Titanic

August 5, 2025

[6]:
```
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
import joblib
```

[31]:
```
#load dataset
df=pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/
 ↪master/titanic.csv")
```

[32]:
```
df.head()
```

[32]:
```
   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
```

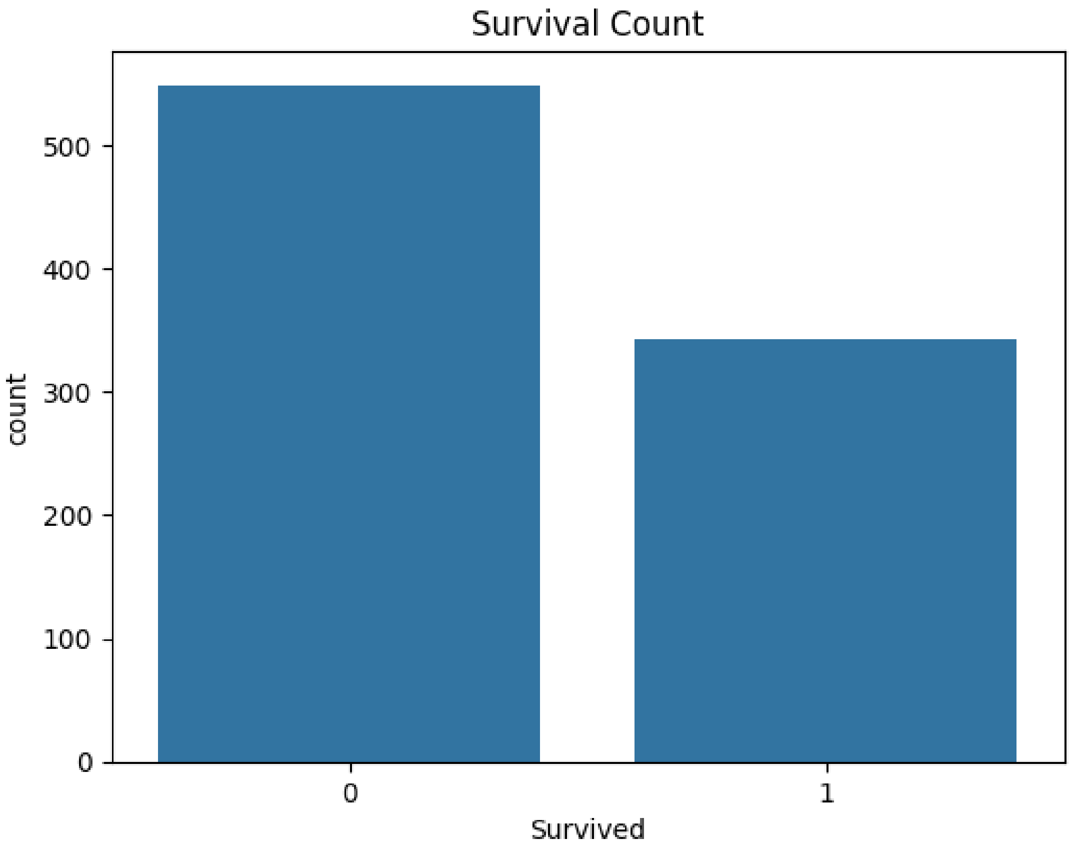
[33]:
```
df.info()
```

```
<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
```

