assignment-3

September 19, 2023

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```
df=pd.read_csv("Titanic-Dataset.csv")
```

[5]:
```
df.head()
```

[5]:
```
   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
```

[6]:
```
df.shape
```

[6]:
```
(891, 12)
```

[7]:
```
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
```

[8]:
```
df.describe()
```

[8]:
```
       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
```

[9]:
```
df.isnull().any()
```

[9]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
```

```
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

[10]:
```
df.isnull().sum()
```

[10]:
```
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
```

[11]:
```
df=df.drop(columns=['Name','Ticket','Cabin'])
```

[12]:
```
df.head()
```

[12]:
```
   PassengerId  Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0            1         0       3    male  22.0      1      0   7.2500        S
1            2         1       1  female  38.0      1      0  71.2833        C
2            3         1       3  female  26.0      0      0   7.9250        S
3            4         1       1  female  35.0      1      0  53.1000        S
4            5         0       3    male  35.0      0      0   8.0500        S
```

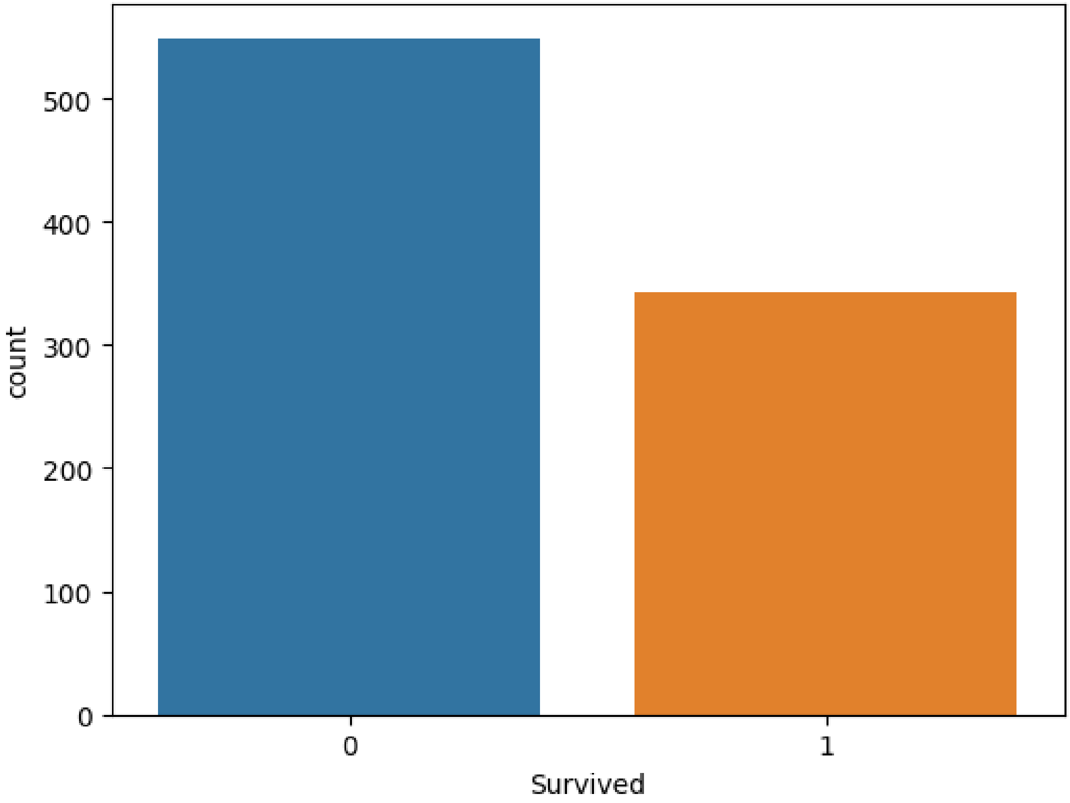
[13]:
```
age = df['Age'].median()
df['Age'].fillna(age, inplace=True)
```

[14]:
```
df.head()
```

[14]:
```
   PassengerId  Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0            1         0       3    male  22.0      1      0   7.2500        S
