titanic-survival-prediction

March 17, 2024

Import necessary libs

[5]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Data Collection and Processing

[9]:
```python
df=pd.read_csv('train.csv')
df.head()
```

[9]:
```
   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
```

[11]:
```python
df.shape
```

```
[11]: (891, 12)
```

```
[12]: df.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
```

```
[13]: df.isnull().sum()
```

```
[13]: PassengerId      0
      Survived         0
      Pclass           0
      Name             0
      Sex              0
      Age            177
      SibSp            0
      Parch            0
      Ticket           0
      Fare             0
      Cabin          687
      Embarked         2
      dtype: int64
```

```
[14]: #remove missing/null values
      df = df.drop(columns='Cabin', axis=1)
```

```
[15]: #replacing missing values with mean number
      df['Age'].fillna(df['Age'].mean(), inplace=True)
```

```
[16]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 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  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

[17]:

```python
df.isnull().sum()
```

[17]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       2
dtype: int64
```

[18]:

```python
#lets fix Embarked
print(df['Embarked'].mode())
```

```
0    S
Name: Embarked, dtype: object
```

[19]:

```python
print(df['Embarked'].mode()[0])
```

```
S
```

[21]:

```python
#replace the mode value with the missing value
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

[24]:

```python
df.isnull().sum()
```

[24]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Analysing the data

[25]:

```
df.describe()
```

[25]:

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  891.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   13.002015    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   22.000000    0.000000
50%     446.000000    0.000000    3.000000   29.699118    0.000000
75%     668.500000    1.000000    3.000000   35.000000    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
```

[26]:

```
#how many survived?
df['Survived'].value_counts()
```

[26]:

```
0    549
1    342
Name: Survived, dtype: int64
```

[27]:

```
#visualizing data
sns.set()
```

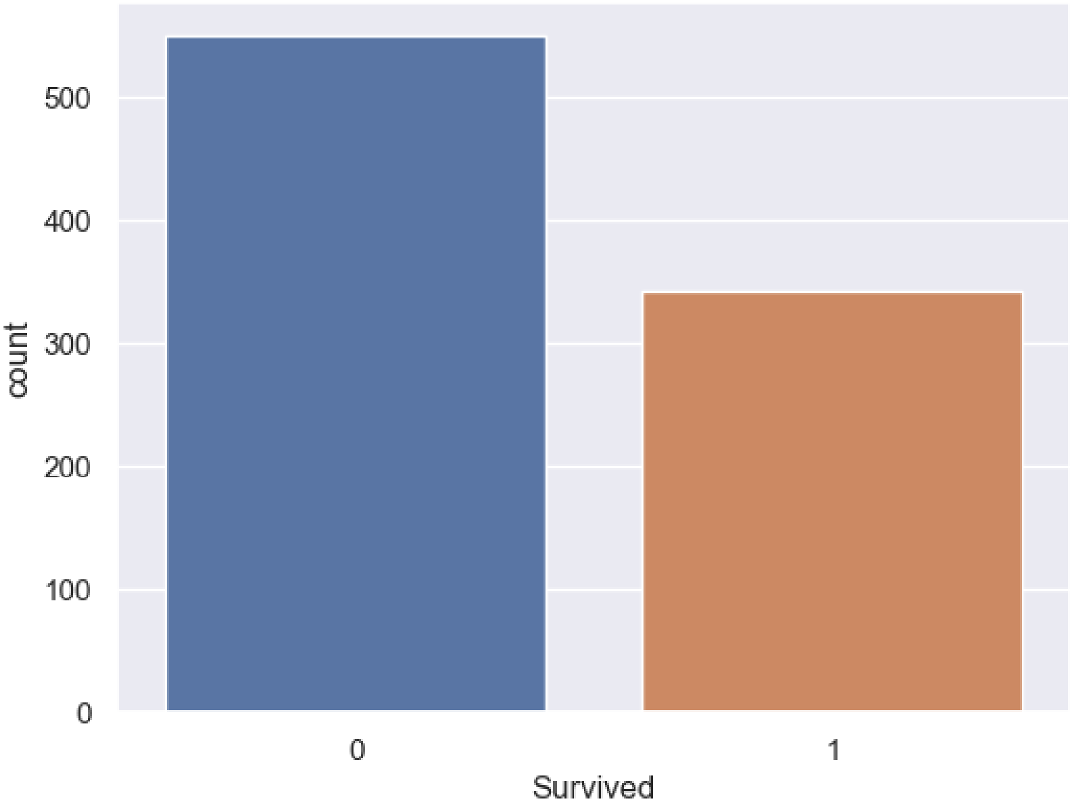
[28]:

```
sns.countplot(df['Survived'])
```

```
C:\Users\CHANDRA ADITYA\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[28]:

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



[29]:

```
df['Sex'].value_counts()
```

[29]:

```
male      577
female    314
Name: Sex, dtype: int64
```

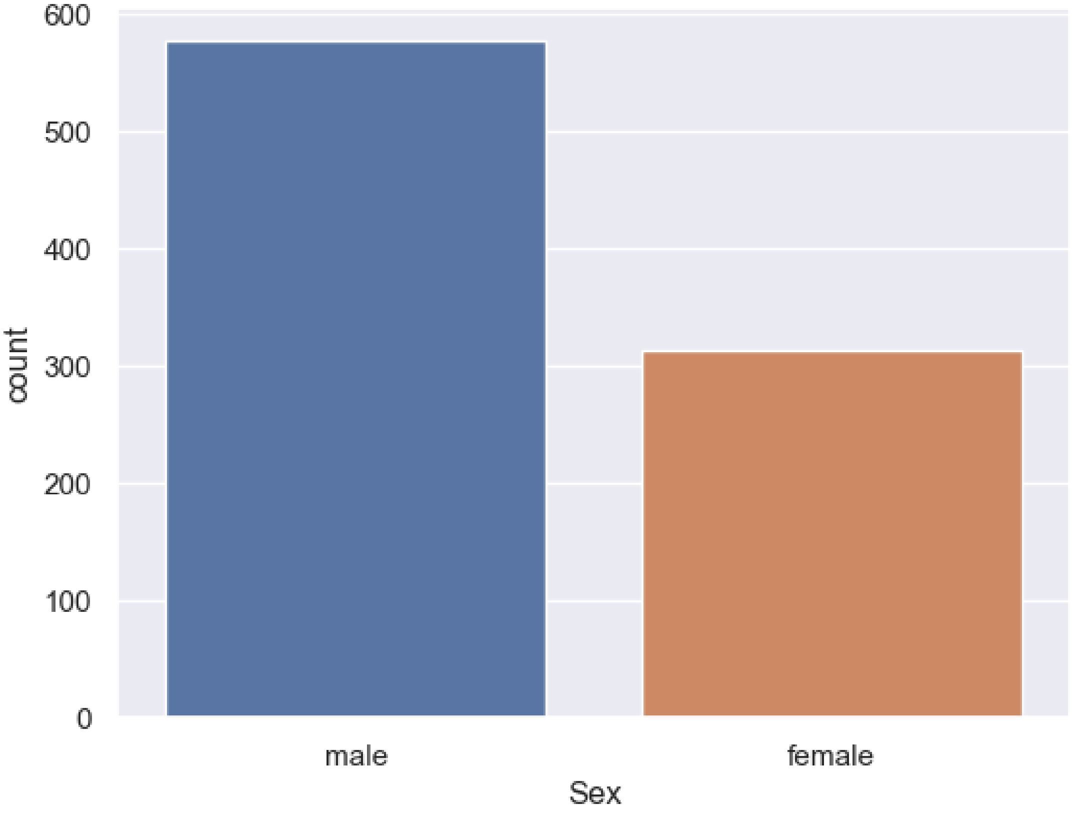
[30]:

```
#count plot for sex column
sns.countplot(df['Sex'])
```

```
C:\Users\CHANDRA ADITYA\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
```

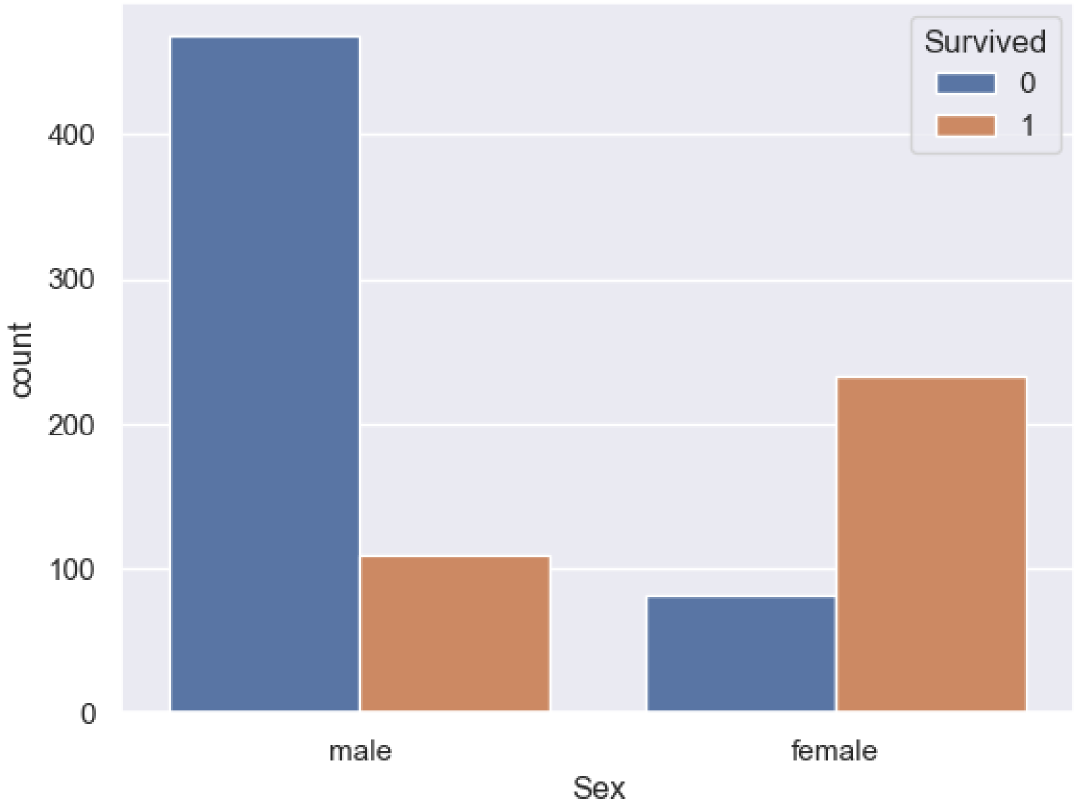
```
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[30]:
```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



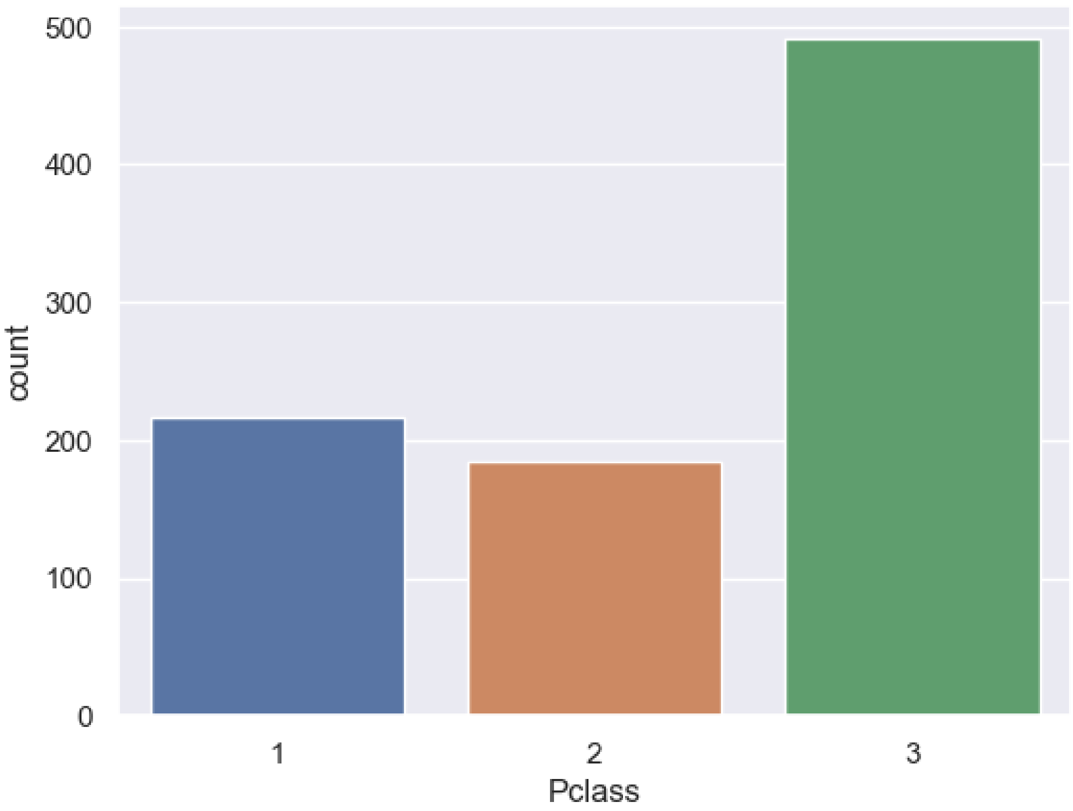
[31]:
```
#Analysing gender wise survivours
sns.countplot(x='Sex', hue='Survived', data=df)
```

[31]:
```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