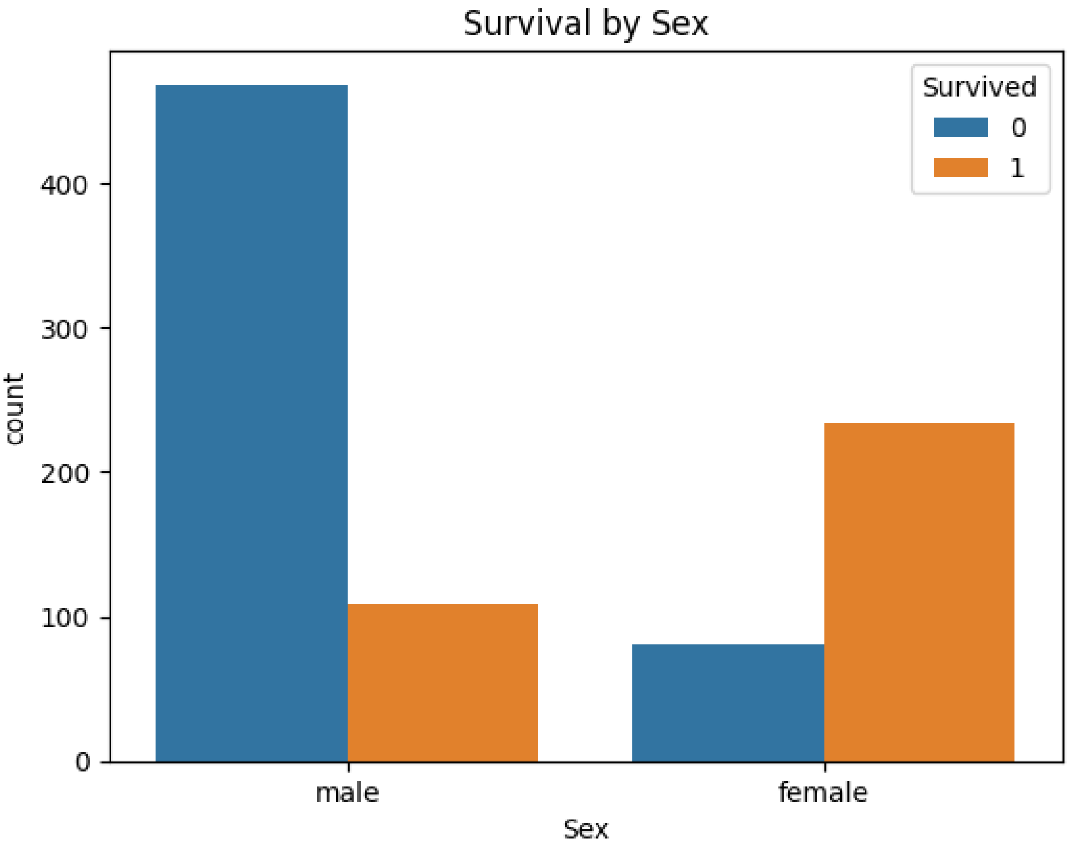
[34]:

```
#survival Count
sns.countplot(x='Survived', data=df)
plt.title('Survival Count')
plt.show()
```



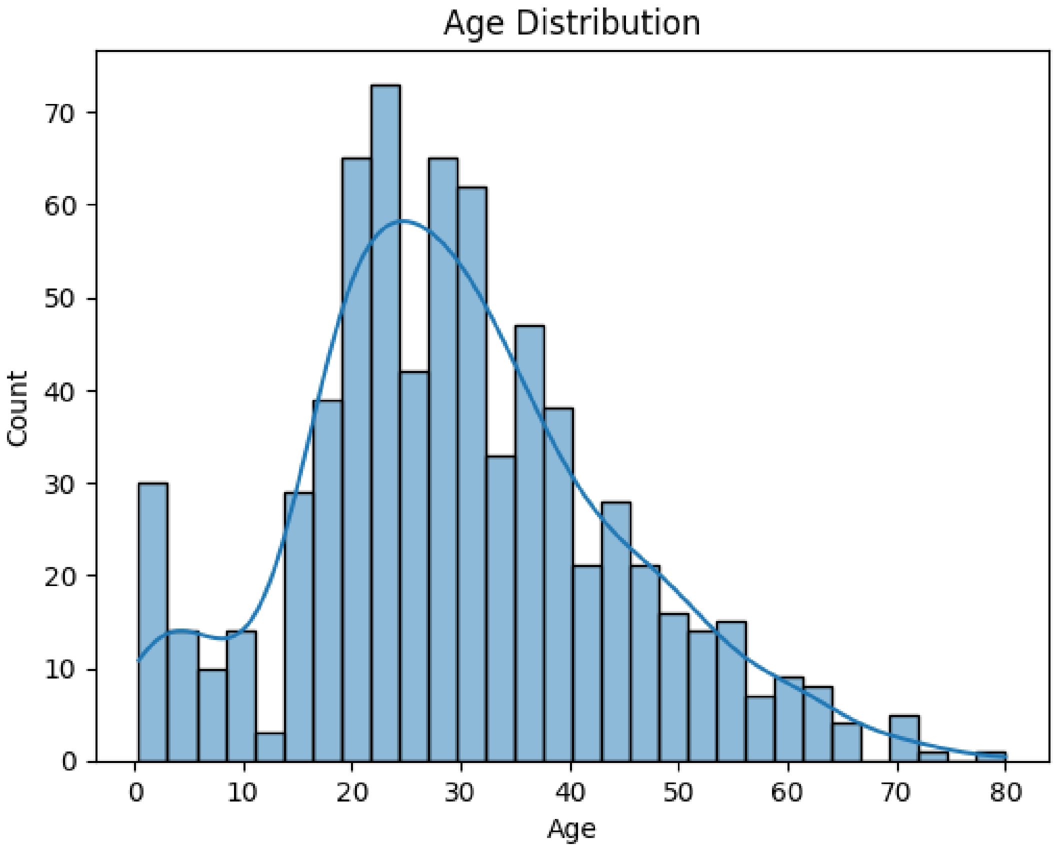
[35]:

```
#survival by sex
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title('Survival by Sex')
plt.show()
```

