9cljsneew

February 9, 2025

```
[ ]: import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import accuracy_score, classification_report,␣
      ↪confusion_matrix
     from sklearn.preprocessing import LabelEncoder
```

```
[ ]: titanic_data = pd.read_csv('/content/Titanic-Dataset.csv')
```

```
[ ]: titanic_data
```

```
[ ]:      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]
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7a836c985710>
```

```
from matplotlib import pyplot as plt
titanic_data['PassengerId'].plot(kind='hist', bins=20, title='PassengerId')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Survived'].plot(kind='hist', bins=20, title='Survived')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Pclass'].plot(kind='hist', bins=20, title='Pclass')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Sex'].plot(kind='hist', bins=20, title='Sex')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7a836d533510>
```

```
from matplotlib import pyplot as plt
titanic_data.plot(kind='scatter', x='PassengerId', y='Survived', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data.plot(kind='scatter', x='Survived', y='Pclass', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data.plot(kind='scatter', x='Pclass', y='Sex', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data.plot(kind='scatter', x='Sex', y='Age', s=32, alpha=.8)
```

```
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7a836d530510>
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['PassengerId']
  ys = series['Survived']

  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = titanic_data.sort_values('PassengerId', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('PassengerId')
_ = plt.ylabel('Survived')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['PassengerId']
  ys = series['Pclass']

  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = titanic_data.sort_values('PassengerId', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('PassengerId')
_ = plt.ylabel('Pclass')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['PassengerId']
  ys = series['Sex']

  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = titanic_data.sort_values('PassengerId', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('PassengerId')
```

```
_ = plt.ylabel('Sex')
```

```
from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['PassengerId']
  ys = series['Age']

  plt.plot(xs, ys, label=series_name, color=palette[series_index % len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = titanic_data.sort_values('PassengerId', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('PassengerId')
_ = plt.ylabel('Age')
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7a836d5338d0>
```

```
from matplotlib import pyplot as plt
titanic_data['PassengerId'].plot(kind='line', figsize=(8, 4),␣
 ↪title='PassengerId')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Survived'].plot(kind='line', figsize=(8, 4), title='Survived')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Pclass'].plot(kind='line', figsize=(8, 4), title='Pclass')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
titanic_data['Sex'].plot(kind='line', figsize=(8, 4), title='Sex')
plt.gca().spines[['top', 'right']].set_visible(False)
```

[ ]:

```
data.describe()
```

[ ]:

```
              Year       Rating
count  7919.000000  7919.000000
mean   1993.321758     5.841621
std      20.463770     1.381777
min    1917.000000     1.100000
25%    1979.500000     4.900000
50%    1997.000000     6.000000
75%    2011.000000     6.800000
max    2021.000000    10.000000
```

[ ]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7919 entries, 1 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      7919 non-null   object
 1   Year      7919 non-null   float64
 2   Duration  5851 non-null   object
 3   Genre     7817 non-null   object
 4   Rating    7919 non-null   float64
 5   Votes     7919 non-null   object
 6   Director  7914 non-null   object
 7   Actor 1   7794 non-null   object
 8   Actor 2   7719 non-null   object
 9   Actor 3   7627 non-null   object
dtypes: float64(2), object(8)
memory usage: 680.5+ KB
```

[ ]:

```
titanic_data['Age'].fillna(titanic_data['Age'].median(), inplace=True)
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0],␣
 ↪inplace=True)
```

```
<ipython-input-38-4c0e5003b5da>:1: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  titanic_data['Age'].fillna(titanic_data['Age'].median(), inplace=True)
<ipython-input-38-4c0e5003b5da>:2: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0],
inplace=True)
```

[ ]:

```
label_enc = LabelEncoder()
titanic_data['Sex'] = label_enc.fit_transform(titanic_data['Sex'])
titanic_data['Embarked'] = label_enc.fit_transform(titanic_data['Embarked'])
```

[ ]:

```
features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
X = titanic_data[features]
y = titanic_data['Survived']
```

[ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[ ]:

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

[ ]:

```
LogisticRegression(max_iter=1000)
```

[ ]:

```
y_pred = model.predict(X_test)
```

[ ]:

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.8100558659217877

Classification Report:
               precision    recall  f1-score   support

           0       0.83      0.86      0.84       105
           1       0.79      0.74      0.76        74

    accuracy                           0.81       179
   macro avg       0.81      0.80      0.80       179
weighted avg       0.81      0.81      0.81       179


Confusion Matrix:
 [[90 15]
 [19 55]]
```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

[ ]:

```
```