1            2         1       1  female  38.0      1      0  71.2833        C
2            3         1       3  female  26.0      0      0   7.9250        S
3            4         1       1  female  35.0      1      0  53.1000        S
4            5         0       3    male  35.0      0      0   8.0500        S
```

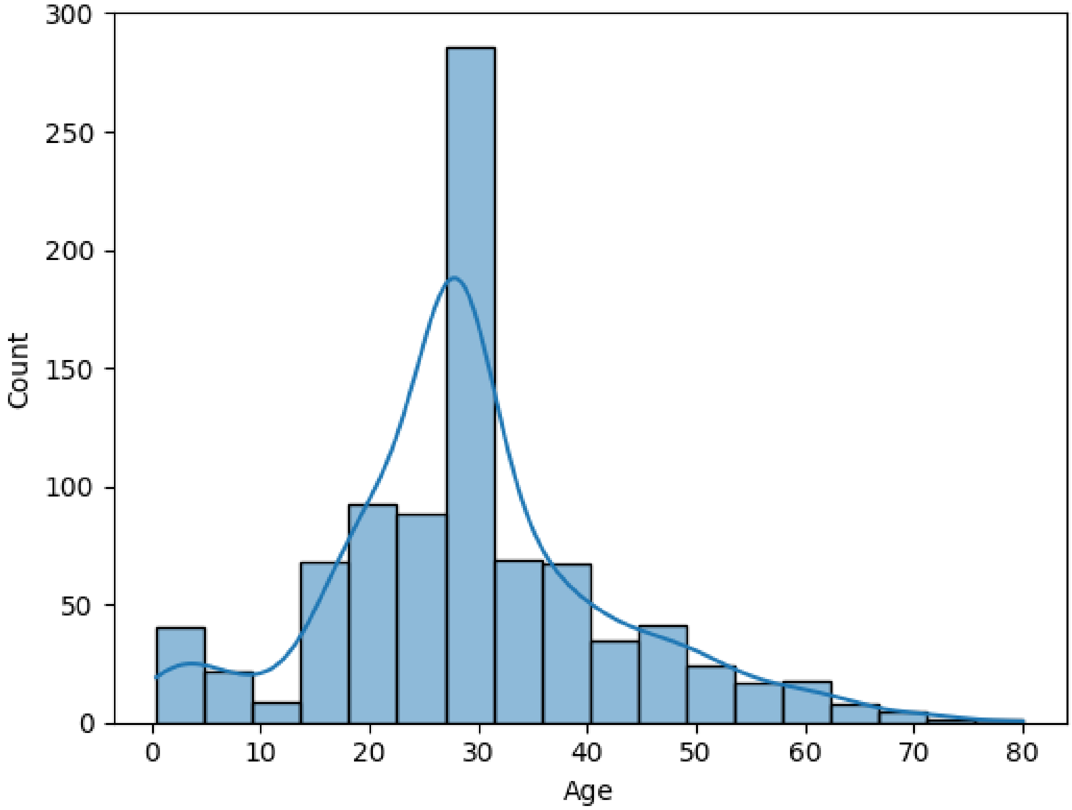
[15]:
```
sns.countplot(data=df, x='Survived')
```

[15]:
```
<Axes: xlabel='Survived', ylabel='count'>
```



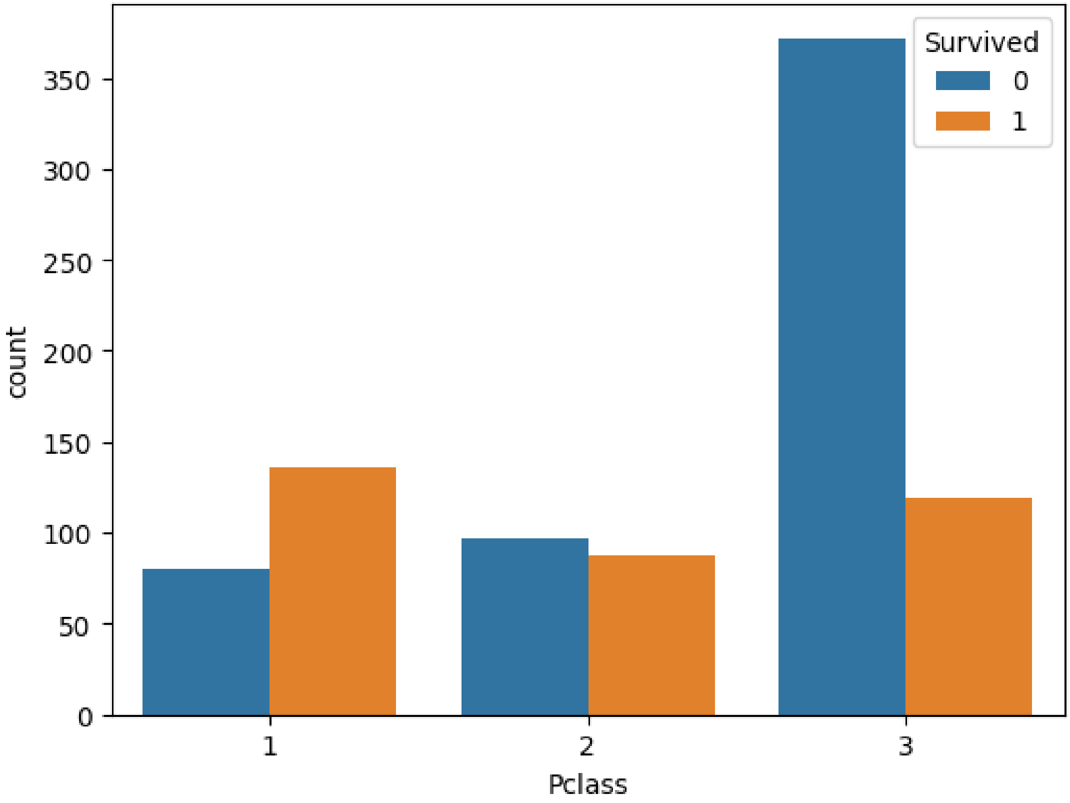
[16]:
```
sns.histplot(data=df, x='Age', bins=18, kde=True)
```

[16]:
```
<Axes: xlabel='Age', ylabel='Count'>
```



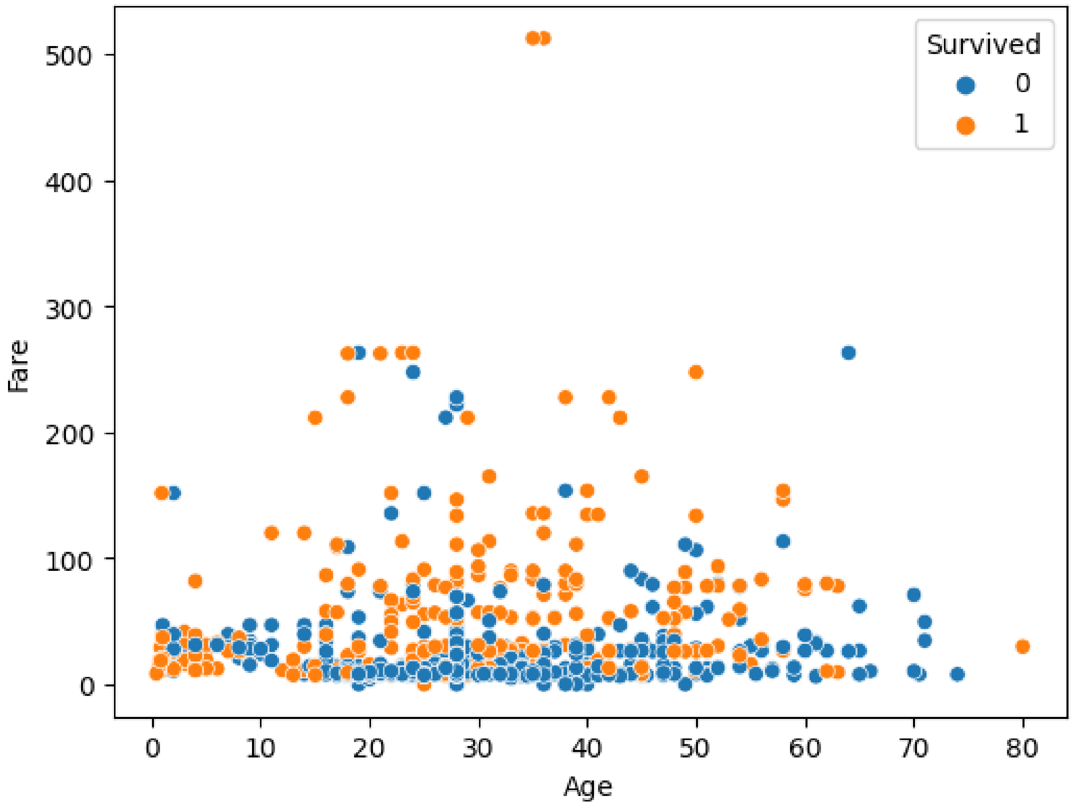
```
[17]: sns.countplot(data=df, x='Pclass', hue='Survived')
```