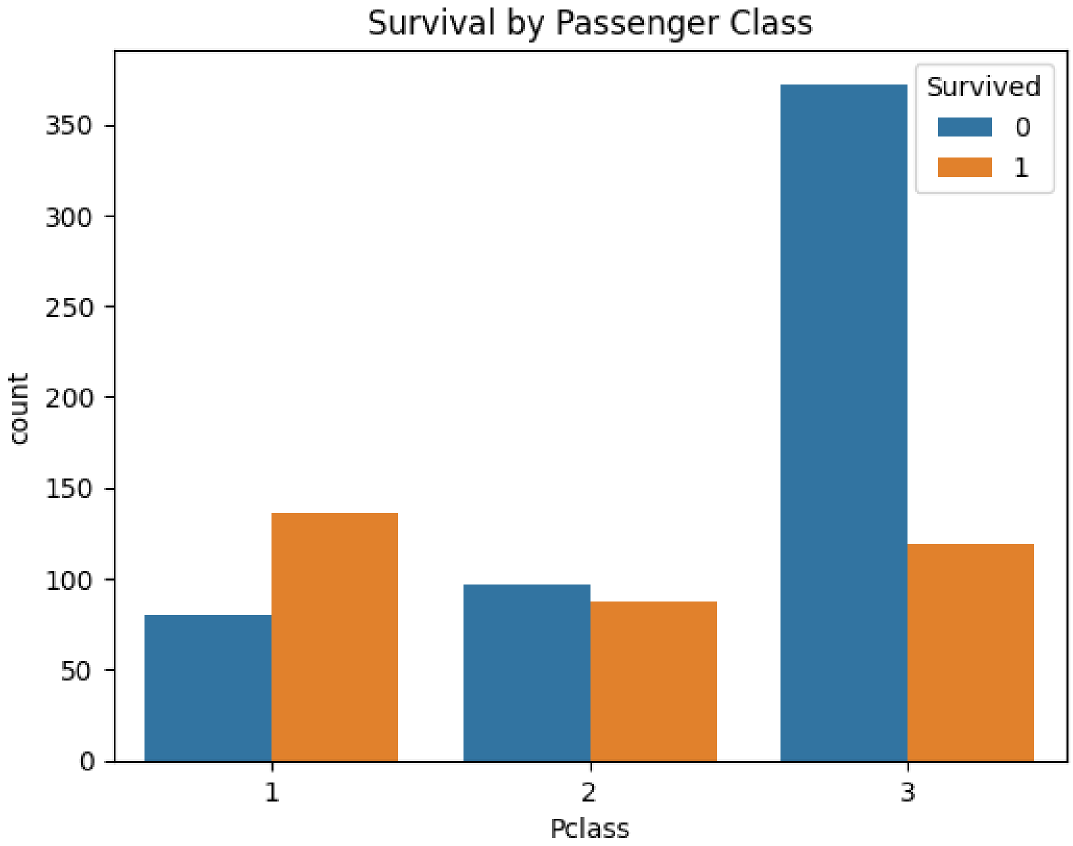


[36]:

```
#age distribution
sns.histplot(df['Age'],kde=True,bins=30)
plt.title('Age Distribution')
plt.show()
```



[15]:
```
#survival by class
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.show()
```



[37]:
```
#train the model

#select the useful feaatures
df=df[['Survived','Pclass','Sex','Age','Fare']]
df.dropna(inplace=True)
```

[38]:
```
#convert sex to numeric

df['Sex']=df["Sex"].map({'male':0,'female':1})
```

[39]:
```
# Features and Target
X = df[['Pclass', 'Sex', 'Age', 'Fare']]
y = df['Survived']
```

[40]:
```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[41]:

```
# Train model
model = LogisticRegression()
model.fit(X_train, y_train)
```

[41]:

```
LogisticRegression()
```

[43]:

```
#save model
joblib.dump(model, 'titanic_model.pkl')
```

[43]:

```
['titanic_model.pkl']
```

[44]:

```
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {acc:.2f}")
```

```
Model Accuracy: 0.76
```

[45]:

```
from sklearn.metrics import classification_report, confusion_matrix

y_pred = model.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[68 19]
 [16 40]]
              precision    recall  f1-score   support

           0       0.81      0.78      0.80        87
           1       0.68      0.71      0.70        56

    accuracy                           0.76       143
   macro avg       0.74      0.75      0.75       143
weighted avg       0.76      0.76      0.76       143
```

[ ]:

```

```