0  0.0  1.0
2            894       2  62.00000      0      0    9.6875  0.0  1.0  0.0
3            895       3  27.00000      0      0    8.6625  0.0  0.0  1.0
4            896       3  22.00000      1      1   12.2875  0.0  0.0  1.0
..           ...     ...       ...    ...    ...       ...  ...  ...  ...
413         1305       3  30.27259      0      0    8.0500  0.0  0.0  1.0
414         1306       1  39.00000      0      0  108.9000  1.0  0.0  0.0
415         1307       3  38.50000      0      0    7.2500  0.0  0.0  1.0
416         1308       3  30.27259      0      0    8.0500  0.0  0.0  1.0
417         1309       3  30.27259      1      1   22.3583  1.0  0.0  0.0

     Female  Male
0       0.0   1.0
1       1.0   0.0
2       0.0   1.0
3       0.0   1.0
4       1.0   0.0
..      ...   ...
413     0.0   1.0
414     1.0   0.0
415     0.0   1.0
```

```
416     0.0   1.0
417     0.0   1.0

[418 rows x 11 columns]
```

[51]:
```
final_test_data.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   Age          418 non-null    float64
 3   SibSp        418 non-null    int64
 4   Parch        418 non-null    int64
 5   Fare         417 non-null    float64
 6   C            418 non-null    float64
 7   S            418 non-null    float64
 8   Q            418 non-null    float64
 9   Female       418 non-null    float64
 10  Male         418 non-null    float64
dtypes: float64(7), int64(4)
memory usage: 36.1 KB
```

[55]:
```
X_final_test = final_test_data
X_final_test = X_final_test.fillna(method="ffill")

scaler = StandardScaler()
X_data_final_test = scaler.fit_transform(X_final_test)
```

[56]:
```
predictions = prod_final_clf.predict(X_data_final_test)
```

[57]:
```
predictions
```

[57]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
```

```
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0],
      dtype=int64)
```

[58]:
```
final_df = pd.DataFrame(titanic_test_data['PassengerId'])
final_df['Survived'] = predictions
final_df.to_csv('C:/Users/ASUS/Downloads/Titanic/predictions.csv', index=False)
```

[59]:
```
final_df
```

[59]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	1
..	...	...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

[418 rows x 2 columns]