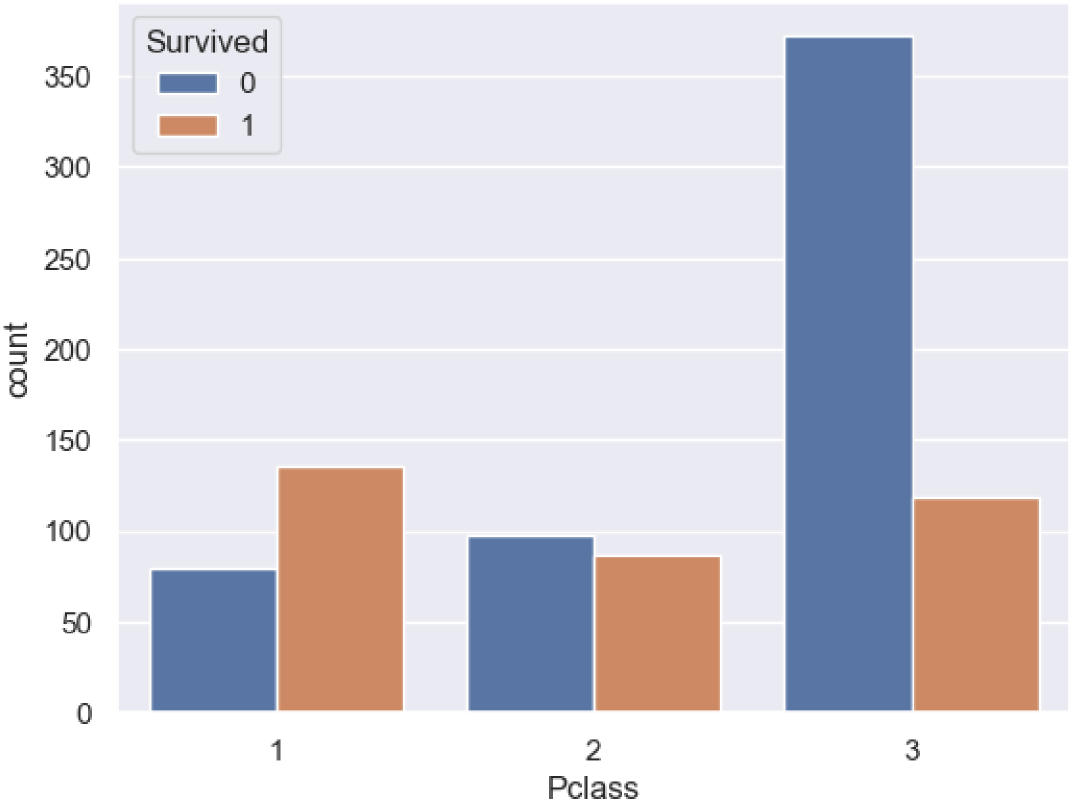
```
[32]: #count plot for "Pclass" column
      sns.countplot(x='Pclass', data=df)
```

```
[32]: <AxesSubplot:xlabel='Pclass', ylabel='count'>
```



```
[33]: sns.countplot(x='Pclass', hue='Survived', data=df)
```

```
[33]: <AxesSubplot:xlabel='Pclass', ylabel='count'>
```