```
[17]: <Axes: xlabel='Pclass', ylabel='count'>
```



```
[18]: sns.scatterplot(data=df, x='Age', y='Fare', hue='Survived')
```

```
[18]: <Axes: xlabel='Age', ylabel='Fare'>
```



[19]:
```
x=df.drop('Survived', axis=1)   #Independent variable
```

[20]:
```
type(x)
```

[20]:
```
pandas.core.frame.DataFrame
```

[21]:
```
x.head()
```

[21]:
```
   PassengerId  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0            1       3    male  22.0      1      0   7.2500        S
1            2       1  female  38.0      1      0  71.2833        C
2            3       3  female  26.0      0      0   7.9250        S
3            4       1  female  35.0      1      0  53.1000        S
4            5       3    male  35.0      0      0   8.0500        S
```

[24]:
```
y=df.iloc[:,1:2]   #Dependent variable
```

[23]:
```
type(y)
```

[23]:
```
pandas.core.series.Series
```

[25]:

```
y.head()
```

[25]:

```
   Survived
0         0
1         1
2         1
3         1
4         0
```

[26]:

```
df.shape
```

[26]:

```
(891, 9)
```

[27]:

```
x.shape
```

[27]:

```
(891, 8)
```

[28]:

```
y.shape
```

[28]:

```
(891, 1)
```

[29]:

```
#Encoding
from sklearn.preprocessing import LabelEncoder
```

[30]:

```
le=LabelEncoder()
```

[31]:

```
x["Sex"]=le.fit_transform(x["Sex"])
```

[32]:

```
x["Sex"]
```

[32]:

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

[33]:

```
x.Embarked.value_counts()
```

[33]:

```
S    644
C    168
```

```
Q     77
Name: Embarked, dtype: int64
```

[34]:
```
embarked=pd.get_dummies(x["Embarked"], drop_first=True)
```

[35]:
```
embarked.head()
```

[35]:
```
   Q  S
0  0  1
1  0  0
2  0  1
3  0  1
4  0  1
```

[36]:
```
x=pd.concat([x,embarked],axis=1)
```

[37]:
```
x.head()
```

[37]:
```
   PassengerId  Pclass  Sex   Age  SibSp  Parch     Fare Embarked  Q  S
0            1       3    1  22.0      1      0   7.2500        S  0  1
1            2       1    0  38.0      1      0  71.2833        C  0  0
2            3       3    0  26.0      0      0   7.9250        S  0  1
3            4       1    0  35.0      1      0  53.1000        S  0  1
4            5       3    1  35.0      0      0   8.0500        S  0  1
```

[38]:
```
x.drop(["Embarked"],axis=1,inplace=True)
```

[39]:
```
x.head()
```

[39]:
```
   PassengerId  Pclass  Sex   Age  SibSp  Parch     Fare  Q  S
0            1       3    1  22.0      1      0   7.2500  0  1
1            2       1    0  38.0      1      0  71.2833  0  0
2            3       3    0  26.0      0      0   7.9250  0  1
3            4       1    0  35.0      1      0  53.1000  0  1
4            5       3    1  35.0      0      0   8.0500  0  1
```

[40]:
```
#Splitting into training and testing set

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)
```

[41]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[41]:
```
((623, 9), (268, 9), (623, 1), (268, 1))
```

[42]:
```
#Feature scaling
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
[43]: x_train=sc.fit_transform(x_train)
      x_test=sc.fit_transform(x_test)
```

```
[44]: x_train
```

```
[44]: array([[ 1.59014094, -1.5325562 ,  0.72592065, …, -0.12253019,
              -0.31426968,  0.60269272],
             [-1.52952238, -1.5325562 , -1.37756104, …,  0.91812372,
              -0.31426968, -1.65922031],
             [-0.23515275,  0.84844757,  0.72592065, …,  0.29950338,
              -0.31426968,  0.60269272],
             …,
             [ 0.70655928,  0.84844757,  0.72592065, …, -0.51276504,
               3.18198052, -1.65922031],
             [ 0.43528421,  0.84844757, -1.37756104, …, -0.31228976,
              -0.31426968,  0.60269272],
             [ 0.91970398, -0.34205431,  0.72592065, …,  0.13566725,
              -0.31426968,  0.60269272]])
```

```
[ ]:
```