Encode categorical columns/data

[34]:
```
df['Sex'].value_counts()
```

[34]:
```
male      577
female    314
Name: Sex, dtype: int64
```

[35]:
```
df['Embarked'].value_counts()
```

[35]:
```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

[37]:
```
df.replace({'Sex':{'male':0, 'female':1}, 'Embarked':{'S':0, 'C':1, 'Q':2}})
```

[37]:
```
     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    0  22.000000      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…    1  38.000000      1
2                               Heikkinen, Miss. Laina    1  26.000000      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.000000      1
4                             Allen, Mr. William Henry    0  35.000000      0
..                                                  …  …        …    …
886                              Montvila, Rev. Juozas    0  27.000000      0
887                       Graham, Miss. Margaret Edith    1  19.000000      0
888           Johnston, Miss. Catherine Helen "Carrie"    1  29.699118      1
889                              Behr, Mr. Karl Howell    0  26.000000      0
890                                Dooley, Mr. Patrick    0  32.000000      0

     Parch            Ticket     Fare  Embarked
0        0         A/5 21171   7.2500         0
1        0          PC 17599  71.2833         1
2        0  STON/O2. 3101282   7.9250         0
3        0            113803  53.1000         0
4        0            373450   8.0500         0
..     …                …       …        …
886      0            211536  13.0000         0
887      0            112053  30.0000         0
888      2        W./C. 6607  23.4500         0
889      0            111369  30.0000         1
890      0            370376   7.7500         2

[891 rows x 11 columns]
```

```
[40]: X = df.drop(columns = ['PassengerId', 'Name', 'Ticket', 'Survived'], axis=1)
      Y = df['Survived']
```

```
[41]: print(X)
```

```
    Pclass     Sex        Age  SibSp  Parch     Fare Embarked
0         3    male  22.000000      1      0   7.2500        S
1         1  female  38.000000      1      0  71.2833        C
2         3  female  26.000000      0      0   7.9250        S
3         1  female  35.000000      1      0  53.1000        S
```

```
4          3    male  35.000000      0      0   8.0500        S
..       …      …        …    …    …       …       …
886        2    male  27.000000      0      0  13.0000        S
887        1  female  19.000000      0      0  30.0000        S
888        3  female  29.699118      1      2  23.4500        S
889        1    male  26.000000      0      0  30.0000        C
890        3    male  32.000000      0      0   7.7500        Q

[891 rows x 7 columns]
```

[42]:

```
print(Y)
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```